T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE ARAÇ TESPİTİ: YILDIZ TEKNİK ÜNİVERSİTESİ KAMPÜS OTOPARK ALANLARI ÖRNEĞİ

Şennur ÖZTÜRK

YÜKSEK LİSANS TEZİ

Harita Mühendisliği Anabilim Dalı

Uzaktan Algılama ve CBS Programı

Danışman

Doç. Dr. Anime Melis UZAR

Şubat, 2022

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE ARAÇ TESPİTİ: YILDIZ TEKNİK ÜNİVERSİTESİ KAMPÜS OTOPARK ALANLARI ÖRNEĞİ

Şennur ÖZTÜRK tarafından hazırlanan tez çalışması 09.02.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Dalı, Uzaktan Algılama ve CBS Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Anime Melis UZAR
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Anime Melis UZAR, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Nursu TUNALIOĞLU ÖCALAN, Üye
Yıldız Teknik Üniversitesi _____

Doç. Dr. Saygın ABDİKAN Üye
Hacettepe Üniversitesi _____

Danışmanım Doç. Dr. A. Melis UZAR sorumluluğunda tarafımca hazırlanan Derin Öğrenme İle Araç Tespiti: Ytü Kampüs Otopark Alanları Örneği başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Şennur ÖZTÜRK

İmza

Aileme…

TEŞEKKÜR

Tez çalışmam boyunca zamanını, bilgisini ve rehberliğini esirgemeyen değerli danışma hocam Doç. Dr. Melis UZAR'a,

Bana veri sağlayan ve önerilerde bulunan Arş Gör. Can Onur BAYRAK'a,

Derin öğrenme konusunda çalışmaya teşvik eden Dr. Öğr. Üyesi Alper ŞEN'e,

Beni her zaman destekleyen başta kıymetli annem Emine ÖZTÜRK olmak üzere tüm aile üyelerime,

Sürekli motive edip yanımda olan Muhsin COŞKUN'a

Tezi bitirmemde zamansal olarak büyük destekleri olan Ekol Gayrimenkul Değerleme ve Danışmanlık ailesine teşekkürü borç bilirim.

Şennur ÖZTÜRK

İÇİNDEKİLER

SİMGE LİSTESİ vii
KISALTMA LİSTESİ viii
ŞEKİL LİSTESİ ix
TABLO LİSTESİ xi
ÖZET xii
ABSTRACT xiv
1 GİRİŞ 1
1.1. Literatür Özeti ... 1
1.1.1. Derin Öğrenme Çalışmaları ... 2
1.1.2. Araç Tespiti Çalışmaları ... 5
1.2. Tezin Amacı ... 12
1.3. Hipotez ... 13
2 DERİN ÖĞRENME 14
2.1 Yapay Zeka ve Makine Öğrenmesi Kavramları ... 14
2.1.1. Denetimli Öğrenme ... 15
2.1.2. Denetimsiz Öğrenme ... 15
2.1.3. Yarı Denetimli Öğrenme ... 15
2.1.4. Pekiştirmeli öğrenme ... 15
2.2. Derin Öğrenme ... 16
2.2.1. Aktivasyon Fonsiyonları ... 17
2.2.2. Geriye Yayılım (Backpropagation) ... 19
2.2.3. Kayıp Fonksiyonları ... 20
2.2.4. Optimizasyon Algoritmaları ... 21
2.2.5. Derin Öğrenme Mimarileri ... 21
2.3. Evrişimli Sinir Ağları (Convolution Neural Network-CNN) ... 22
2.3.1. Hiperparametreler ... 24
2.4. YOLO (You Only Look Once) Mimarisi ... 26
2.4.1. YOLOv1 ... 27
2.4.2. YOLOv2 ... 27
2.4.3. YOLOv3 ... 28
2.4.4. YOLOv4 ... 29

2.4.5. YOLOv5 30
3 YÖNTEM 32
3.1. Veri Seti 33
3.2. Veri Etiketleme 33
3.3. Veri Artırımı 34
3.4. Aktarımlı Öğrenme 35
3.5. Kullanılan YOLO Mimarileri 36
3.5.1. Ölçeklendirilmiş YOLOv4 36
3.5.2. YOLOv5 Ölçekleri 38
3.6. Doğruluk Analizi Ölçütleri 39
4 UYGULAMA 42
4.1. Çalışma Alanı 42
4.2. Veri Setinin Hazırlanması 42
4.3. YOLO Model Eğitimleri 45
4.4. Kullanılan Teknolojiler 45
4.5. Model Eğitimlerinin Doğruluk Analizleri 46
5 SONUÇ VE ÖNERİLER 55
KAYNAKÇA 58
TEZDEN ÜRETİLMİŞ YAYINLAR 63

SİMGE LİSTESİ

w	Ağırlık Değeri
b	Bias Değeri
g	b/2
k	Girdi Değeri
x	Girdi Değeri
b	Kanal sayısı
L	Kayıp Fonksiyonu
y	Referans Değer
$\hat{y}$	Tahmin Değeri
N	Veri Örneği Sayısı

KISALTMA LİSTESİ

AP	Average Precion
CBS	Coğrafi Bilgi Sistemi
CNN	Convolutional Neural Network
CSP	Cross-Stage-Partial-connections
COCO	Common Objects in Context
DPİ	Dots Per İnch
DBA	Dinamik Bayes Ağı
DNN	Deep Neural Network
DLNN	Deep Learning Neural Network
DP	Doğru Pozitif
DN	Doğru Negatif
Fps	Frame Per Second
FLOPS	Floating Point Operations Per Second
GANet	Guided Attention Network
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
İHA	İnsansız Hava Aracı
mAP	Mean Average Precion
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
SSD	Single Shot Detector
SGD	Stochastic Gradient Descent
SIFT	Scale-İnvariant Feature Transform
VOTT	Visual Object Tagging Tool
YTÜ	Yıldız Teknik Üniversitesi
YSA	Yapay Sinir Ağı
YOLO	You Only Look Once
YP	Yanlış Pozitif
YN	Yanlış Negatif

ŞEKİL LİSTESİ

Şekil 2.1 Derin öğrenmenin yapay zeka kavramı içerisindeki konumu.....14
Şekil 2.2 Genelleştirilmiş derin sinir ağı yapısı.....16
Şekil 2.3 Yapay sinir hücresi modeli 17
Şekil 2.4 Örnek aktivasyon fonksiyonlarının grafikleri.....18
Şekil 2.5 Geri beslemeli ağ yapısı [41].....20
Şekil 2.6 Genel CNN yapısı [46'dan uyarlanmıştır].....22
Şekil 2.7 Evrişim işlemi ve adım kaydırma [47].....23
Şekil 2.8 Havuzlama İşlemi.....23
Şekil 2.9 Düzleştirme (flatten) İşlemi 24
Şekil 2.10 Seyreltme örneği.....26
Şekil 2.11 YOLOv1 yapısı [49'dan uyarlanmıştır] 27
Şekil 2.12 Darknet-19 [50' den uyarlanmıştır].....28
Şekil 2.13 Darknet-53 [51'den uyarlanmıştır].....29
Şekil 2.14 CSP yapısı modeli.....30
Şekil 2.15 YOLOv5 genel yapı [54'den uyarlanmıştır] 31
Şekil 3.1 İş Akışı 32
Şekil 3.2 Veri setinden örnekler.....33
Şekil 3.3 Veri artırımı örnekleri [56] 35
Şekil 3.4 Tüm ağlara uygulanabilecek cspdark modeli [60'dan uyarlanmıştır] ..36
Şekil 3.5 YOLOv4-tiny hesaplama katmanı [60'dan uyarlanmıştır].....37
Şekil 3.6 Scaled YOLOv4 katman yapıları [60'dan uyarlanmıştır].....38
Şekil 3.7 YOLOv5 ölçekleri 38
Şekil 4.1 YTÜ Davutpaşa Kampüsü (Google Earth).....42
Şekil 4.2 Veri Arttırımı Temsili Örnekler.....43
Şekil 4.3 Etiketli veri dosyası (.txt) 44
Şekil 4.4 Araç tespitinde ağaçların etkisi.....47
Şekil 4.5 Araç tespitinde gölge etkisi 48
Şekil 4.6 Koyu renkli araçlarda tespit 48
Şekil 4.7 Koyu renkli araçlarda tespit 49
Şekil 4.8 Araç Dışı Nesne Tespiti.....50
Şekil 4.9 Araç dışı nesne tespiti 50

Şekil 4.10 Araç dışı nesne tespiti 51

TABLO LİSTESİ

Tablo 1.1 Literatür İnceleme Tablosu 8
Tablo 3.1 YOLOv5 modellerinin özellikleri [61] 39
Tablo 3.2 Hata Matrisi (Confusion Matrix) 39
Tablo 4. 1 Etiketli verilerin sınıf dağılımları.....44
Tablo 4.2 Eğitimde kullanılan hiperparametre değerleri.....45
Tablo 4.3 Modellerin Eğitim Analizleri 46
Tablo 4.4 YOLOv4-YOLOv5 karşılaştırma tablosu.....52
Tablo 4.5 Sınıflara ait mAP analizleri 53
Tablo 4.6 Sınıfların tespit performansı incelemesi 54

ÖZET

Derin Öğrenme ile Araç Tespiti:
Yıldız Teknik Üniversitesi Kampüs Otopark Alanları Örneği

Şennur ÖZTÜRK

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Melis UZAR

Günümüzde otomatik araç tespiti, akıllı şehirlerde ulaşım, afet ve kriz yönetimi, otoyol yönetimi, gerçek zamanlı araç tespiti ve otopark yönetimi gibi birçok farklı alanda kullanılmaktadır. Son yıllarda, araç tespit çalışmalarında yaygın olarak derin öğrenme yöntemleri tercih edilmektedir. Bu yöntemin kullanımının yüksek doğruluk ve hız avantajının yanı sıra, modeldeki katman sayısı, bağlantı özellikleri ve değişkenlerin seçiminden kaynaklanan problemleri de beraberinde getirdiği tespit edilmiştir. Ayrıca, yağmur, sis, güneş gibi olumsuz hava şartlarında sayısal görüntülerden araç tespitinde buğu ve gölge kaynaklı araçları tespit edememe, çift tespit etme ve sınıf karışıklığı gibi önemli sorunlara neden olmaktadır. Bu sebeple, araç tespitinde güncel, hızlı ve doğru bilgi elde etmek hala üzerinde çalışılan ve çözümlenmesi gereken önemli bir problemdir. Bu çalışmada amaç, derin öğrenme mimarilerinden olan YOLO versiyonlarının araçları tespit etmedeki model performansı ve otopark alanlarındaki başarısını analiz etmektir. Bu kapsamda YTÜ Davutpaşa kampüsü 2018 yılına ait, uygun hava koşullarında insansız hava aracı ile elde edilmiş 5472*3648 piksel boyutunda, 72 dpi çözünürlüğe sahip görüntüler kullanılmıştır. Elde edilen veri seti araba, otobüs ve minibüs olmak üzere VoTT

programı kullanılarak etiketlenmiştir. Etiketlenen veri seti, YOLOv4-CSP, YOLOv4-tiny YOLOv4-P5, YOLOV4-P6, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x mimarilerinde, aktarımlı öğrenme kullanarak eğitilmiştir. Elde edilen ağırlıklar belirlenen otopark alanlarında uygulanarak sonuçlar karşılaştırılmıştır. Sonuçların doğruluk analizleri hesaplanan F1-Skor değerine göre YOLOv4-CSP %68, YOLOv4-tiny %89, YOLOv4-P5 %61, YOLOv4-P6 %53, YOLOv5n %0.79 YOLOv5s %81, YOLOv5m %0.79, YOLOv5l %0.79 ve YOLOv5x %80 olarak bulunmuştur. Hesaplanan mAP doğruluk değerine göre ise YOLOv4-CSP %76, YOLOv4-tiny %75, YOLOv4-P5 %73, YOLOv4-P6 %75, YOLOv5n %0.82 YOLOv5s %80, YOLOv5m %0.84, YOLOv5l %0.79 ve YOLOv5x %79 olarak bulunmuştur. Sonuçta, F1-Skor değerinde YOLOv4-tiny modeli, mAP değerinde YOLOv5m modellerinin çalışmada analiz edilen diğer modellerden daha iyi sonuç verdiği tespit edilmiştir.

Anahtar kelimeler: Araç tespiti, derin öğrenme, YOLO

ABSTRACT

Vehicle Detection with Deep Learning:

Yildiz Technical University Campus Parking Areas Example

Şennur ÖZTÜRK

Department of Geomatic Engineering

Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Melis UZAR

Nowadays, automatic vehicle detection is used in many different areas such as transportation in smart cities, disaster and crisis management, highway management, real-time vehicle detection and parkinglot management. In recent years, deep learning methods have been widely preferred in vehicle detection studies. It has been determined that the use of this method brings with it the advantages of high accuracy and speed, as well as problems arising from the number of layers in the model, connection properties and selection of variables. In addition, adverse weather conditions such as rain, fog and sun cause important problems such as the inability to detect vehicles caused by mist and shadow, double detection and class confusion in vehicle detection from digital images. For this reason, obtaining up-to-date, fast and accurate information in vehicle detection is still an important problem that is being studied and needs to be resolved. The aim of this study is to analyze the model performance of YOLO versions, which is one of the deep learning architectures, in detecting vehicles and their success in parking areas. In this context, images of the YTU Davutpaşa campus in 2018, taken by an unmanned aerial vehicle in suitable weather

conditions, with a size of 5472 * 3648 pixels and a resolution of 72 dpi were used. The created dataset was labeled as car, bus and minibus using the VOTT program. The tagged dataset was trained in YOLOV4-CSP, YOLOV4-tiny YOLOV4-P5, YOLOV4-P6, YOLOV5n, YOLOV5s, YOLOV5m, YOLOV5l, YOLOV5x architectures using transfer learning. Obtained weights were applied in the designated parking areas and the results were compared. Accuracy analyzes of the results according to the F1-Score values were determined as YOLOv4-CSP 68%, YOLOv4-tiny 89%, YOLOv4-P5 61%, YOLOv4-P6 53%, YOLOv5n 0.79% YOLOv5s 81%, YOLOv5m 0.79%, YOLOv5l 0.75% and YOLOv5l 0.75%. calculated. Accuracy analyzes of the results according to the mAP values were determined as YOLOv4-CSP 76%, YOLOv4-tiny 75%, YOLOv4-P5 73%, YOLOV4-P6 75%, YOLOV5n 0.82% YOLOv5s 80%, YOLOv5m 0.84%, YOLOv5l 0.79% and YOLOv5 calculated. As a result, it was determined that the YOLOV4-tiny model at the F1-Score value and the YOLOv5m model at the mAP value gave better results than the other models analyzed in the study.

Keywords: Vehicle detection, deep learning, YOLO

1 GİRİŞ

Hızla artan nüfus ve trafiğe katılan ulaşım araçları, özellikle kentsel alanlarda otoyol ve otoparkların yetersiz kalmasına ve kötü yönetilmesine sebep olmaktadır. Yapay zekâ ve bilgisayarlı görü alanlarındaki teknolojik gelişmeler, çeşitli disiplinlerdeki araştırmalara yeni bir yön vermiştir. Son on yılda bu gelişmeler araçların gerçek zamanlı tespitini mümkün kılmıştır. Dolayısı ile bu çalışmalar, otopark alanlarının, özel ve kamu kurumlarında, ihtiyaçlara yönelik olarak akıllı sistemlere entegre edilip aktif olarak kullanılabilmesi ve karar verme süreçlerinin hızlanmasına olanak tanımaktadır. Bu amaçla başta akademik alanda ilerleyen çalışmalarla geliştirilen yöntem ve algoritmalar, uygulama aşamasında sorunlarla karşılaşmaktadır. Bu sorunların genel olarak geliştirilen yöntemlerin hızından ve doğruluk performansından kaynaklı amacına hizmet edememesinden veya hava olaylarının yol açtığı görüntülerdeki bulanıklık ve parlama gibi görüntüyü kalitesizleştiren etkilerden kaynaklandığı bilinmektedir.

Nesne algılama teknolojileri araştırmacılar tarafından doğruluk ve performans açısından optimum sonuçlara ulaşmak için çalışılan güncel bir konudur. Özellikle derin öğrenme yöntemleri üzerine yoğunlaşılarak modeller geliştirilmektedir.

Derin öğrenme, beynin yapısı ve işlevinden esinlenerek oluşturulmuş makine öğreniminin alt alanıdır. Derin öğrenme, çok katmanlı sinir ağları ve geri yayılım algoritmaları kullanarak ağırlıkların aşama aşama güncellendiği bir öğrenme yapısıdır. Derin öğrenme mimarilerinden olan Evrişimli Sinir Ağları (Convolutional Neural Network-CNN), bilgisayarlı görü çalışmalarında nesnelerin sınıflandırılması ve tespitinde büyük bir başarı göstermiştir.

1.1. Literatür Özeti

Bu bölümde araştırmacılara ait araç tespiti çalışmaları başta olmak üzere nesne tespiti ile ilgili olarak pek çok bilgisayar teknolojilerine dayalı yöntemin gelişimleri, sınırları, doğrulukları ve performansları araştırılmıştır. İlk olarak derin öğrenme ve YOLO mimarisinin kullanım alanları incelenmiştir. Daha sonra araç

tespitinde derin öğrenme ve yapay zekâ tabanlı yöntemlerin uygulandığı çalışmalar araştırılmıştır.

1.1.1. Derin Öğrenme Çalışmaları

İlk yapılan çalışmalar incelendiğinde sinir ağlarına dayanmayan makine öğrenmesi tekniklerinde nesne tespiti için,

- Özellik Tabanlı Kademeli Sınıflandırıcılar (Jones ve Viola nesne tespit yapısı-Haar),
- SIFT Metodu,
- Yönlendirilmiş Gradyanların Histogramı (Histogram of Oriented Gradients-HOG),

gibi aşamalı yöntemler kullanılmıştır. Bu yöntemler, özellikleri tanımlamak için karar ağaçları [2] ya da destek vektör makineleri (Support Vektor Machine-SVM) [3] kullanarak veriyi sınıflandırma işlemlerini gerçekleştirmektedir. Fakat daha iyi gösterimlere gereksinim duyan karmaşık problemlerde bu teknikler yetersiz kalmaktadır [4]. Sinir ağlarına dayanan, bölgesel çalışan R-CNN, FAST R-CNN, FASTER R- CNN modelleri ve görüntü bütünü ile çalışan SSD, YOLO modelleri gibi derin öğrenme yaklaşımlarının karmaşık problemleri çözebilme yeteneği sebebiyle pek çok farklı alanda birçok çalışma yapılmıştır [5, 6, 7, 8]. Ses tanıma, yüz tanıma, hedef kitlesi tespiti ve sürücüsüz araç kullanımı gibi alanlar, yöntemin başarılı olarak kullanıldığı alanlarından birkaç tanesidir

Derin öğrenmenin yaygın olarak kullanıldığı konulardan biri doğal afetlerin risk analizinin yapıldığı çalışma alanlarıdır. Bu çalışmalardan Nhua vd. [9]'da heyelan duyarlılık haritalaması için farklı optimizasyon algoritmaları kullanarak derin öğrenmenin etkinlik değerlendirmesini sunulmaktadır. Makalede heyelana sebebiyet verebilecek eğim, toprak tipi gibi özniteliklerin derin ağda girdi olarak ele alınarak ağın geri yayılım özelliği ile her faktörün ağırlıkları oluşturulmuş nihayetinde %84 performans ile karşılaştırılan diğer rastgele orman, karar ağaçları ve lojistik modelden daha iyi performans gösterdiği izlenmiştir. Bir diğer afet çalışması olan Bui vd. [10]'da ani sel olaylarını tahmin etmek için yeni bir derin öğrenme sinir ağı yaklaşımı sunmaktadır. Bu çalışmada da sele etki eden unsurlar

belirlenmiş, destek vektör makineleri ile karşılaştırılmıştır. Sonuçta derin öğrenme sinir ağı (DLNN) %92 doğruluk sonucuyla daha başarılı bir model olmuştur.

Derin öğrenmede segmentasyon çalışmaları özellikle tıp [13, 14], madencilik [16], tarım [12], orman [11] ve şehir alanlarında, yoğunlukla kullanıldığını görmekteyiz. Örneğin: Wang vd. [11]'de zararlı bir bitki türü olan solanum rostratum dunal'ın hasar tespitini belirlemeyi amaçlayan derin öğrenmeye dayalı bir görüntü segmentasyon yöntemi üzerine çalışılmıştır. Araştırılan bu makalede insansız hava aracı (İHA) ile elde edilen görüntüler kullanılarak, solanum rostratum dunal bitkisinin olduğu alanlar piksel boyutunda ayıklanıp etiketlenmiştir. Daha sonra U-Net tabanlı evrişimsel sinir ağına dayalı geliştirilen Deep Solanum-Net kullanılarak eğitilmiştir. Bu yöntem CNN tabanlı PSP ve Deeplab V3+ ile karşılaştırılmıştır. Boyutu 640*640 olan görüntüler ile çalışıldığında % 90 kesinlik ve % 90 duyarlılık değerlerine ulaşan modelin bu tip çalışmalarda başarıyla kullanılabileceği sonucuna ulaşılmıştır.

Rawat vd. [12]'de segmentasyon yöntemi ile çeltik alanlarının zamansal analizi tespiti amaçlamıştır. Uydu görüntüleri kullanılan bu makalede, CBSI-NDVI, 1D-CNN, 1D-LSTM modelleri analiz edilmiş ve karşılaştırılmıştır. Küçük boyutlu veri kümeleri ile çalışıldığında 1D-CNN bu karşılaştırmada %94 başarısı ile en iyi çözümü sağlamıştır.

Segmentasyon çalışması tıbbi görüntülerle de çok başarılı sonuçlarla devrim yaratmıştır [13]. Kanser, tümör gibi farklı görüntüleme yöntemleri ile izleyebildiğimiz önemli hastalıkların görüntülerden otomatik tespitinin, doktorlara, teşhis ve takip aşamalarında büyük yardımı olmakta ve iş yükünü azaltmaktadır [14].

Bir diğer derin öğrenmenin yoğunlukla kullanıldığı alan ise görüntülerden nesne tespitidir. Nesne tespitinde objeler sınırlayıcı bir kutu (Bounding Box) ile temsil edilirler. Şehir alanlarında özellikle akıllı kentlerin yönetimine yardımcı olacak araç, bina ve insan tespiti üzerine çalışmalar yürütülmektedir. Safanova vd. [15]'te CNN modelini, ağaç hasar tespit sınıflandırılmasında kullanılmıştır. Modelde ağaçlar farklı hasar aşamalarında etiketlenerek evrişimsel sinir ağında eğitilmiştir. Model doğruluğu çalışmada 1 olarak hesaplanmış ve geliştirilen

modelin insansız hava aracından elde edilen renkli görüntülerde başarılı sonuçlar verdiği gözlenmiştir. Ayrıca, aynı çalışmada veri arttırma teknikleri kullanıldığında veya kullanılmadığında alınan sonuçlar değerlendirilmiştir. Bu çalışmada veri artırımı uygulanarak eğitilmiş tüm CNN ağlarında doğruluğun artırdığı tespit edilmiştir.

Aktaş [17]'de görme bozukluğuna sahip insanlara yardımcı olan dokunsal parke yüzeylerinin otomatik tespitini amaçlayan bir model tasarlanmıştır. Modelde YOLOv3 modelini DenseNet modeli ile birleştirerek dokunsal parke yüzeylerinin tespiti yapılmıştır. Oluşturulan YOLOv3-DenseNet modeli %81 ile YOLOv2 ve YOLOv3 ile karşılaştırıldığında en iyi F1-Skoruna ulaşmıştır.

YOLO mimarisinin sahip olduğu tespit hızı, otomatik sürüş sistemlerinin gelişimi içinde büyük yarar sağlamıştır [18, 19]. Otomatik Sürüş Sistemlerinde (ADS) ve Sürücü Yardım Sistemlerinde (DAS) nesne algılama hayati bir rol oynamakla birlikte, küçük araç nesneleri için mevcut gerçek zamanlı algılama modelleri, düşük hassasiyet ve düşük performans sorunlarına sahiptir [20]. Han vd. [20]'de bu sorunları çözmek amacıyla, küçük araç nesnelerine uygulanacak, YOLOv2 derin öğrenme çerçevesini temel alan optimize edilmiş YOLOv2 (O-YOLOv2) adlı yeni bir gerçek zamanlı nesne algılama modeli önermişlerdir. Önerilen modelde, YOLOv2'ye evrişim katmanları eklenerek ağın özellik çıkarma yeteneğini güçlendirecek yeni bir yapı tanıtılmıştır. Açık kaynaklı veri seti olan KITTI kullanılarak etkinliği araştırılan modelde %94 doğruluğa ulaşılmıştır.

Wu vd. [21]'de yaptığı çalışmada kanal budama tabanlı YOLOv4 kullanılarak elma çiçekleri tespit edilmiştir. Ağaçlardan alınan elma çiçekleri görüntüleri, Labellmg programı ile etiketlenerek 416*416 piksel boyutlarında aktarımlı öğrenme kullanılarak eğitilmiştir. Önerilen yöntemin doğruluğunun %97 ve algılama hızının 72.33 fps (frame per second) değerine ulaşabileceği görülmüştür.

Tan vd. [22]'de MS COCO verisi ile eğitilmiş YOLOv4 modelini insansız hava aracı görüntülerine uygun hale getirmeyi amaçlamışlardır. Bunun için YOLOv4 yapısına bazı eklemelerde bulunmuşlardır. Model YOLOv4'e göre %5 iyileşme sağlamıştır.

Li vd. [23]'de sofralık üzümlerin tespiti ve üzüm toplama robotlarına altyapı sunmak amacıyla YOLO-Grape modeli önermişlerdir. Yaprak altlarında kalan veya

üst üste binen üzümler için YOLOv4-tiny modeline bazı eklemeler yapılmıştır. Eklenen alt örnekleme füzyon yapısı, Mish etkinleştirme işlevi ve dikkat mekanizmasının tahmin edilen kutuların kaybolması sorunu azalttığı tespit edilmiştir. YOLO-Grape modeli, Faster-RCNN (Resnet50), SSD300, YOLOv4 ve YOLOv4-tiny ile karşılaştırıldığında %90 F1-Skoru ile hepsinden daha iyi sonuç vermiştir.

1.1.2 Araç Tespiti Çalışmaları

Araçların niteliklerinin iyileşmesi ile hayatımız daha rahat hale gelse de daha çok araç daha fazla trafik problemine yol açmaktadır [33]. Bu sorunların çözümü ve yönetimi için gerçek zamanlı olarak araçların doğru tespit edilmesi büyük önem arz etmektedir.

Cheng vd. [24]'te araçları tespit etmek için piksel bazlı bir çalışma yapmış araç ve araç dışı nesnelerin renklerini ayıran bir renk filtresi uygulamışlardır. Araçları Canny kenar yakalama dedektörü kullanarak belirlemişler ve Dinamik Bayes Ağı (DBN) ile sınıflandırmışlardır. Cheng vd. [25]'te yaptığı bir çalışmada, eski yöntemlerin karmaşık arka plan olduğunda yetersiz kaldığını ileri sürmüşlerdir. Bu sebeple derin sinir ağının (DNN) eğitim verilerinden, zengin özellikleri öğrenebilir yapısından dolayı DNN kullanmaya karar vermişlerdir. Fakat DNN üzerinde modelin farklı ölçekte verileri de tespit edebilmesi amacı ile havuzlama katmanında değişiklikler yaparak HDNN (Hybrid DNN) modeli sunmuşlardır.

Almeida vd. [26]'da otopark alanlarına ilişkin bir veri seti tanıtılmıştır. Bu veri setinde otoparktaki boş ve dolu alanlar etiketlenmiştir. SVM kullanılarak bu veri seti eğitilmiştir. Hem eğitim hem de test için aynı verinin kullanıldığı deneylerde, %99'un üzerinde doğruluk oranlarına ulaşılmıştır. Ancak farklı verilerle çalışıldığında en iyi sonuç yaklaşık %89 olmuştur. Bu fark çalışmanın eğitim setine bağımlılığını göstermiştir.

Cai vd. [27]'de araç tespiti için GANet (Guided Attention Network) mimarisini kullanılmaktadır. GANet temel olarak 3 bölümden oluşur. İlki VGG-16 ve ResNet-50 gibi bilinen derin öğrenme modellerini temel alan omurgadır, ikinci aşamada omurga ağının yan çıkışları yukarıdan aşağıya kaynaştırılır. Son aşamada ise arka plan dikkat (Background Attention-BA) modülü devreye girer. Arka plan dikkat

modülü, özellik piramidi için ayırt edici arka plan bilgilerini öğrenmek amacıyla özellik haritalarının farklı kanallarını güçlendirir. Önerilen yöntemin etkinliğini göstermek için üç veri seti (UAVDT, CARPK ve PUCPR+) üzerinde eğitimler yapılmıştır. Makalede farklı veri setlerinde pek çok farklı analizler yapılmış ve yöntemin etkinliği ortaya konmuştur.

Mustafa [28]'de içerisinde araçlarında olduğu etiketli bir eğitim seti üzerinde HOG ile özellik çıkarma işlemi gerçekleştirilmiştir. Model daha sonra SVM ve karar ağaçları ile veri sınıflandırılmıştır. Bu iki yöntem aynı veri seti ile eğitilmiş ve değerlendirme sonucunda SVM'nin karar ağacından daha iyi performans gösterdiği tespit edilmiştir.

Wang vd. [29]'da KITTI veri seti kullanarak Faster-RCNN, R-FCN, SSD, RetinaNet ve YOLOv3 modellerini test etmiştir. Ulaşılan sonuçlara göre iki aşamalı olarak bölgesel çalışan R-CNN tipi modeller daha doğru sonuç verirken görüntüyü bütün olarak ele alan YOLO, SSD gibi modellerin hızının daha yüksek olduğu görülmüştür. Makalede hız ve doğruluk analizleri düşük, orta ve iyi olarak sınıflandırılmış ihtiyaca yönelik karar verilmesi ve kullanılmasına zemin hazırlamıştır.

Tan [30]'da farklı marka ve model olmak üzere 2000 görüntüden oluşan araç veri setini, transfer öğrenme kullanarak Faster-R-CNN Resnet 50, Faster-RCNN ResNet101, R-FCN ResNet 101 ve SSD mobilenet ile tekrar eğitmiş ve test verileri ile model doğruluklarını kıyaslamıştır. Ulaşılan sonuçlar incelendiğinde %94 ile Faster-RCNN-ResNet 50 modeli en başarılı model olmuştur.

Bui vd. [31]'de karmaşık alanlarda (örn: kavşaklar) trafik analizi yapabilmek için YOLO ve DeepSort gibi nesne algılama ve izleme modellerini kullanmıştır. Araç sayma problemini iyileştirmek için, özellikle karmaşık hareketlerin olduğu alanlarda, çeşitli senaryolarla iyi çalışabilen bölge izleme yaklaşımı önermişlerdir. Önerilen model CVPR AI City Challenge 2020 veri setinde değerlendirilmiştir. Buna göre, yöntem %85 doğruluk elde etmiştir.

Miao vd. [32]'de araçları tespit etmek için gece alınan görüntüleri kullanmıştır. Görüntülerde MSR algoritmasını kullanarak iyileştirme yapılmış daha sonra etiketlenen görüntüler YOLOv3 ile eğitilmiş ve %93 doğruluk 30.03 fps hızına

ulaşılmıştır. Sonuçlar Faster-RCNN ve SSD ile karşılaştırılmıştır YOLOv3 modelinin hız ve doğruluk açısından daha iyi olduğu gözlenmiştir.

Mandal vd. [33]'de yollardaki araç sayısını otomatik olarak saymak için bir tespit-izleme çerçevesi uygulamışlardır. En iyi araç sayma çerçevesine erişmek için farklı izleme sistemleriyle birleştirilmiş birden çok nesne algılama modeli kombinasyonu uygulanmıştır. Louisiana Ulaştırma ve Geliştirme Departmanından elde edilen 9 saatten fazla trafik video verilerinin manuel olarak sayılan veriler ile karşılaştırılmıştır. Sonuçlar, CenterNet ve DeepSORT, Detectron2 ve DeepSORT ile YOLOv4 ve DeepSORT kombinasyonlarının tüm araçlar için en iyi toplam sayım yüzdesini ürettiğini göstermektedir. Makalede sayısal doğruluk değerleri verilmemektedir.

Kılıç vd. [34]'de İHA ile elde edilen görüntülerde arabaları tespit etmek ve saymak için görüntüyü bütün olarak ele alan tek aşamalı nesne dedektörü modeli önermektedir. HLCNN (Heatmap Learner Convolutional Neural Network) olarak adlandırılan model, hedef araba örneklerinin ısı haritasını tahmin etmek için kullanılmıştır. Hedef arabaların ısı haritasını öğrenmek için, tam bağlantılı katmanlar yerine adaptasyon katmanları olarak üç evrişim katmanı ekleyerek CNN mimarisi geliştirilmiştir. Omurga ağı olarak VGG-16 kullanılan modelde, önerilen yöntem, başarılı bir şekilde araba sayısını belirlemekte ve hedef arabaların merkezini hassas bir şekilde tespit etmektedir. Modelin performansını değerlendirmek için iki farklı araba veri seti (PUCPR+ ve CARPK) üzerinde model uygulanmıştır. Önerilen HLCNN yöntemi, dönemin bazı mevcut yöntemleri ile karşılaştırıldığında en düşük hata payına sahip model olduğunu tespit edilmiştir.

Tablo 1.1 Literatür İnceleme Tablosu

No.	Başlık	Yıl	Yazar	Veri Seti	Yöntem	Kay. No.
1	Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area	2020	Nhu V. vd	Heyelan oluşumuna etki eden faktörler	DLNN	[9]
2	A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area	2020	Bui D. T. vd.	Sel oluşumuna etki eden faktörler	DLNN	[10]
3	An image segmentation method based on deep learning for damage assessment of the invasive weed Solanum rostratum Dunal	2021	Wang Q. vd.	Hava Fotoğrafları	U-Net	[11]
4	Deep learning-based models for temporal satellite data processing: Classification of paddy transplanted fields	2021	Rawat A. vd.	Uydu Görüntüsü	CNN	[12]

Tablo 1.1 Literatür İnceleme Tablosu (devamı)

No.	Başlık	Yıl	Yazar	Veri Seti	Yöntem	Kay. No.
5	A review of deep learning based methods for medical image multi-organ segmentation	2021	Fu Y. vd.	Tıbbi görüntüler	FCN ve GAN	[13]
6	Integration of deep learning-based image analysis and genomic data in cancer pathology: A systematic review	2022	Schneider L. vd.	Tıbbi görüntüler	CNN	[14]
7	Detection of Fir Trees (Abies sibirica) Damaged by the Bark Beetle in Unmanned Aerial Vehicle Images with Deep Learning	2019	Safonova A. vd.	Hava Fotoğrafları	CNN	[15]
8	Derin Öğrenme Yöntemleri İle Görüntü İşleme Uygulamalar	2020	Aktaş A.	Yersel Fotoğraflar	YOLOv3-DenseNet	[17]
9	Real-time object detection based on YOLO-v2 for tiny vehicle object	2020	Han X. Vd	Yersel Fotoğraflar	YOLOv2	[20]

Tablo 1.1 Literatür İnceleme Tablosu (devamı)

No.	Başlık	Yıl	Yazar	Veri Seti	Yöntem	Kay. No.
10	Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments	2020	Wu D. vd.	Yersel Fotoğraflar	YOLOv4	[21]
11	YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm	2021	Tan L. vd.	Hava Fotoğrafları	YOLOv4	[22]
12	A real-time table grape detection method based on improved YOLOv4-tiny network in complex background	2021	Li H. vd.	Yersel Fotoğraflar	YOLOv4	[23]
13	Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks	2012	Cheng H. Y. vd.	Hava Fotoğrafları	Dinamik Bayes Ağı (DBN)	[24]
14	Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks	2014	Chen X. vd.	Uydu Görüntüsü	CNN	[25]

Tablo 1.1 Literatür İnceleme Tablosu (devamı)

No.	Başlık	Yıl	Yazar	Veri Seti	Yöntem	Kay. No.
15	PKLot – A robust dataset for parking lot classification	2015	Almeida P. vd.	Yersel fotoğraflar PKLot	SVM	[26]
16	Guided Attention Network for Object Detection and Counting on Drones	2019	Cai Y. vd.	Hava Fotoğrafları "UAVDT, CARPK, PUCPR+"	GAN	[27]
17	Vehıcle Detectıon and Trackıng Usıng Machıne Learnıng Technıques	2019	Mustafa S. M.	Yersel Fotoğraflar	SVM	[28]
18	A Comparative Study of Stateof-the-Art Deep Learning Algorithms for Vehicle Detection	2019	Cai Y. vd.	Yersel Fotoğraflar	CNN	[29]
19	Derin Öğrenme Yardımıyla Araç Sınıflandırma	2019	Tan Z.	İnternet Kaynaklı Araç Fotoğrafları	Faster-RCNN ResNet50	[30]

Tablo 1.1 Literatür İnceleme Tablosu (devamı)

No.	Başlık	Yıl	Yazar	Veri Seti	Yöntem	Kay. No.
20	A Vehicle Counts by Class Framework using Distinguished Regions Tracking at Multiple Intersections	2020	Bui K.H.N. vd.	Trafik videosu	YOLO + DEEPSORT	[31]
21	A Nighttime Vehicle Detection Method Based on YOLO v3	2020	Yan M. vd.	Yersel Fotoğraflar	YOLOv3	[32]
22	Object Detection and Tracking Algorithms for Vehicle Counting: A Comparative Analysis	2020	Mandal V. vd.	Trafik videosu	YOLOv4 + Deep SORT	[33]
23	An accurate car counting in aerial images based on convolutional neural networks	2021	Kilic E. Ve Ozturk S.	Hava Fotoğrafları ‘’CARPK”	HLCNN	[34]

1.2. Tezin Amacı

Bu çalışmada, derin öğrenme mimarilerinden olan YOLO (You look Only Once) mimarisi modelleri kullanılarak YTÜ Davutpaşa Kampüsü otopark alanlarında araç tespiti ile modellerin performans ve doğruluk analizleri amaçlanmaktadır. Bu hedefle çalışma alanı kapsamında derin öğrenme yöntemlerinin avantaj ve dezavantajları mesleki disiplin açısında araştırılmıştır.

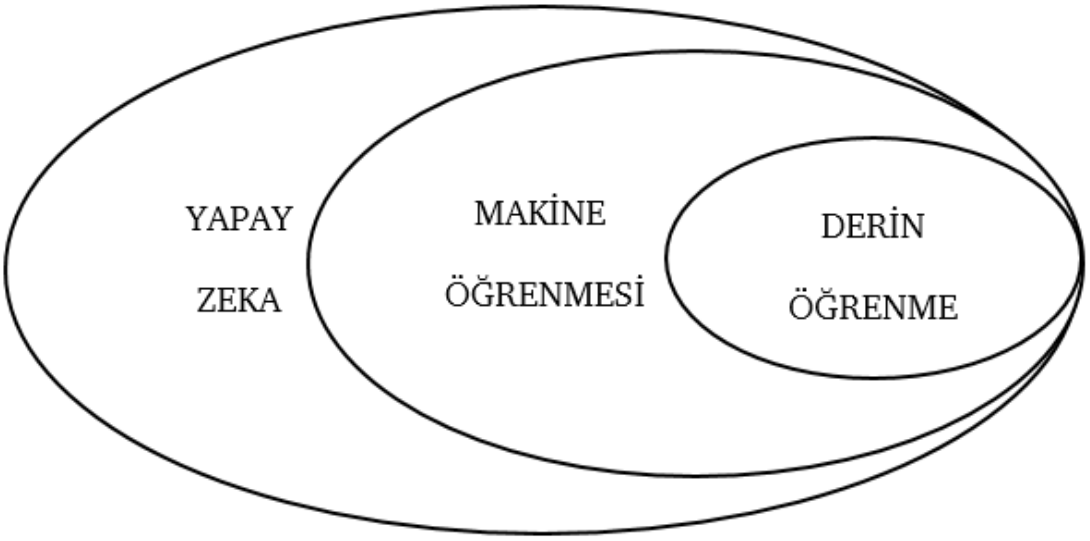
1.3. Hipotez

Çalışma İHA ile elde edilen YTÜ Davutpaşa kampüsüne ait hava fotoğraflarından derin öğrenme yöntemi olan YOLO modelleri kullanılarak araç çıkarımı yapılabilmektedir hipotezi üzerine kurulmuştur.

2
DERİN ÖĞRENME

2.1 Yapay Zeka ve Makine Öğrenmesi Kavramları

Yapay zekâ kavram olarak, 1930'lu yıllarda "makineler düşünebilir mi?" sorusu ile ortaya çıkmış ve 1950'li yıllarda gelişmeye başlamıştır. Bu kavram 1980'lere kadar sadece geliştirilen kuralların izlenmesi üzerine gelişme göstermiştir. Yapay zekâ, insanlar tarafından yerine getirilen düşünsel faaliyetlerin otonom haline getirme olarak tanımlanabilir [4]. Ayrıca, yapay zekâ, akıllı sistemler oluşturmak gibi kesin bir hedefe sahiptir. Ancak bu amaca giden yolda kullanılan yöntemler kesin olarak belirlenmemiştir. Bu sebeple günümüzde yapay zekâ çok sayıda alt disiplinden oluşmaktadır [35].



Şekil 2.1 Derin öğrenmenin yapay zeka kavramı içerisindeki konumu

Yapay zekâ makine öğrenmesi ve derin öğrenmeyi de kapsayan önemli bir teknolojik gelişmedir (Şekil 2.1). Geliştirilen ilk yapay zekâ uygulamaları tamamen kurallara bağımlı kaldığından makine öğrenmesi tekniği olarak kabul edilmemektedir.

Makine öğrenmesinde, bilgisayarların direkt programlama işlemi yapılmadan özellikleri keşfetmesi beklenmektedir. Makine öğrenmesi, programlamanın

mümkün olmadığı ya da çok karmaşık çözümlemelerin gerektiği alanlar için önemli bir yapay zekâ alt dalı olarak ortaya çıkmıştır. Klasik yapay zekâ yapısında sembolik yaklaşımlar kullanılmaktadır. Bu yapıda, girdi olarak kurallar ve veriler ortaya konulur, veriler bu kuralları takip ederek çözüm elde eder. Makine öğrenmesi ise veriyi ve beklenen çıktıları kullanarak yeni verilere özgün cevaplar üretebilecek kuralları öğrenir. Makine öğrenmesinin sıklıkla kullanılan yöntemlerini dört başlıkta vermek mümkündür [4].

- Denetimli Öğrenme
- Denetimsiz Öğrenme
- Yarı Denetimli Öğrenme
- Pekiştirmeli Öğrenme

2.1.1. Denetimli Öğrenme

Bu öğrenme tipi en yaygın kullanılan yöntemdir. Bu yöntemde çıktılar bilinmektedir ve verilen girdiler ile aralarında sağlanan korelasyon sayesinde öğrenme gerçekleşir. Dizi oluşturma, nesne tespiti, görüntü bölütleme (segmentasyon) gibi alanlarda kullanılmaktadır [4].

2.1.2. Denetimsiz Öğrenme

Bu yöntemde verilere bağlı olarak çıktı değerler oluşturulmaktadır. Denetimsiz öğrenme algoritmaları insan müdahalesine gerek olmadan verileri keşfeder. Genellikle denetimli öğrenme öncesi boyut azaltma, kümeleme gibi veriyi anlama amaçlı çözümlemeleri kullanır.

2.1.3. Yarı Denetimli Öğrenme

Denetimli öğrenmede etiketlemenin yapılmadığı veya sınırlı etiketlemenin yapıldığı bir yöntemdir. Etiketlere ilişkin değerleri algoritma keşifsel olarak öğrenir.

2.1.4. Pekiştirmeli öğrenme

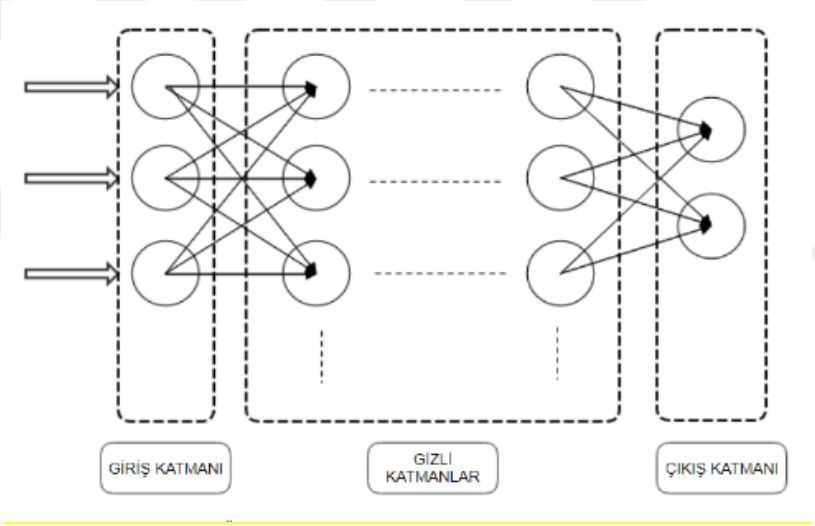
Davranışsal makine öğrenmesi modelidir. Çevresindeki bilgileri toplayarak ödülün en yüksek olduğu hamleyi yapar. Oyun sektöründe sıklıkla kullanılmaktadır.

2.2. Derin Öğrenme

Derin Öğrenme, makine öğrenmesinin alt dalıdır. Verilen bir veri kümesi ile çıktıları tahmin edecek modelin eğitilmesine olanak sağlar. Modeli eğitmek için hem denetimli hem de denetimsiz öğrenme kullanılabilir.

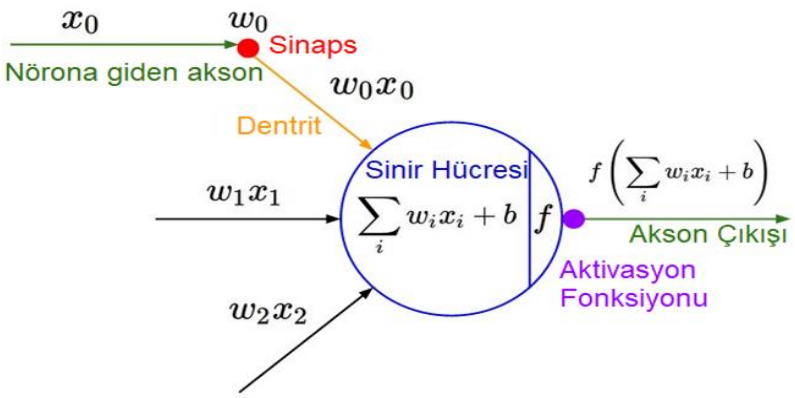
Derin öğrenme ile makine öğrenmesini birbirinden ayıran özellik, kurulan algoritmaların öğrenme şeklidir. Diğer makine öğrenmesi yöntemlerinden farklı olarak, derin öğrenme ağında veri bütün olarak ele alınır böylece verideki ara kayıplar engellenmiş olur.

Derin Öğrenme Sinir Ağı (DLNN) genel olarak bir giriş katmanı, gizli katmanlar ve bir çıkış katmanından oluşur (Şekil 2.2).



Şekil 2.2 Genelleştirilmiş derin sinir ağı yapısı

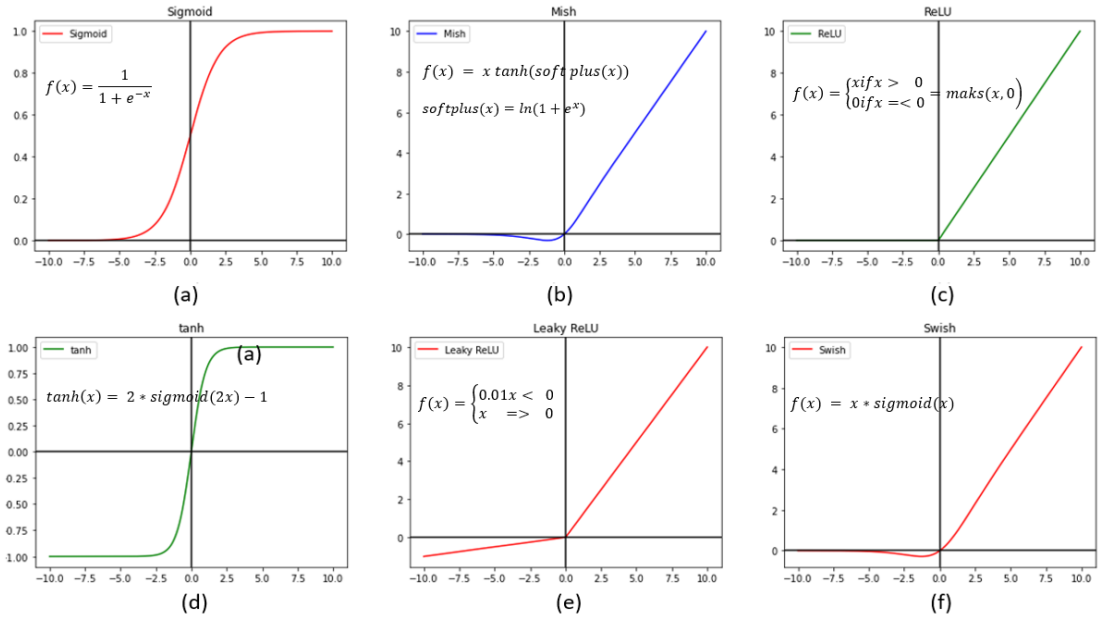
Bir yapay sinir hücresinde, verilerden veya diğer hücrelerden gelen bilgiler (x) ağırlıklar (w) ile çarpılır ve bias (b) değeriyle toplanarak hücreye bağlanır (Şekil 2.3). Hücreye gelen bağlantılar toplanır. Daha sonra bir aktivasyon fonksiyonundan geçirilerek normalize edilir.



Şekil 2.3 Yapay sinir hücresi modeli

2.2.1. Aktivasyon Fonsiyonları

Her yapay sinir hücresi aktivasyon fonksiyonuna sahiptir. Aktivasyon fonksiyonları sinir hücresinin verdiği sonucu normalize eder. Aktivasyon fonksiyonu, derin öğrenmede doğrusal olmayan problemler ele alındığından doğrusal olmayan fonksiyonlardan seçilmektedir. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi alınmaktadır. Türevinin yüksek işlem gerektirdiği fonksiyonlar ağın genel hızına etki etmektedir. Bu sebeple türevi kolay alınan fonksiyonların seçilmesi çok önemlidir. Ayrıca doğrusal fonksiyonların türevi sabit olduğundan öğrenme işlemini gerçekleştirmezler. Modellerde genellikle kullanılan Sigmoid, Hiperbolik Tanjant (tanh), ReLU, Leaky ReLU, Mish ve Swish aktivasyon fonksiyonlarının grafikleri (Şekil 2.4)'te verilmiştir. Bununla birlikte kullanılan farklı aktivasyon fonksiyonları da mevcuttur. Hangisinin kullanılacağına karar verilirken, verinin büyüklüğü, yapısı veya modelin diğer özellikleri göz önünde bulundurulur.



Şekil 2.4 Örnek aktivasyon fonksiyonlarının grafikleri

- **Sigmoid**

Aktivasyon fonksiyonlarından biri olan Sigmoid fonksiyonunda x değerleri arttıkça çıktı sabit bir değere yaklaşır (Şekil 2.4 (a)). Buna karşın Sigmoid fonksiyonu çıkış katmanındaki nöronlar için bir aktivasyon fonksiyonu olarak kullanılmaya devam etmektedir. Sigmoid fonksiyonunu kullanmamızın ana nedeni sıfır ile bir arasında olmasıdır. Bu nedenle olasılığı tahmin etmemiz gereken modeller için oldukça uygundur [36].

- **Hiperbolik Tanjant (tanh)**

Sigmoid fonksiyonuna oldukça benzeyen bu fonksiyon da ikili sınıflandırma için kullanılır. Fakat tanh [-1,1] değer aralığında değerler almaktadır. Tanh fonksiyonu da uç noktalarda türev sıfıra yakınsadığından gradyan kaybolması problemiyle karşılaşır. Çok katmanlı ağlarda tercih edilmemektedir [37].

- **ReLU**

ReLU en yaygın kullanılan aktivasyon fonksiyonudur. ReLU'nun etkinliğinin sebebi türevinin bir sayıya yakınsamamasıdır. Bu sayede öğrenme sürecini önemli ölçüde yavaşlatabilen gradyan kaybolması probleminin ortadan kaldırılmasına yardımcı olmaktadır [36].

- **Leaky ReLU (Sızıntı ReLU)**

ReLU fonksiyonun türevi negatif değerlerde sıfır olmaktadır. Bu durum Geri Yayılım esnasında öğrenme işleminin gerçekleşmeyeceği anlamına gelir. Bu probleme Dying (ölen) ReLU adı verilir. Leaky ReLU, Dying ReLU problemine karşı geliştirilen çözümlerden biridir. Leaky ReLU'da negatif değerler sıfıra çok yakındır fakat tam sıfır değildir (Şekil 2.4). Böylece türevinin sıfır olması engellenmiş olur. Bu durum geri yayılım esnasında öğrenmenin negatif tarafta da gerçekleşeceği anlamına gelir [38].

- **Swish**

Bir diğer aktivasyon fonksiyonu olan Swish, Google tarafından geliştirilmiştir. Girdi değerini sigmoid değeri ile çarparak çıktı üretmektedir. Her yerde türevi vardır. Daha derin katmanlı sinir ağlarında tercih edilebilir [38].

- **Mish**

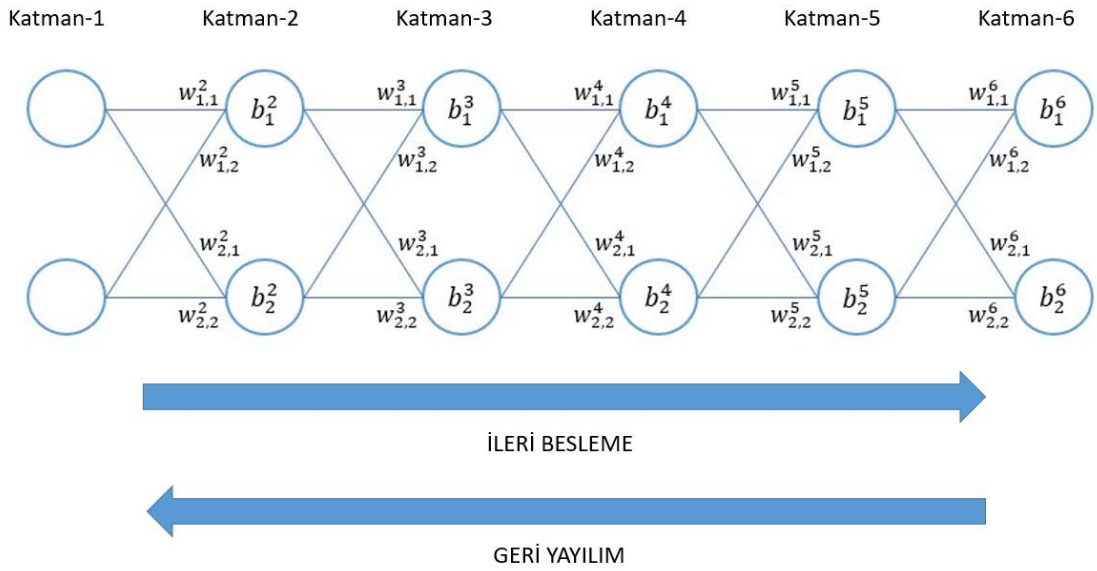
Mish fonksiyonu pozitif değerlerde herhangi bir yüksekliğe çıkabilir yani sınırlama olmaması nedeniyle öğrenme doygunluğunu önler. Ayrıca negatif değerler için Leaky ReLU'da olduğu gibi sıfır sınırına karşı daha iyi gradyan akışına izin vermektedir [39].

2.2.2. Geriye Yayılım (Backpropagation)

Sinir ağlarında çözümlenmesi gereken en önemli konu en uygun ağırlık ve bias değerlerinin bulunmasıdır. Çok katmanlı sinir ağları eğitilirken denetimli bir öğrenme algoritması olan geriye yayılımı kullanmaktadır (Şekil 2.5). Ağlarda eğitime başlamadan önce ağırlık değerleri aktarımlı öğrenme ile veya sezgisel olarak küçük bir değerle başlatılmaktadır. Geri yayılım algoritması, delta kuralı veya gradyan iniş olarak adlandırılan bir teknik kullanarak hata fonksiyonunun minimum değerini bulmayı amaçlar. Bunun için ağırlık değerlerinin en optimum duruma gelmesi gerekmektedir. Algoritma üç adımdan oluşmaktadır.

- İleri Besleme
- Geri Yayılım

- Tüm değerleri bir araya getirmek ve güncellenmiş ağırlık değerini hesaplamak [40].



Şekil 2.5 Geri beslemeli ağ yapısı [41]

2.2.3. Kayıp Fonksiyonları

Derin öğrenmenin en zor kısımlarından biri yapay sinir ağını eğitmektir. Ağın eğitimi için büyük veri kümeleri kullanıldığından derin ağlar çok fazla miktarda hesaplama gücüne ihtiyaç duyar. Eğitim aşamasında tahmin değerleri referans değerleri ile karşılaştırılmaktadır ve oluşan farklar kayıp değeri olarak adlandırılmaktadır. Bu değer eğitim devam ettikçe düşmektedir. Kayıp fonksiyonu çıktısının sıfır olması ağın çıktılarıyla girdileri arasında fark olmadığını göstermektedir.

$$L(w,b) = \frac{1}{2}(\hat{y} - y)^2 \qquad 2.1$$

Burada y referans değeri ve $\hat{y}$ tahmin değeri çıktısıdır.

- **Çapraz Entropi Kaybı**

İkili sınıflandırma problemleri için kullanılır. 0 ile 1 arasında bir olasılık değeri olan bir sınıflandırma modelinin performansını ölçer. Bu makine öğrenmesi modeline örnek olarak lojistik regresyon verilebilir. Tahmin edilen olasılık asıl değerden uzaklaştıkça çapraz entropi kaybı artar [42].

- **Ortalama Karesel Hata (Mean Squared Error-MSE)**

Ortalama karesel hata, tahmin edilen ve referans değerler arasındaki kare farklarının ortalaması olarak hesaplanır [43]. Genellikle regresyon problemlerinde, yani sürekli değerleri tahmin etmede kullanılır.

2.2.4. Optimizasyon Algoritmaları

Sinir ağında en önemli işlem ağırlıkların belirlenmesi ve kayıp değerlerinin en aza indirilmesidir. Kayıp fonksiyonunu minimize etmek için en uygun ağırlık 'W' değerleri bulunmalıdır. Bunun için genellikle gradyan inişi algoritmalarından faydalanılmaktadır. Bu algoritmalar (RMSProp ve ADAM) ağırlıkları, öğrenme hızları veya momentumu kullanarak güncellemektedir [44].

$$L(w) = \frac{1}{N} Li(w) \qquad 2.2$$

W ağın tüm ağırlıklarını depolayan sütun vektörü olsun. Kayıp fonksiyonunu L (w) en aza indirgemek amacıyla Formül 2.2'deki gibi yazılabilmektedir. Formülde verilen N, veri örneği sayısı ve Li, veri örneği i ile ilişkili kayıp fonksiyonudur.

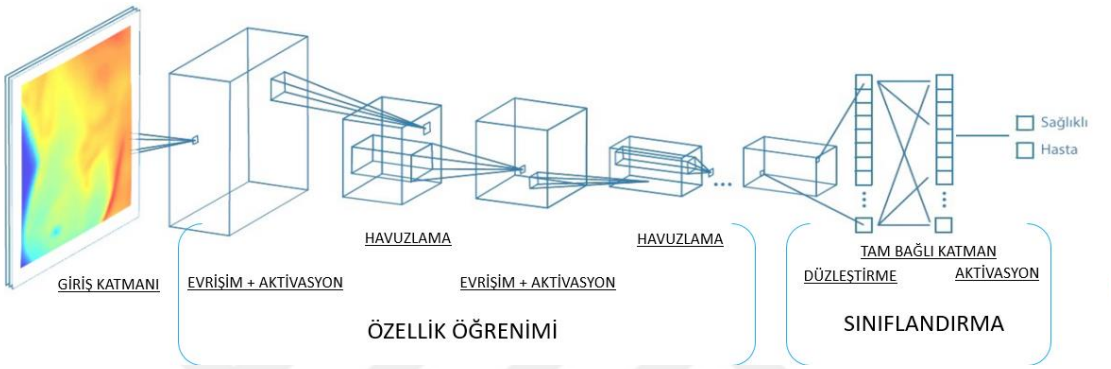
2.2.5. Derin Öğrenme Mimarileri

Derin öğrenme mimarileri oluşturulurken, yapay sinir ağlarında (YSA) ihtiyaçlara göre katmanlar ve bağlantılarda yapılan değişiklikler ile oluşturulmuş farklı mimariler bulunmaktadır. Sınıflandırma, tespit, tanı, tahmin, teşhis vb. sonuçların üretilmesi için farklı sinir ağı modelleri tercih edilir[45].

- Evrişimsel Sinir Ağları (Convolution Neural Network-CNN)
- Tekrarlayan Sinir Ağları (Recurrent Neural Network-RNN)
- Uzun-Kısa Süreli Hafıza (Long Short-Term Memory-LSTM)
- Kısıtlı Boltzmann Makineleri (Restricted Boltzmann Machines-RBM)
- Derin İnanç Ağları (Deep Belief Nets-DBN)
- Derin Oto-Kodlayıcılar (Auto Encoder)

2.3. Evrişimli Sinir Ağları (Convolution Neural Network-CNN)

Derin öğrenmede bilgisayarlı görü gerektiren problemlerde genellikle evrişimli sinir ağları kullanılmaktadır. Temel olarak CNN mimarileri sınıflandırma sorununun çözümü için standart YSA kullanır. Bilgileri belirlemek ve bazı özellikleri tespit etmek için diğer katmanlardan yararlanır [46]. Bu katmanlar giriş katmanı, özelliklerin öğrenildiği evrişim katmanları ve sınıflandırmanın yapıldığı tam bağlı katmanlardır (Şekil 2.6).



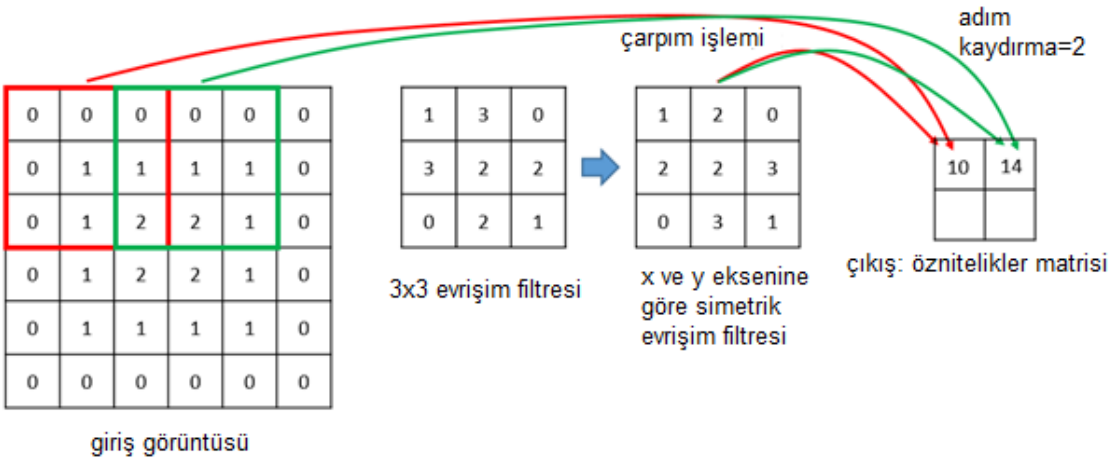
Şekil 2.6 Genel CNN yapısı [46'dan uyarlanmıştır]

- **Giriş Katmanı**

Görüntülerin evrişimli sinir ağına dahil edildiği katmandır.

- **Evrişim Katmanı**

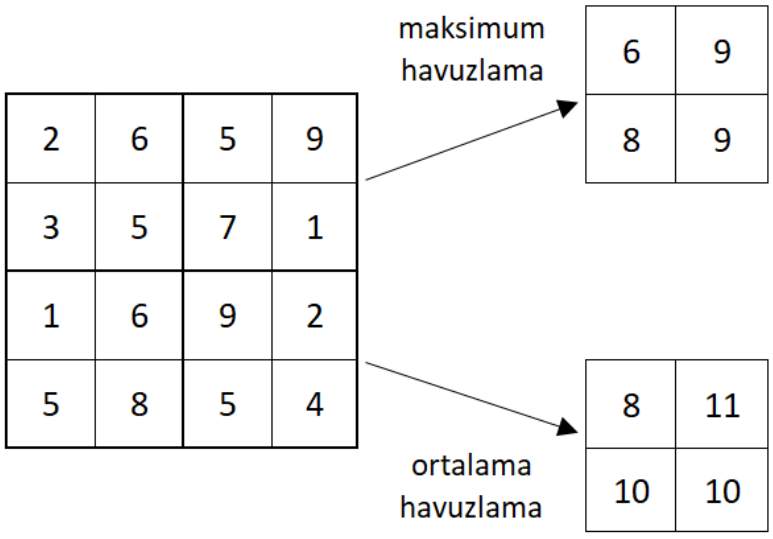
Evrişim katmanında amaç özellikleri öğrenmektir. Bu katmanda çeşitli özellikleri tespit etmek için oluşturulmuş filtre matrisleri görüntü üzerinde ilerleyerek bu özelliklerin tespit edilmesini sağlarlar (Şekil 2.7). CNN algoritmalarında öğrenilen parametreler bu filtrelerdeki değerlerdir. Model sürekli olarak bu değerleri günceller ve özellikleri daha da iyi tespit etmeye başlar. Bazı bilinen filtrelerle görseller keskinleştirilebilir veya kenar tespiti yapılabilir.



Şekil 2.7 Evrişim işlemi ve adım kaydırma [47]

- **Havuzlama (Pooling) Katmanı**

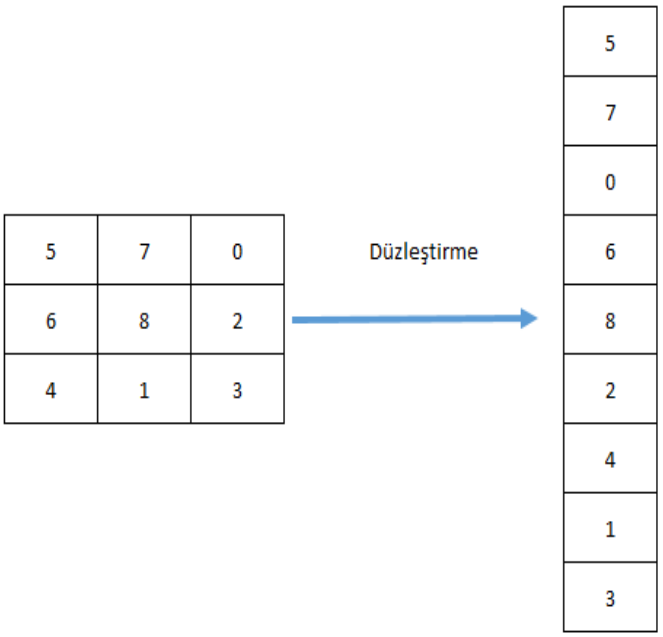
Havuzlama katmanı görüntü boyutunun azaltılmasını sağlar. Katman bunu yaparken bazı özellikleri elemiş olur. Girdi görüntüsünün boyutunun azalması işlem hızını arttırmaktadır. Havuzlama işlemi değişik şekillerde yapılabilmektedir. (Şekil 2.8)'de havuzlama yöntemleri örneklenmiştir. Maksimum havuzlama yönteminde belirlenen filtre içerisinde bulunan en yüksek değerli piksel alınmaktadır. Ortalama havuzlamada ise bu değerlerin ortalaması alınır.



Şekil 2.8 Havuzlama İşlemi

- **Düzleştirme (Flatten) Katmanı**

Tam bağlı katmanda piksel matrisleri tek boyutlu hale getirilir (Şekil 2.9).



Şekil 2.9 Düzleştirme (flatten) İşlemi

- **Tam Bağlı Katman.**

Sınıflandırma işlemi bu katmanda gerçekleşmektedir [46].

2.3.1. Hiperparametreler

Teknolojik gelişmelere paralel olarak bilgisayarların artan işlem gücü araştırmacıların daha fazla katmandan oluşan daha karmaşık yapılar tasarlanabilmesine imkân vermiştir. Çok katmanlı ağ yapısını en iyi şekilde tasarlamak ve modelin ihtiyaçlara cevap vermesini sağlamak amacı ile modeli tasarlayanın sezgisine ve deneyimine bağlı olarak seçilen değişkenlere hiperparametre denilmektedir.

- **Fitre Boyutu**

Evrişim katmanındaki filtrelerin boyutu öğrenme üzerinde oldukça etkili bir hiperparametredir. Filtre boyutu ile ne kadar genişlikte verinin birbirini etkileyeceğine karar verilmektedir. Genelde 3x3, 5x5, 7x7 boyutta filtreler kullanılmaktadır [48].

- **Kaydırma Değeri (Stride)**

Bu değer filtrenin ana görsel üzerinde kaç piksel boyunca kayacağını belirler.

- **Veri setinin Boyutu**

Derin öğrenme uygulamalarında oluşturulan veri setinin tüm veriyi temsil yeteneği, öğrenmeyi doğrudan etkilemektedir. Bu sebeple verinin sayısı ve çeşitliliği veri toplama aşamasında dikkatle incelenmesi gereken bir konudur. Veri setinin artışı ile model doğruluğu sürekli artmamaktadır. Bir noktada artış eğrisi durağanlık kazanacaktır. Verinin sayıca artması bir noktada gereksiz işlem yüküne sebep olmaktadır.

- **Öğrenme Hızı ve Momentum Katsayısı**

Derin öğrenmede ağırlık değerlerinin güncellenmesi geri yayılım algoritması ile yapılmaktadır. Bu güncelleme için optimizasyon algoritması seçimine bağlı olarak sabit bir değer olan öğrenme hızı ve momentum katsayısı kullanılır.

- **Küme Boyutu**

Model eğitilirken veriler belli sayıda parçalar halinde eğitime katılırlar. Küme boyutu modelin aynı anda kaç veri işleyeceğine ilişkin bir hiperparametredir [48].

- **Eğitim Tur (Epoch) Sayısı**

Modelde, ilk küme eğitilir, modelin başarısı test edilir, daha sonra buna bağlı olarak geri yayılım ile ağırlıklar güncellenir [48]. İşlemler tekrarlanarak en uygun ağırlık değerleri hesaplanmaya çalışılır. Bu eğitim adımlarının her birine eğitim tur sayısı denilmektedir.

- **Ağırlık (Weight) Başlangıç Değerlerinin Belirlenmesi**

Modelin ilk başlangıç ağırlık değerleri bazı kurallara bağlı olarak rastgele belirlenmektedir. Aktarımlı öğrenme ile ise modeller daha uygun ağırlıklar ile başlayacağından optimum ağırlıklara ulaşması kısa sürecektir.

- **Katman Sayısı**

Derin öğrenmenin karmaşık problemlerde iyi sonuç vermesini sağlayan en önemli özellik katman sayısıdır. Katman sayısı arttıkça modelin daha iyi öğrendiği

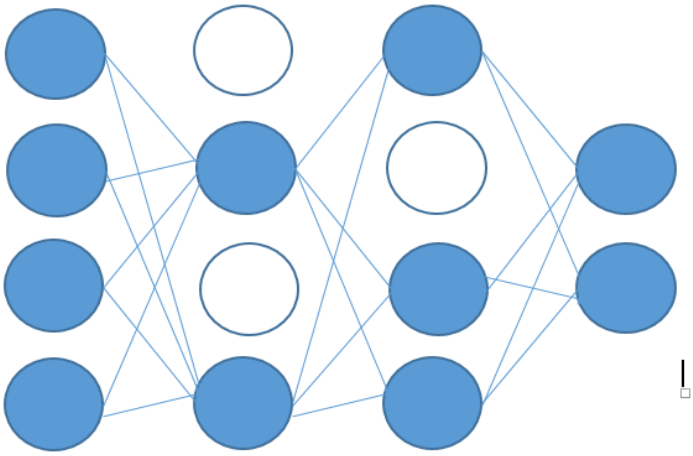
gözlenmiştir [48]. Ancak, katman sayısının artırılmasının eğitimdeki etkisi gittikçe durağan olmaktadır.

- **Gizli Katmanlardaki Nöron Sayısı**

Nöronlar hafızada tutulan bilgi sayısını ifade eder [48]. Nöron sayısının az olması özellik kaybına sebep olurken çok olması da yüksek bellek gerektirmektedir.

- **Seyreltme (Dropout)**

Bir sinir ağında, az veriyle, uzun süreli eğitim yapıldığında ağ aşırı öğrenme denilen ezberleme (overfitting) sorunuyla karşılaşır. Ezberlemeyi engellemek amacıyla ağ içerisinde bulunan bağlantılar kaldırılır (Şekil 2.10). Bu yöntem ağın performansına katkı sağlamaktadır.



Şekil 2.10 Seyreltme örneği

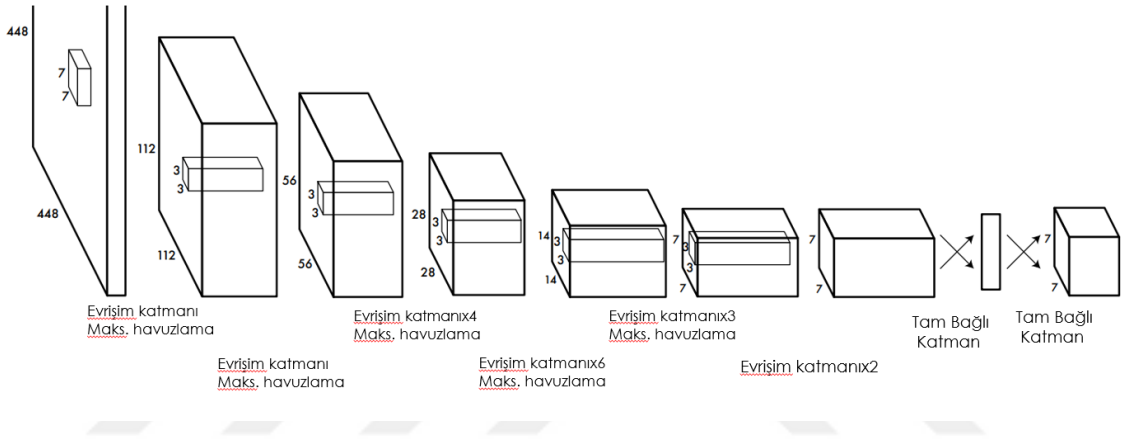
2.4. YOLO (You Only Look Once) Mimarisi

İlk derin öğrenme çalışmaları karmaşık hatlar üzerinde yapılmaktaydı. Bu karmaşık hatları optimize etmek hem çok zor hem de yüksek performans gerektirmektedir. R-CNN gibi bölge bazlı nesne tespit algoritmaları, önce nesnelerin bulunmasının muhtemel olduğu alanları belirler ardından o alanlarda ayrı ayrı CNN (Evrişimsel Sinir Ağları) sınıflandırıcıları yürütür. Bu yöntemler doğruluk açısından iyi sonuçlara ulaşsa dahi, görüntüler iki ayrı işleme tabi tutulduğu için işlem sayısı artmakta ve düşük fps alınmasına sebep olmaktadır.

YOLO mimarisi gerçek zamanlı olarak çalışma yapısı nedeniyle algılama hızını artırır. Bu mimarinin diğer nesne algılama mimarilerinden daha hızlı olmasının sebebi görüntünün tamamını tek seferde nöral bir ağdan geçiriyor olmasıdır.

2.4.1. YOLOv1

Nesne tespiti amacı ile YOLO yaklaşımı ilk olarak 2016 yılında Joseph Redmon ve arkadaşları tarafından tanıtılmıştır [49]. Tanıtılan YOLO derin öğrenme modeli, 24 adet konvolüsyonel katmana ve ardından 2 adet tam bağlantılı katmana sahiptir (Şekil 2.11).



Şekil 2.11 YOLOv1 yapısı [49'dan uyarlanmıştır]

2.4.2. YOLOv2

Daha sonra YOLOv2 Joseph Redmon ve arkadaşları tarafından bazı iyileştirmeler yapılarak tanıtılmıştır [50]. Bu modelde tüm evrişim katmanlarına toplu normalleştirme katmanları eklenmiştir. Bu yapılan işlemin mAP değerini %2'den fazla arttırdığı gözlenmiştir. YOLOv1, sınırlayıcı kutu (bounding box) koordinatlarını doğrudan tahmin ederken, YOLOv2 sınırlayıcı kutu önceliklerini tahmin ofsetleri ile değiştirmiştir. Bu durum ortalama hassasiyeti %0,3 düşürmesine karşın duyarlılıkta %81'den %88'e bir artış ile modelin iyileştirilmesi için daha fazla alana sahip olduğunu kanıtlamıştır [50]. YOLOv2, Darknet-19 adlı yeni bir model kullanır. Darknet-19, 19 evrişim katmanına ve 5 tane maksimum havuzlama katmanına sahiptir (Şekil 2.12). Darknet-19 daha az işlem gerektirerek YOLOv2'yi, YOLOv1'den daha hızlı hale getirmektedir [50].

KATMAN TİPİ	FİLTRE	BOYUT	ÇIKTI
Evrişim	32	3 x 3	224 x 224
Maksimum Havuzlama		2 x 2/2	112 x 112
Evrişim	64	3 x 3	112 x 112
Maksimum Havuzlama		2 x 2/2	56 x 56
Evrişim	128	3 x 3	56 x 56
Evrişim	64	1 x 1	56 x 56
Evrişim	128	3 x 3	56 x 56
Maksimum Havuzlama		2 x 2/2	28 x 28
Evrişim	256	3 x 3	28 x 28
Evrişim	12	1 x 1	28 x 28
Evrişim	256	3 x 3	28 x 28
Maksimum Havuzlama		2 x 2/2	14 x 14
Evrişim	512	3 x 3	14 x 14
Evrişim	256	1 x 1	14 x 14
Evrişim	512	3 x 3	14 x 14
Evrişim	256	1 x 1	14 x 14
Evrişim	512	3 x 3	14 x 14
Maksimum Havuzlama		2 x 2/2	7 x 7
Evrişim	1024	3 x 3	7 x 7
Evrişim	512	1 x 1	7 x 7
Evrişim	1024	3 x 3	7 x 7
Evrişim	512	1 x 1	7 x 7
Evrişim	1024	3 x 3	7 x 7
Evrişim	1000	1 x 1	7 x 7
Ortama Havuzlama		Global	1000
Softmax			

Şekil 2.12 Darknet-19 [50' den uyarlanmıştır]

2.4.3. YOLOv3

Mayıs 2018'de Redmon ve arkadaşları Darknet-19 modelini geliştirerek daha fazla evrişim katmanı ve artık (residual) katmanlar eklediler [51]. Darknet-19'da olduğu gibi model toplam evrişim katmanı sayısı ile Darknet-53 olarak isimlendirildi (Şekil 2.13). Darknet-53 modeli işlem yükünü artırsa da Darknet-19 modeline göre çok daha iyi doğruluk performansı göstermektedir [51].

	KATMAN TİPİ	FİLTRE	BOYUT	ÇIKTI
	Evrişim	32	3 x 3	256 x 256
	Evrişim	64	3 x 3/2	128 x 128
1 x	Evrişim	32	1 x 1	
	Evrişim	64	3 x 3	
	Artık			128 x 128
	Evrişim	128	3 x 3/2	64 x 64
2 x	Evrişim	64	1 x 1	
	Evrişim	128	3 x 3	
	Artık			64 x 64
	Evrişim	256	3 x 3/2	32 x 32
8 x	Evrişim	128	1 x 1	
	Evrişim	256	3 x 3	
	Artık			32 x 32
	Evrişim	512	3 x 3/2	16 x 16
8 x	Evrişim	256	1 x 1	
	Evrişim	512	3 x 3	
	Artık			16 x 16
	Evrişim	1024	3 x 3/2	8 x 8
4 x	Evrişim	512	1 x 1	
	Evrişim	1024	3 x 3	
	Artık			8 x 8
	Ortama Havuzlama		Global	
	Tam Bağlı		1000	
	Softmax			

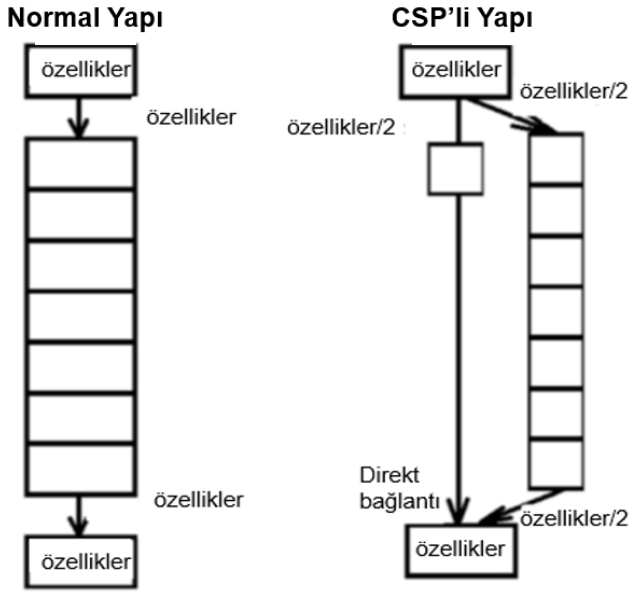
Şekil 2.13 Darknet-53 [51'den uyarlanmıştır]

2.4.4. YOLOv4

YOLO modelini versiyon üçe kadar Joseph Redmon geliştirmiştir. YOLOv4, 2020 yılında CSP (Cross Stage Partial) yapısı (Şekil 2.14) ile Alexey Bochkovskiy ve arkadaşları tarafından "Nesne Tespitinde Optimal Hız ve Doğruluk" makalesiyle yayınlanmıştır [52]. YOLOv4 modeli; omurga, boyun ve baş şeklinde yapılanmaktadır.

Omurga, özellik çıkarımının (feature extraction) yapıldığı katmandır. YOLOv4'te CSPDarknet53 yapısı kullanılmaktadır. Boyun, omurga ve kafa arasında yer alan katmandır. Nesne tahmini yapılırken daha fazla özellik bulmak için ara katman olarak eklenmiştir. YOLOv4'te, FPN (Feature Pyramid Network) yerine SAM (

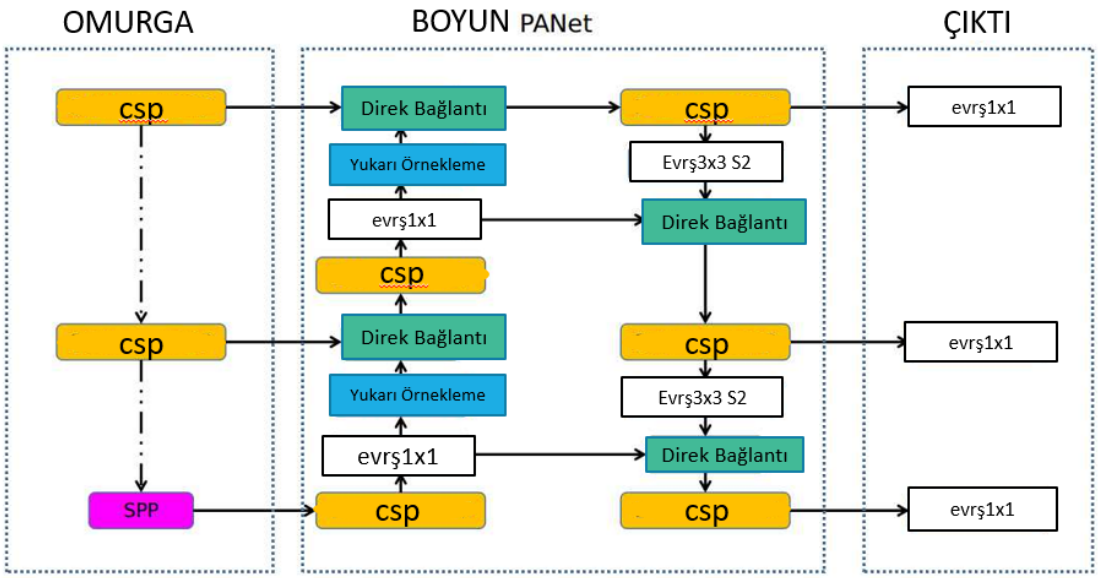
Spatial Attention Module), PAN (Path Aggregation Network) ve SPP (Spatial Pyramid Pooling) kullanılır. Baş kısmında sınırlayıcı kutular bulunur ve her kutunun sınıfı tahmin edilir. Burada YOLOv3'te uygulanan modeller kullanılır [52].



Şekil 2.14 CSP yapısı modeli

2.4.5. YOLOv5

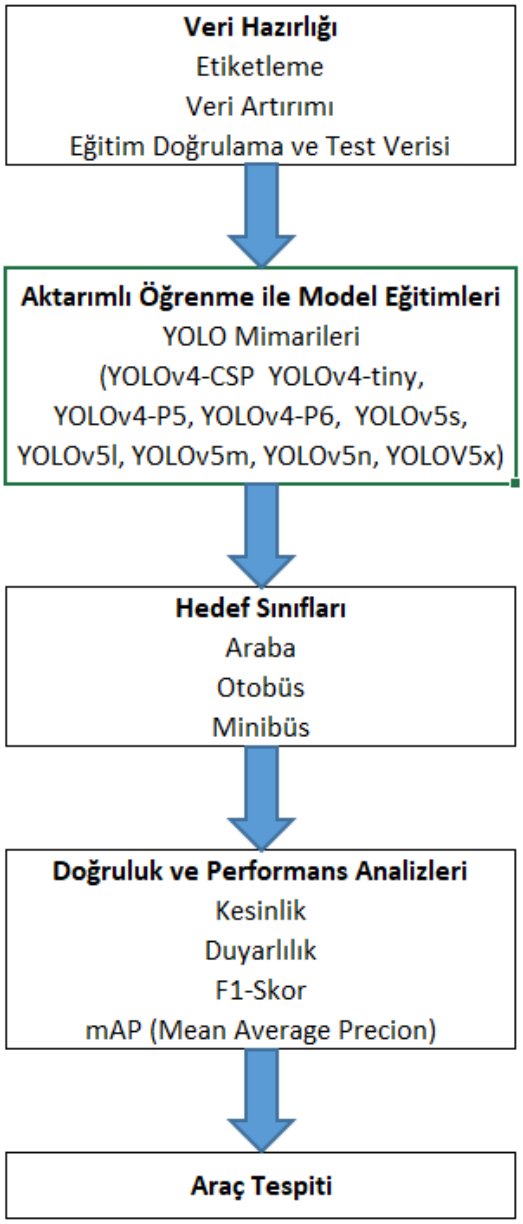
YOLOv5, YOLOv4'ten kısa bir süre sonra Glenn Jocher tarafından kapsamlı bir araştırma ve karşılaştırma yapılmadan ortaya atılmıştır. Modele ait kodlar Ultralytics LLC'nin deposundadır. Kamu lisansına sahip olmakla birlikte iOS işletim sisteminde uygulaması bulunmaktadır. Şekil 2.15'de YOLOv5'in genelleştirilmiş yapısı verilmiştir. YOLOv5, diğer YOLO versiyonlarından farklı olarak bir PyThorch uygulamasıdır. YOLOv4'teki gibi CSPNet omurga (backbone) ve PA-NET boyun (neck) kullanılır. Baş (head) kısmında ise YOLOv4'te kullanılan model kullanılır [53].



Şekil 2.15 YOLOv5 genel yapı [54'den uyarlanmıştır]

3
YÖNTEM

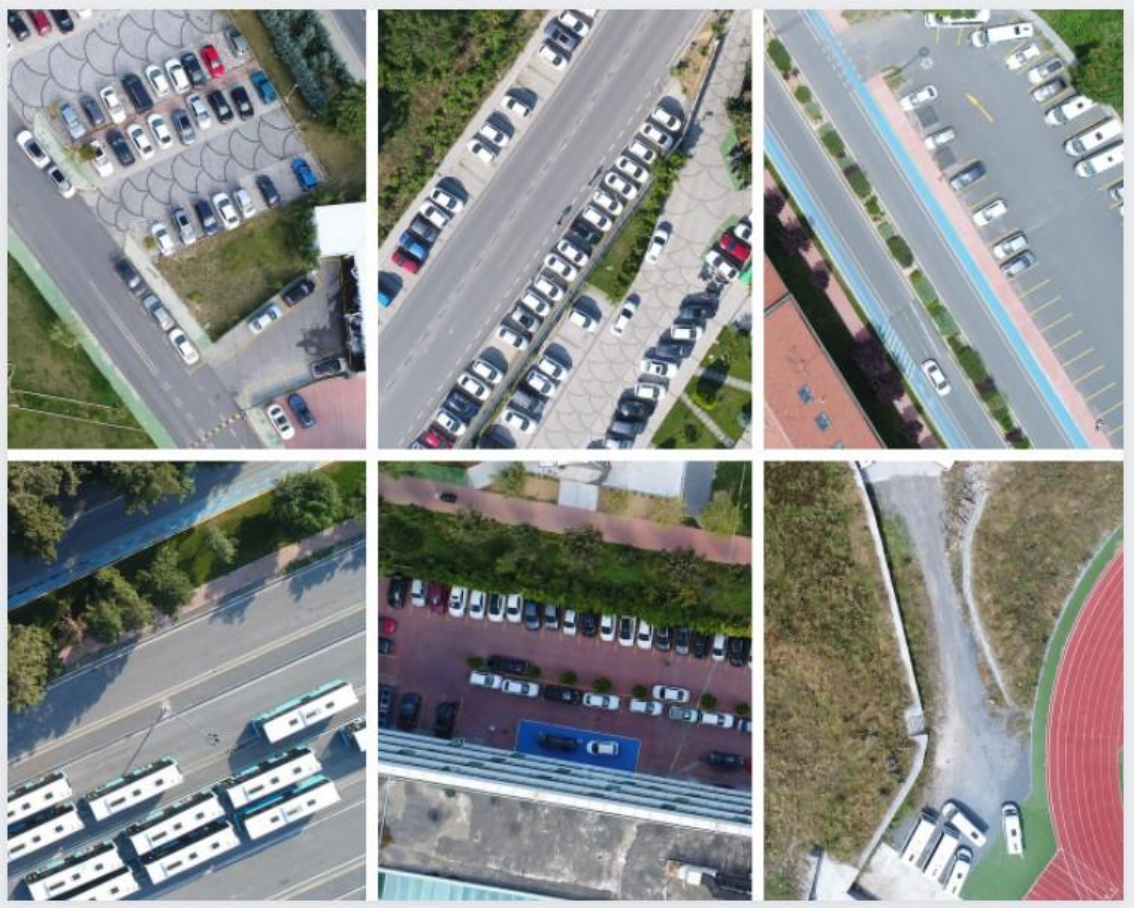
Bu bölüm, veri etiketleme ve veri artırımından oluşan veri hazırlığı adımları, seçilen YOLO mimarilerinin aktarımlı öğrenme kullanılarak eğitilmesi, doğruluk ve performans analizleri ve araç tespiti aşamalarından oluşmaktadır.



Şekil 3.1 İş Akışı

3.1. Veri Seti

Araç tespiti çalışması için 2018 yılında DJI Phantom marka kamera ile İHA ile elde edilmiş 5472*3648 piksel boyutunda, 72 dpi çözünürlüğe sahip YTÜ kampüs içi hava fotoğrafları kullanılmıştır. Araçların yoğun olarak bulunduğu 94 adet fotoğraf seçilmiş ve modele entegre edilebilmesi amacıyla 2048*2048 boyutlarına düzenlenmiştir. Şekil 3.2'de elde edilen veriye ilişkin görseller yer almaktadır.



Şekil 3.2 Veri setinden örnekler

3.2. Veri Etiketleme

Derin öğrenme modellerini oluşturmak için büyük miktarda etiketlenmiş eğitim verisine ihtiyaç duyulmaktadır. Veri etiketleme temel olarak görüntü verisi, yazı verisi veya ses verisinin algılanan özelliklerle işaretlenmesi ve anlamlandırılmasıdır. Yapay zekâ tabanlı modellerde istenen çıktının üretilmesi, büyük oranda doğru ve iyi etiketlenmiş verilere bağımlıdır. Meta veriler görüntülere tanımlayıcı anahtar sözcükler biçiminde atanır. Etiketleme

aşamasında dikkat edilmesi gereken özellikler [57]'ye göre aşağıda listelendiği gibidir.

- Etiketleme çalışmasını yapan kişinin veriyle ilgili bilgisi olması
- Etiketlenen verilerin gerçeğe uygun, doğru ve tutarlı olması
- Herhangi bir kategoriye ait olmayan içeriklerin etiketlenmemesi
- Verilerde, gerekli sayıda etiketleme yapılması
- Her bir etikete ait verilerin yeterince çeşitli olması

3.3. Veri Artırımı

Derin evrişimli sinir ağları, birçok bilgisayarlı görü çalışmasında oldukça iyi performans göstermiştir. Bununla birlikte bu ağlar, ezberlemeden kaçınmak için büyük verilere bağımlıdır [55]. Büyük verilere ulaşılamayan durumlarda model, eğitilen sete karşı çok iyi performans gösterse de, dışarıdan dahil edilen veri üzerinde aynı performans görülememektedir. Bu sorun, ezberleme denilen durumu göstermektedir. Veri artırmanın etkinliğini gösteren ilk çalışmalar, yatay çevirme, renk uzayı büyütmeleri ve rastgele kırpma gibi basit dönüşümlerden gelir [55]. Derin öğrenme yaklaşımlarında yoğunlukla geometrik dönüşümler, renk dönüşümleri, görüntüleri karıştırma, rastgele silme olarak veri artırımı teknikleri uygulanmaktadır.

Çalışmalarda bu yöntemlerden bir kısmı veya tamamı ana veri setine uygulanmakta ve veri setinin temsil yeteneğine katkıda bulunmaktadır. Şekil 3.3'te veri artırımına ilişkin bir görsel bulunmaktadır.



Şekil 3.3 Veri artırımı örnekleri [56]

3.4. Aktarımlı Öğrenme

Daha önce başka bir problem için eğitilmiş modelin yeni bir problem üzerinde yeniden kullanılması aktarımlı öğrenme (Transfer Learning) olarak bilinir. Karmaşık modelleri eğitmek için veri gerçek durum temsil edecek nitelikte ve çoklukta olmalıdır. Ancak çoğu zaman bu veri kümesini elde edebilecek vakit ve uzmanlığa sahip olunmadığından daha önce eğitilmiş veri kümelerinden faydalanılır. Milyonlarca veriye duyulan ihtiyacı azaltmasından dolayı veri bilimi açısından aktarımlı öğrenme değerlidir [58]. Aktarımlı öğrenmeyi kullanırken, aktarılan veriler, modelle benzer olduğunda daha iyi sonuçlar elde edilebilmektedir.

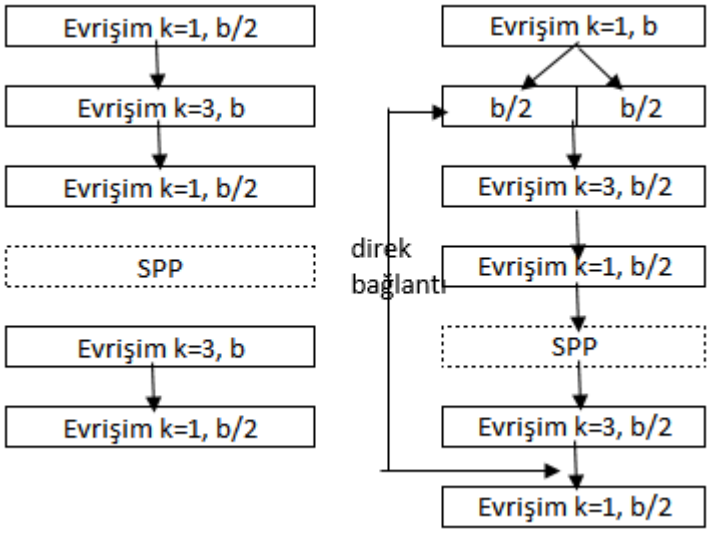
Evrişim katmanlarından bazıları benzer türde bir özellik öğrenirse, bu katmanlarda öğrenilen filtreler farklı evrişim katmanları tarafından yeniden kullanılabilir. Buna, CNN'de transfer öğrenimi denir. Ağırlıklar modele başlangıç ağırlıkları olarak dahil edilirler bu sayede gereksiz ilişkilendirmelere engel olunarak işlem yükü azaltılabilmektedir. Derin öğrenme yöntemlerini sıfırdan

eğitmek yerine aktarımlı öğrenme kullanmak, daha az veri ile daha az eğitim süresinde daha iyi performans elde edilmesini sağlar [59].

3.5. Kullanılan YOLO Mimarileri

3.5.1. Ölçeklendirilmiş YOLOv4

Ölçeklendirilmiş (Scaled) YOLOv4 Wang ve arkadaşları tarafından 2021 yılında tanıtılmıştır. CSP yaklaşımına dayalı YOLOv4 nesne algılama sinir ağının hem yukarı hem de aşağı ölçeklendiği halidir. Model optimum hız ve doğruluğu korurken bütün ağ yapılarına uyarlanabilir. YOLO mimarisine eklenen CSP katmanı (Şekil 3.4) basit bir yapıya sahip olmakla birlikte oldukça verimli sonuçlar sunar. CSP yapısında modelin yarısı hat boyunca ilerlerken semantik bilgiyi üretir, diğer yarısı modele sonradan eklenerek hem mekânsal bilgiyi, hem de bazı semantik özellikleri korur. Ölçeklendirilmiş YOLOv4 modelinin ölçekleri arasında sadece ağın derinliğini, genişliğini, çözünürlüğü değil, aynı zamanda yapısıda değişir [60].

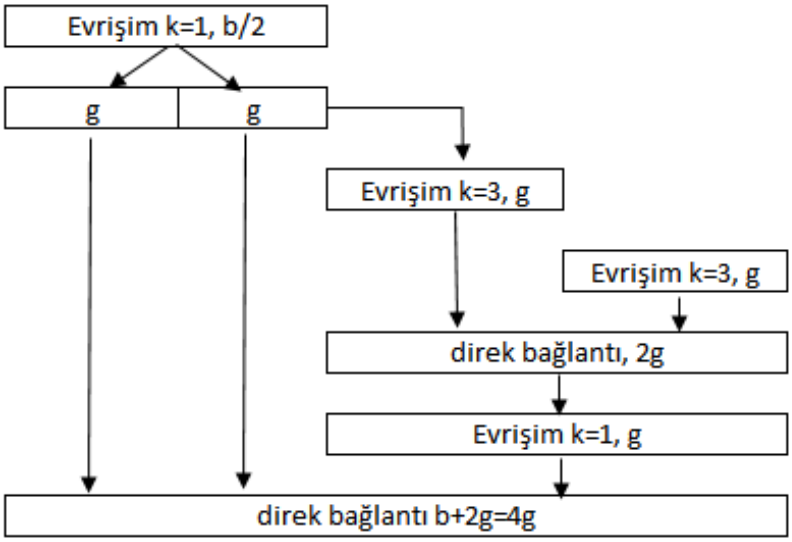


Şekil 3.4 Tüm ağlara uygulanabilecek cspdark modeli [60'dan uyarlanmıştır]

- **YOLOv4-tiny**

Düşük GPU'lar için geliştirilmiş bir modeldir. YOLOv4-tiny modeli (Şekil 3.5), Ölçeklendirilmiş YOLOv4 modelinden farklı düşüncelere sahiptir çünkü bellek bant genişliği ve bellek erişimi gibi farklı kısıtlamalar devreye girmektedir.

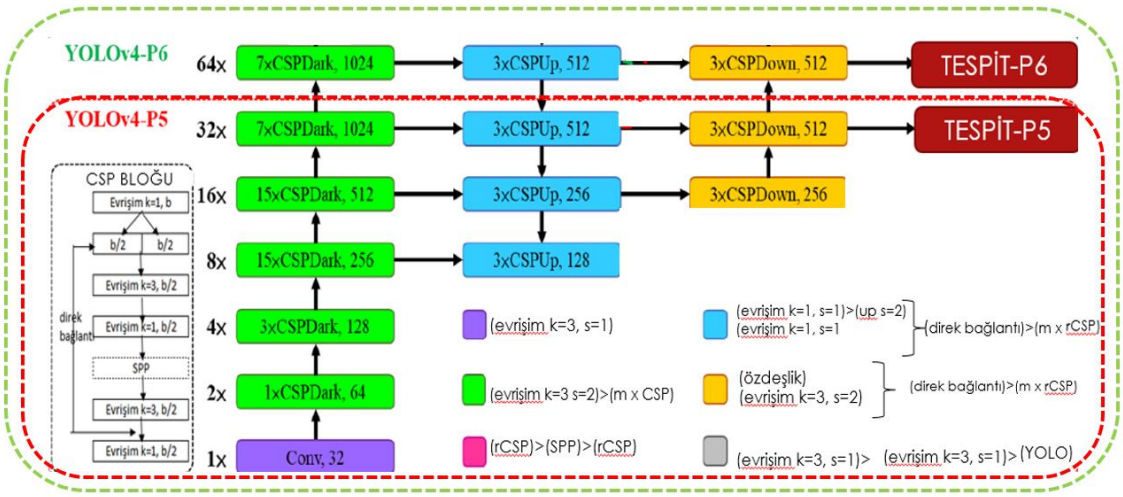
YOLOv4-tiny modelinin omurga yapısı küçük derinlikte elverişli hesaplama karmaşıklığı sunan OSANet'i kullanmaktadır. Sonraki aşamalar için YOLOv3 tasarımı takip edilmiştir [60].



Şekil 3.5 YOLOv4-tiny hesaplama katmanı [60'dan uyarlanmıştır]

- **YOLOv4-Large**

Bu model yüksek GPU'lar için geliştirilmiştir. Bu çalışmada modelin P5 ve P6 ölçekleri kullanılmıştır. Ölçek arttıkça yapının derinliği ve genişliği karmaşıklığı artmaktadır (Şekil 3.6). Eklenen katmanların girişinde her aşamada derinlik ölçeği P5 ve P6 modellerinin ikisinde de 7 olarak verilmektedir. Ayrıca ek genişlik ölçekleme gerçekleştirmek için çıkarım süresi kısıtlama olarak kullanılmaktadır. YOLOv4-P6 modelinde genişlik ölçekleme faktörü 1'e eşit olduğunda 30 fps videoda gerçek zamanlı performansa ulaşabilirler [60].



Şekil 3. 6 Scaled YOLOv4 katman yapıları [60'dan uyarlanmıştır]

3.5.2. YOLOv5 Ölçekleri

YOLOv5'in genel yapısına (Şekil 2.15) bağlı olarak ölçeklendirilen YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x modelleri çalışmada analiz edilmiştir (Şekil 3.7).



Şekil 3.7 YOLOv5 ölçekleri

Ölçeklendirmede en büyük versiyon ile en küçük versiyonun katmanlarının yapısı farklılık göstermez. Aralarındaki fark, ağın genişliği ve derinliğinin ölçekleme çarpanlarıdır. Tablo 3.1'de versiyonlara ilişkin katman sayısı bilgisi, parametre sayıları ve hız özellikleri bilgileri verilmiştir.

Tablo 3.1 YOLOv5 modellerinin özellikleri [61]

Model	Katman Sayısı	Parametre Sayısı	FLOPS
YOLOv5n	270	1.767.976	4.2
YOLOv5s	270	7.027.720	15.9
YOLOv5m	369	20.879.400	48.1
YOLOv5l	468	46.149.064	108.0
YOLOv5x	567	86.231.272	204.2

3.6. Doğruluk Analizi Ölçütleri

Bilgisayarlı görü yapılan YOLO gibi nesne tespiti modellerinde değerlendirme yapmak için mAP ve F1-Skor metrikleri kullanılmaktadır. Doğruluk ölçütü hesaplamalarında referans değer olan etiketlenmiş sınırlayıcı kutu değerleri ile tahmin değerleri karşılaştırılır (Tablo 3.2). Tahminlerin referans değerlere uygunluğu, doğruluğu arttırmaktadır.

Tablo 3.2 Hata Matrisi (Confusion Matrix)

HATA MATRİSİ		**Referans Durum**	
		Pozitif	**Negatif**
Tahmin	**Pozitif**	DP (Doğru Pozitif)	YP (Yanlış Pozitif)
	Negatif	YN (Yanlış Negatif)	DN (Doğru Negatif

Tabloda (DP) doğru pozitif ve (DN) doğru negatif olarak verilen değerler modelin doğru tahmin edildiğini, (YP) yanlış pozitif (YN) yanlış negatif olarak verilenler modelin hatalı tahmin edildiğini göstermektedir.

- **Kesinlik (Precision)**

Bu değer doğru olarak tahmin edilen değerlerin kaç tanesinin doğru olduğunu göstermektedir.

$$Kesinlik = \frac{DP}{DP + YP} \tag{3.1}$$

- **Duyarlılık (Recall)**

Bu değer ise doğru olarak tahmin etmemiz gerekenlerden ne kadarını doğru olarak tahmin ettiğimizi gösteren bir metriktir.

$$Duyarlılık = \frac{DP}{DP + YN} \tag{3.2}$$

- **F1-Skor**

F1 Skor değeri kesinlik (Precision) ve duyarlılık (Recall) değerlerinin harmonik ortalamasını ifade eder.

$$F1\ Skor = 2 * \frac{kesinlik + duyarlılık}{kesinlik * duyarlılık} \tag{3.3}$$

- **Average Precion (AP)**

AP, kesinlik-duyarlılık eğrisinin altında kalan alan olarak tanımlanır.

$$AP = \sum_{k=0}^{k=n-1} [duyarlılık(k) - duyarlılık(k+1)] * kesinlik(k) \tag{3.4}$$

- **Mean Average Precion (mAP)**

Map, AP değerlerinin ortalamasıdır.

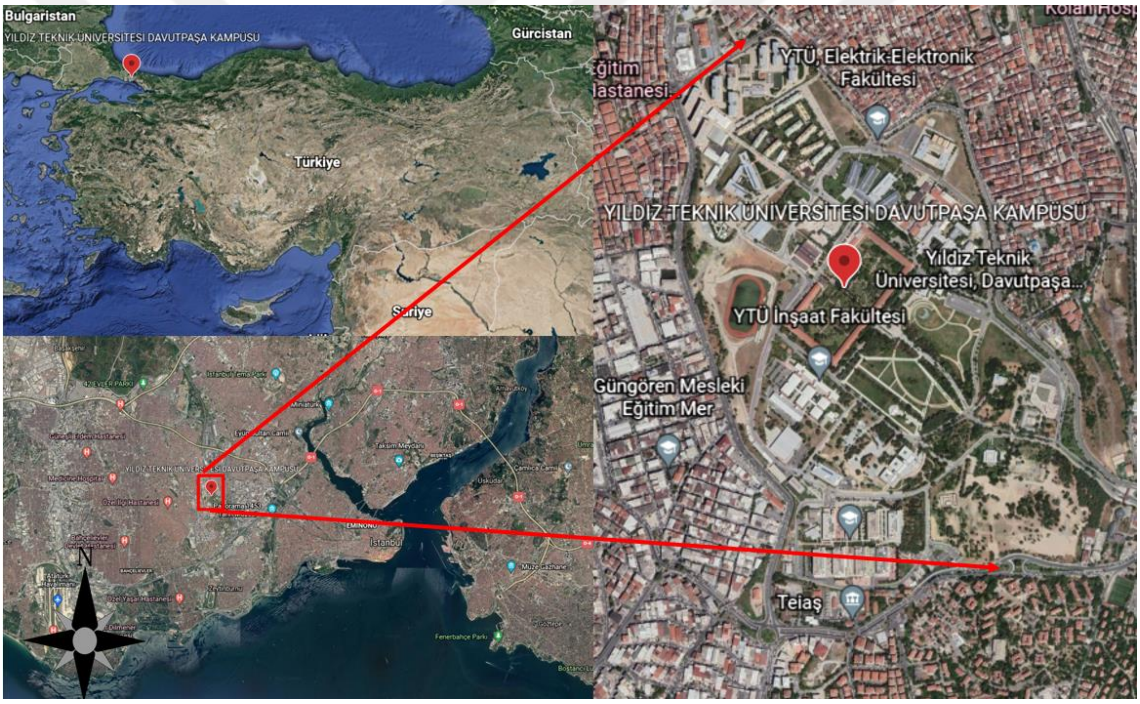
$$mAP = 1/n \sum_{k=1}^{k=n} AP_k \quad (3.5)$$

Burada AP_k, k sınıfında bulunan ortalama kesinlik değerleridir. n ise sınıf numarasıdır.

4
UYGULAMA

4.1. Çalışma Alanı

Bu çalışma, Şekil 4.1'de yaklaşık olarak gösterilen İstanbul, Esenler, Yıldız Teknik Üniversitesi (YTÜ), Davutpaşa Kampüsünde yapılmıştır. Çalışmada 2018 yılında İHA (İnsansız Hava Aracı) ile elde edilen toplam 94 adet hava fotoğrafı kullanılmıştır. Bu fotoğraflarda yer alan kampüs içi otopark alanları bu çalışmanın konusu kapsamındadır.



Şekil 4.1 YTÜ Davutpaşa Kampüsü (Google Earth)

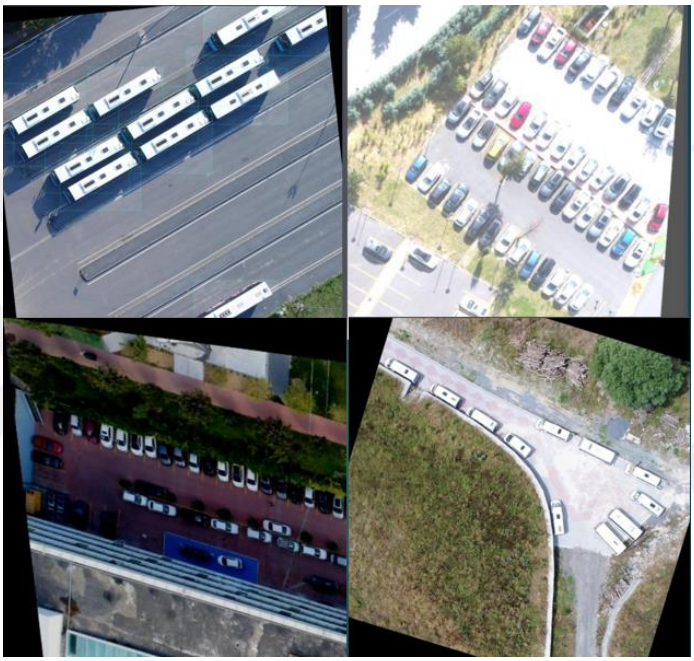
4.2. Veri Setinin Hazırlanması

YTÜ Davutpaşa Kampüsü içerisinden elde edilen hava fotoğraflarında YOLO mimarisine uygunluğunu sağlamak ve doğruluğu arttırmak için bazı ön işlemler uygulanmıştır.

İlk aşamada elde edilen fotoğraflar, modellere uygulanabilmesi için 2048 x 2048 boyutlarına kırpılmıştır. Daha sonra veriyi daha iyi temsil etmesini sağlamak için bazı veri artırma yöntemleri uygulamıştır. Uygulanan veri artırımı yöntemleri;

- -15, +15, -25, +25,+90, -90 ve 180 derece döndürme,
- 15 derece dikey, 15 derece yatay kaydırma deformasyonu,
- Parlaklık arttırmadan oluşmaktadır.

Şekil 4.2 de uygulanan veri artırımına ilişkin görseller yer almaktadır.



Şekil 4.2 Veri Arttırımı Temsili Örnekler

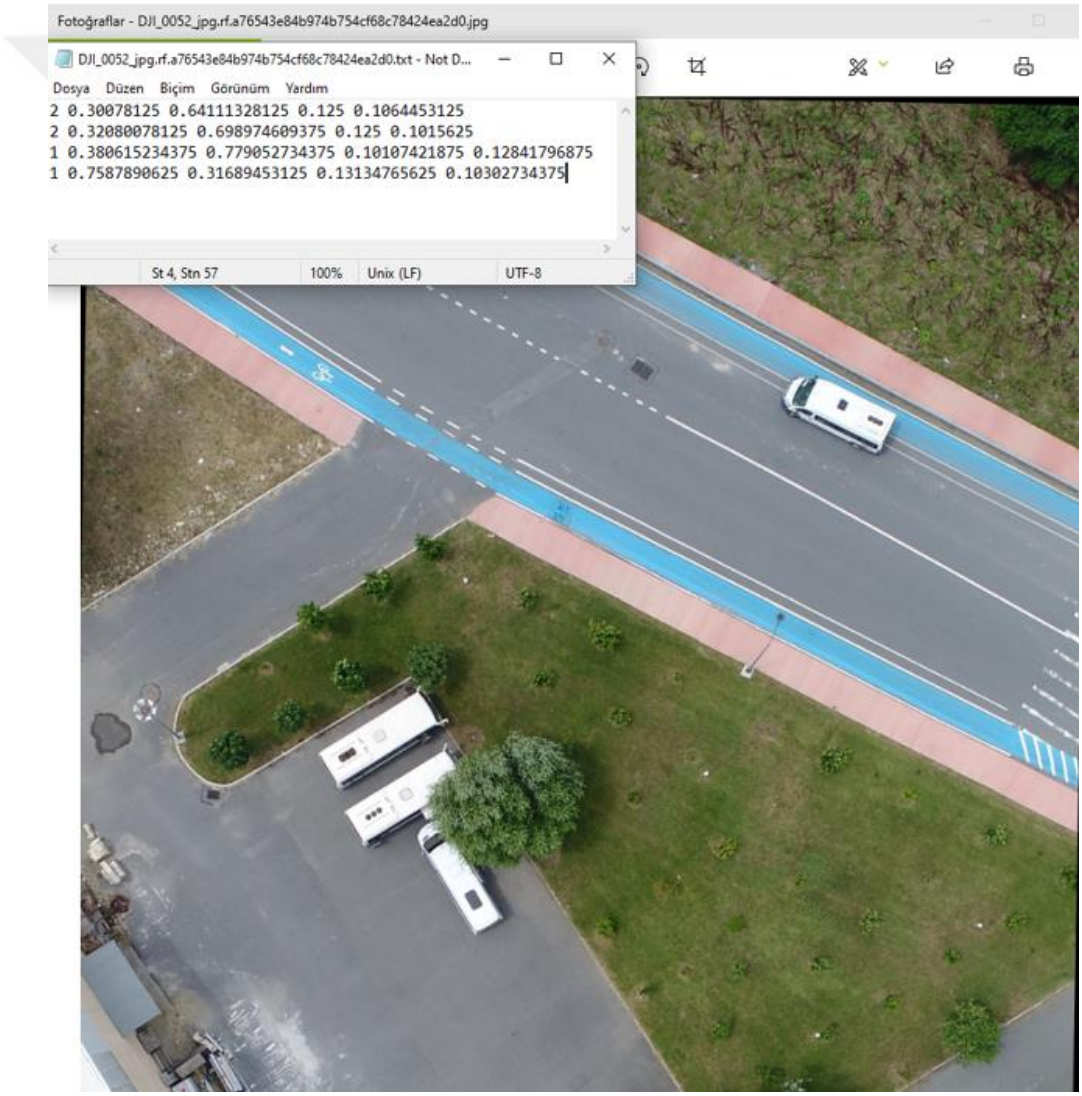
Veri artımını yapılan hava fotoğraflarından, çalışmada kullanılacak otopark alanları, test verisi olarak ayrılmıştır. Bu sayede ağın tespit performansı daha önce özelliklerini öğrenmediği veriler ile test edilebilmiştir.

Kalan fotoğraflar kampüs içerisinde yoğun olarak rastlanabilen araçlar olan araba, otobüs ve minibüs olmak üzere üç sınıf halinde etiketlenmiştir. Verilerde Tablo 4.1'de gösterildiği gibi 1223 araba etiketi, 122 minibüs etiketi, 54 otobüs etiketi bulunmaktadır.

Tablo 4. 1 Etiketli verilerin sınıf dağılımları

SINIFLAR	Araba	Minibüs	Otobüs
ETİKET SAYISI	1223	122	54

Etiketlenen veriler %80 eğitim ve %20 doğrulama olmak üzere rastgele ayrılmıştır. Daha sonra modellerin gerekliliklerine uygun olarak (.txt) formatında eğitime hazır hale getirilmiştir (Şekil 4.3).



Şekil 4.3 Etiketli veri dosyası (.txt)

4.3. YOLO Model Eğitimleri

YOLOv4-CSP, YOLOv4-tiny, YOLOv4-P5, YOLOv4-P6, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l ve YOLOv5x modellerinin her biri Tablo 4.2'de verilen hiperparametreler kullanılarak eğitilmiştir.

Tablo 4.2 Eğitimde kullanılan hiperparametre değerleri

Hiperparametreler	**Değerleri**
Küme Boyutu	16
Görüntü Giriş Boyutu	416 x 416
Momentum	Momentum: 0.937 İlk öğrenme hızı: 0.01
Eğitim Tur Sayısı	200
Aktivasyon Fonksiyonları	YOLOv4: Mish YOLOv5: Leaky ReLU
Optimizasyon Algoritması	ADAM

Eğitim aşamasında MS COCO veri seti ile eğitilen ağırlıklar başlangıç ağırlıkları kabul edilmiş ve derin öğrenme ağına aktarımlı öğrenme kullanarak dahil edilmiştir.

4.4. Kullanılan Teknolojiler

Araç tespiti çalışmamızda, bütün kodlar Google-Colabratory'nin GPU desteği sunan bulut ortamında Python programlama dilinde yazılıp çalıştırılmıştır. Etiketleme için VoTT (Visual Object Tagging Tool) programından faydalanılmıştır.

Kod içerisinde Darknet, Pytorch, Tensorflow, OpenCV ve Ultralytics kütüphaneleri kullanılmıştır. Veri artırımı, çevrimiçi Roboflow uygulamasında gerçekleştirilmiştir.

4.5. Model Eğitimlerinin Doğruluk Analizleri

Bu çalışmada 9 farklı model eğitilmiş olup her birinin doğruluk analizleri Tablo 4.3'de verilmiştir. Değerler, modelin tahmin ettiği sonuç ve elle etiketlenen veri arasındaki karşılaştırma metriklerinden meydana gelmektedir.

Tablo 4.3 Modellerin Eğitim Analizleri

MODEL	KESİNLİK	DUYARLILIK	F1-SKOR	mAP	fps
YOLOv4-CSP	0.61	0.78	0.68	0.76	11.11
YOLOv4-tiny	0.86	0.91	0.89	0.75	62.5
YOLOv4-P5	0.47	0.86	0.61	0.73	9.09
YOLOv4-P6	0.38	0.89	0.53	0.75	7.69
YOLOv5n	0.77	0.80	0.79	0.82	62.5
YOLOv5s	0.85	0.78	0.81	0.80	40,00
YOLOv5m	0.8	0.78	0.79	0.84	25,00
YOLOv5l	0.84	0.75	0.79	0.79	12.19
YOLOv5x	0.87	0.74	0.80	0.79	9.62

Tablo 4.3'de tüm YOLO modelleri için eğitim analiz sonuçları yer almaktadır. Tabloya göre F1-Skor doğruluk kriterinde en iyi F1 Skoruna ulaşan model 0.89 ile YOLO v4 tiny olmuştur. YOLO v5 modelleri yaklaşık performanslar gösterirken

YOLOv4-CSP, YOLOv4-P5 YOLOv4-P6 modellerinin F1 Skor değerlerinin diğer mimarilerden düşük olduğu görülmüştür.

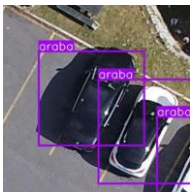
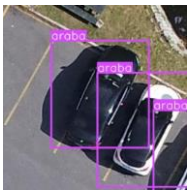
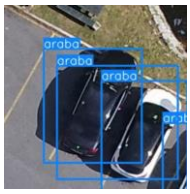
Tablo 4.3'de verilen mAP doğruluk kriterine göre YOLOv5 modellerinin tümünün YOLOv4 den daha iyi doğruluk performansı gösterdiği tablodan görülmüştür. En iyi mAP değerine ulaşan model ise %84 ile YOLOv5m olmuştur.

Araç tespitinin gerçek zamanlı yapılabilmesi için modellerin fps değerleri dikkate alınmıştır. Tablo 4.3 de 62.5 fps değeri ile en hızlı modellerin düşük performanslı sistemler için geliştirilmiş YOLOv4-tiny ve YOLOv5n olduğu görülmektedir. Bu modelleri 40 fps değeri ile ile YOLO v5s ve 25 fps değeri ile YOLO v5m takip etmiştir. En yavaş modellerin yüksek donanıma sahip sistemler için geliştirilmiş YOLOv4-P5, YOLOv4-P6 ve YOLOv5x'in olduğu görülmüştür.

YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

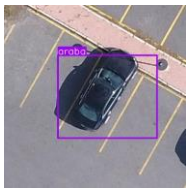
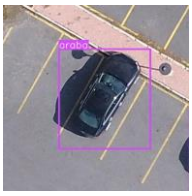
Şekil 4.4 Araç tespitinde ağaçların etkisi

Havadan elde edilen görüntülerde ağaç nesneleri araçların üzerini kapatabilmekte ve model tarafından algılanmasının önüne geçmektedir. Şekil 4.4'te üzeri kısmen kapanan bir aracın YOLOv4-CSP, YOLOv4-tiny, YOLOv4-P6, YOLOv5s, YOLOv5m modelleri tarafından tespit edildiği ancak YOLOv4-P5, YOLOv5n, YOLOv5l, YOLOv5x modelleri tarafından tespit edilemediği görüşmüştür.

YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.5 Araç tespitinde gölge etkisi

Zeminlerde oluşan siyah gölgeler, etiketlemesi yapılmış siyah araçlar ile karışmakta ve araç gölgesinin araç olarak tespit edilmesi sorununa sebep olabilmektedir. YOLOv4-P6 modelinde bu durum gözlenmektedir (Şekil 4.5).

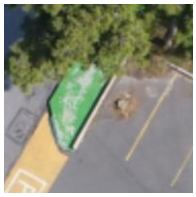
YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.6 Koyu renkli araçlarda tespit

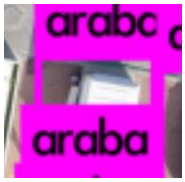
YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.7 Koyu renkli araçlarda tespit

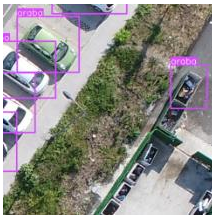
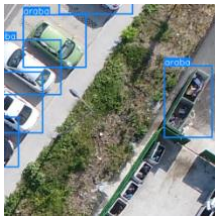
Bu çalışma kapsamında siyah, gri gibi koyu renge sahip araçların, güneş ışığının etkisiyle parlaması sebebiyle görüntüde araca ait renk dokusu yol ile aynı yansıma değerine sahip olmaktadır. Bu sebeple yol ile araç nesneleri bazı YOLO modellerinin yapılarındaki çalışma prensibi ve ağırlık parametrelerine göre tespit işlemi doğru olarak gerçekleşmemiştir. Şekil 4.6 ve Şekil 4.7'de YOLOv4-P6 ve YOLOv5x modellerinde bu problem gözlemlenmiştir.

YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.8 Araç Dışı Nesne Tespiti

YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.9 Araç dışı nesne tespiti

YOLOv4-CSP	YOLOv4-tiny	YOLOv4-P5	YOLOv4-P6	
				
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
				

Şekil 4.10 Araç dışı nesne tespiti

Şekil 4.8, Şekil 4.9 ve Şekil 4.10'da YOLOv4-CSP, YOLOv4-tiny, YOLOv4-P5, YOLOv4-P6, YOLOv5l, YOLOv5x modellerinde araç dışı nesne tespiti problemi gerçekleşmiştir. Bu problemde araç olarak etiketlenen eğitim verisine geometrik ve görüntü piksel/renk değeri yakın olan nesnelerin araç olarak tespit edildiği gözlenmiştir. Buna karşılık, YOLOv5n, YOLOv5s, YOLOv5m modellerinde ise araç olmayan ve araç boyutundaki konteyner, otobüs ile aynı gri değerlere sahip kaldırım ve yol nesnelerinde doğru araç tespiti ortaya çıkmıştır. Sonuç olarak farklı YOLO modellerinin avantaj ve dezavantajları araç tespiti kapsamında araştırılmıştır. Bu amaçla, en yaygın karşılaşılan problemlerden; koyu renkli araç tespiti, araç gölgesi tespiti, ağaçların kapattığı araçların tespiti ve araç tespit edememe durumları YOLOv4 ve YOLOv5 modelleri için Tablo 4.4'de analiz edilmiştir.

Tablo 4.4 YOLOv4-YOLOv5 karşılaştırma tablosu

Problem	YOLOv4-tiny	YOLOv5m
Koyu renkli araç tespiti	0.98	0.95
Araç gölgesi tespiti	0.89	0.99
Ağaçların kapattığı araçların tespiti	0.77	0.70
Araç tespit edememe	0.98	0.88

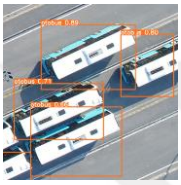
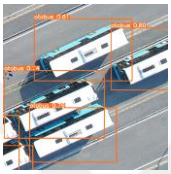
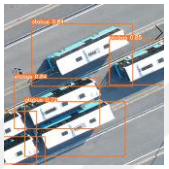
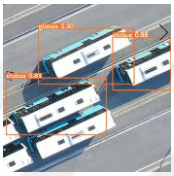
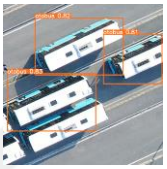
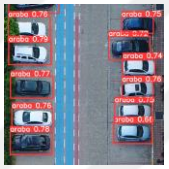
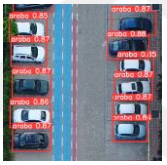
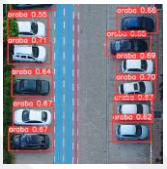
Tablo 4.4'de F1-Skorda en iyi sonucu veren YOLOv4-tiny ve mAP ölçütünde en iyi sonucu veren YOLOv5m modellerinin hata kaynaklarına ilişkin karşılaştırma sunulmaktadır. Çalışmada tespit edilen bu sorunlarda modellerin performans değerleri manuel sayım sonucunun yüzdelik değerleri olarak düzenlenmiştir. Tabloya göre siyah ve gri gibi koyu renkli araçların karşılaşılan güneş ışığının etkisi ile parladığında, yol ile aynı piksel gri değerine sahip olmasının sebep olduğu sorunda, bu iki model 0.95 in üzerinde performans göstermiştir. Araçların sahip olduğu gölgeleri araç olarak tespit etmede ise YOLOv5m 0.99 performansla YOLOv4-tiny modelinden daha iyidir. Ağaçların sebep olduğu havadan görüş kısıtlaması sorununda iki modelde kısmen tespit sorunu yaşamış YOLOv4-tiny %77, YOLOv5m %70 performans değerlerine ulaşabilmişlerdir. Araçları tespit etmede ise YOLOv4-tiny modelinde %98 ile daha iyi sonuç alınmıştır.

Tablo 4.5 Sınıflara ait mAP analizleri

Model	Araba	Minibüs	Otobüs
YOLOv5n	0,94	0,80	0,60
YOLOv5s	0,82	0,73	0,77
YOLOv5m	0,90	0,73	0,74
YOLOv5l	0,94	0,82	0,66
YOLOv5x	0,93	0,82	0,61

Tablo 4.5'de YOLOv5 de araba, otobüs ve minibüs sınıflarının mAP sonuçları yer almaktadır. Sonuçlar arabaların bütün modellerde en iyi tespit edilen sınıf olduğunu göstermiştir. Genel doğruluğu düşüren sınıfların ise otobüs ve minibüs sınıfları olduğu tablodan anlaşılmaktadır.

Tablo 4.6 Sınıfların tespit performansı incelemesi

YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	Şekil No/ Açıklama
					**Şekil 4.13** Minibüs tespiti
					**Şekil 4.14** Otobüs tespiti
					**Şekil 4.15** Araba tespiti

Modellerin sınıflara ilişkin analiz görüntüleri tabloda verilmiştir. Araba sınıfı tüm YOLOv5 modellerinde %80 üzerinde başarı oranı ile tespit edilmiştir. (Tablo 4.5 ve Tablo 4.6). Otobüs ve minibüs sınıflarında ise YOLOv5n, YOLOv5l, YOLOv5x modellerinin aracı tespit edememe ve sınıf karışıklığı problemleriyle çok fazla karşılaştığı görülmüştür. YOLOv5s ve YOLOv5m modellerinin eğitim örnek sayısı az olan otobüs ve minibüs sınıflarında referans değerlerle daha tutarlı tespit sonuçları verdiği gözlemlenmiştir.

5
SONUÇ VE ÖNERİLER

YOLO modelleri kullanarak YTÜ Davutpaşa Kampüs içerisinden elde edilen hava fotoğrafları ile araç tespitinin yapıldığı bu çalışmada, öncelikli olarak fotoğraflar üzerinde model gerekliliğinin yerine getirilmesi amacıyla kırpma işlemi yapılmıştır. Kırpılan fotoğraflar üzerinde modelin doğruluk performansını arttırmak için veri artırımı adımları uygulanmış veri sayıca üç katına büyütülmüştür. Fotoğraflar etiketlenerek eğitime hazır hale getirilmiştir. Elde edilen veri seti YOLO modellerinde aktarımlı öğrenme kullanarak eğitilmiştir. Eğitim sonuçlarında modellerin doğruluk analizi ve hız karşılaştırmaları yapılmıştır. Son aşamada eğitilen ağırlıklar otopark alanlarını içeren test verilerinde uygulanmıştır.

Uygulama aşamasında 2048 x 2048 boyutlarındaki 94 fotoğrafın etiketlenmesi 5 saat sürmüştür. Modellerin eğitimi Google-Colabratory GPU desteği ile bulut ortamında, derinliği az olan modeller için yaklaşık 2 saat sürerken model derinliği ve karmaşıklığı arttıkça süreler 7 saati bulmuştur. Toplam eğitim süresi 9 model için yaklaşık 40 saat sürmüştür.

Kullanılan YOLO mimarisi modellerinin doğruluk analizinde, F1 skor ve mAP doğruluk kriterleri dikkate alınmıştır. F1 skora göre en iyi doğruluk değerine %89 ile YOLOv4-tiny ulaşmıştır. YOLOv5 modellerinin tümü %79-%81 aralığında yakın performanslar gösterirken kullanılan diğer YOLOv4-CSP, YOLOv4-P5 ve YOLOv4-P6 modellerinin performansları %70'in altında kalmıştır. mAP kriterinde en iyi sonucu %84 ile YOLO v5m modeli elde etmiştir.

Modellerin karşılaştığı sorunlar test verilerinden analiz edilmiştir. Modellerin sıklıkla karşılaştığı problemler; elde edilen görüntülerde ağaçların, araçların üzerini kapladığı durumlarda, koyu renkli araçların güneş etkisiyle parlaması ve yol nesnesi ile aynı piksel gri değere sahip olması sebebiyle bazı YOLO modellerinde araçların doğru tespit edilmediği gözlemlenmiştir. Ayrıca araç gölgelerinden ve araca ile aynı geometrik ve renk niteliklerine sahip araç dışı

yabancı nesnelerden kaynaklanan eksik ve doğru olmayan tespitler oluşmuştur. bu durumun önemli sebeplerinden biri de modelin yeterince iyi eğitilememesi diğer bir ifade ile farklı nesnelerin yetersiz etiketlenmesinden dolayı özellik öğrenememesinin sonucu olarak hedef sınıflarının doğru tespit edilememesinden kaynaklanmaktadır.

F1-Skor doğruluk kriterinde en iyi sonuca ulaşan YOLOv4-tiny ve mAP doğruluk kriterinde en iyi sonuca ulaşan YOLOv5m modelleri arasında YOLOv4-tiny modelinde araç dışı tespitlerin daha fazla olduğu ve sınıf karışıklıklarının daha fazla yaşandığı görülmüştür. YOLOv5m modelinde sonuçların referans değerler ile daha tutarlı olduğu gözlenmiştir. YOLOv5m modelinde sınıf karışıklığı ve araç dışı algılama sorunu daha az yaşanırken bu modelin daha çok hata kaynağının, incelenen test görüntülerinde tespit edemediği araçlar olduğu görülmüştür.

Araç tespitinin gerçek zamanlı yapılabilmesi için modellerin fps değerleri dikkate alınmıştır. Tablo 4.3 de 62.5 fps değeri ile en hızlı modellerin düşük performanslı sistemler için geliştirilmiş YOLOv4-tiny ve YOLOv5n olduğu görülmektedir. Bu modelleri 40 fps değeri ile ile YOLO v5s ve 25 fps değeri ile YOLO v5m takip etmiştir. En yavaş modellerin yüksek donanıma sahip sistemler için geliştirilmiş YOLOv4-P5, YOLOv4-P6 ve YOLOv5x'in olduğu görülmüştür.

YOLOv5 modellerinde sınıfların eğitim performansı mAP kriterine göre değerlendirilmiştir. Araba sınıfı, modellerin tümünde %82'nin üzerinde performansa ulaşmıştır. Genel doğruluğu düşüren sınıfların ise otobüs ve minibüs sınıfları olduğu görülmüştür. Eğitim setinin, kampüs sınırlarından meydana geldiği bu çalışmada otobüs ve minibüs sınıflarının sayıca az olması bu sınıflarda yeterince veri sayısına ve çeşitliliğine ulaşılamadığını göstermiştir.

Bu çalışmada sınırlı GPU desteği ile analizler yapıldığından büyük ölçekli modellerin yeterince eğitilemediği düşünülmektedir. Bu sebeple YOLOv4-P5, YOLOv4-P6, YOLOv5l, YOLOv5x modellerinin gerçek performansını görmek için sınırsız GPU desteği ile daha yüksek eğitim tur sayılarında eğitilmesi önerilmektedir.

Bu tez, YOLO modellerinin araçları tespit etmedeki başarılarını analiz etmiş ve karşılaştırmalar sunmuştur. Ancak kapsamlı uygulamalar yapmak için veri setinin

sayıca çoğaltılması ve çeşitlendirilmesi gerektiği düşünülmektedir. Ancak veri setinin artışı büyük veri işleme sonunu da beraberinde getireceğinden yüksek donanımlara ihtiyaç duyulacağı öngörülmektedir.

Son yıllardaki teknolojik ilerleme derin ağların eğitimi ve büyük veri işleme konusunda çok hızlı gelişme göstermektedir. Bu gelişmeler ışığında doğru ve gerçek zamanlı araç tespiti coğrafi bilgi sistemleri ile entegre çalıştığında otoyolların ve otoparkların otomatik olarak yönetilmesine ve planlanmasına zemin hazırlayacağı öngörülmektedir.

KAYNAKÇA

[1] Aghdam, H. H. ve Heravi, E. J., Guide to convolutional neural networks. New York, NY: Springer, 10(978-973), 51, 2017.

[2] Gümüşçü, A., Taşaltın, R., & Aydilek, İ. B. C4. 5 Karar ağaçlarında genetik algoritma ile budama. Dicle University, DUFED, 5(2), 77-80, 2016.

[3] Joshi, A. J. ve Porikli, F., Scene-adaptive human detection with incremental active learning. In 2010 20th International Conference on Pattern Recognition, pp. 2760-2763, IEEE, 2010.

[4] Chollet, F., Python ile Derin Öğrenme. Buzdağı Yayınevi, 71, 2019.

[5] Ren, S., He, K., Girshick, R. ve Sun, J., Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 91-99, 2015.

[6] Özkan, İ. N. İ. K. ve Ülker, E., Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. Gaziosmanpaşa Bilimsel Araştırma Dergisi, 6(3), 85-104, 2017.

[7] Sun X., Wua P., Hoi S C.H., Face detection using deep learning: An improved faster RCNN approach, Neurocomputing, 2018.

[8] Sharma, V., ve Mir, R. N.,. A comprehensive and systematic look up into deep learning based object detection techniques: A review. Computer Science Review, 38, 100301, 2020.

[9] Nhu, V. H., Hoang, N. D., Nguyen, H., Ngo, P. T. T., Bui, T. T., Hoa, P. V., ... ve Bui, D. T., Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area. Catena, 188, 104458, 2020.

[10] Bui, D. T., Hoang, N. D., Martínez-Álvarez, F., Ngo, P. T. T., Hoa, P. V., Pham, T. D., ... ve Costache, R., A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. Science of The Total Environment, 701, 134413, 2020.

[11] Wang, Q., Cheng, M., Xiao, X., Yuan, H., Zhu, J., Fan, C., ve Zhang, J., An image segmentation method based on deep learning for damage assessment of the invasive weed Solanum rostratum Dunal. Computers and Electronics in Agriculture, 188, 106320, 2021.

[12] Rawat, A., Kumar, A., Upadhyay, P., ve Kumar, S., Deep learning-based models for temporal satellite data processing: Classification of paddy transplanted fields. Ecological Informatics, 61, 101214,2021.

[13] Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T., ve Yang, X., A review of deep learning based methods for medical image multi-organ segmentation. Physica Medica, 85, 107-122, 2021.

[14] Schneider, L., Laiouar-Pedari, S., Kuntz, S., Krieghoff-Henning, E., Hekler, A., Kather, J. N., ve Brinker, T. J., Integration of deep learning-based image analysis and genomic data in cancer pathology: A systematic review. European Journal of Cancer, 160, 80-91, 2022.

[15] Safonova, A., Tabik, S., Alcaraz-Segura, D., Rubtsov, A., Maglinets, Y., ve Herrera, F., Detection of fir trees (Abies sibirica) damaged by the bark beetle in unmanned aerial vehicle images with deep learning. Remote sensing, 11(6), 643, 2019.

[16] Liu, Y., Zhang, Z., Liu, X., Wang, L., ve Xia, X., Efficient image segmentation based on deep learning for mineral image classification. Advanced Powder Technology, 32 (10), 3885-3903, 2021.

[17] Aktaş, A., Doğan, B., ve Demir, Ö., Derin öğrenme yöntemleri ile dokunsal parke yüzeyi tespiti. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 35(3), 1685-1700, 2020.

[18] Peng, B., Keskin, M. F., Kulcsár, B., ve Wymeersch, H., Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning. Communications in Transportation Research, 1, 100017, 2021.

[19] Ataei, B., Development of video processing algorithm (YOLO) in autonomous vessels operations, Yüksek Lisans Tezi, University of South-Eastern Norveç, 2019.

[20] Han, X., Chang, J., ve Wang, K., Real-time object detection based on YOLO-v2 for tiny vehicle object. Procedia Computer Science, 183, 61-72, 2021.

[21] Wu, D., Lv, S., Jiang, M., ve Song, H., Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 178, 105742, 2020.

[22] Tan, L., Lv, X., Lian, X., ve Wang, G., YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm. Computers & Electrical Engineering, 93, 107261, 2021.

[23] Li, H., Li, C., Li, G., ve Chen, L., A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. Biosystems Engineering, 212, 347-359, 2021.

[24] Cheng, H. Y., Weng, C. C., ve Chen, Y. Y. Vehicle detection in aerial surveillance using dynamic Bayesian networks. IEEE transactions on image processing, 21(4), 2152-2159, 2011.

[25] Chen, X., Xiang, S., Liu, C. L., ve Pan, C. H., Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and remote sensing letters, 11(10), 1797-1801, 2014.

[26] De Almeida, P. R., Oliveira, L. S., Britto Jr, A. S., Silva Jr, E. J., ve Koerich, A. L., PKLot–A robust dataset for parking lot classification. Expert Systems with Applications, 42(11), 4937-4949, 2015.

[27] Cai, Y., Du, D., Zhang, L., Wen, L., Wang, W., Wu, Y., ve Lyu, S., Guided attention network for object detection and counting on drones. arXiv preprint arXiv:1909.11307, 2019.

[28] Dimililer, K., Ever, Y. K. ve Mustafa, S. M., Vehicle detection and tracking using machine learning techniques. In International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions, pp. 373-381. Springer, Cham, 2019.

[29] Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., ve Liu, Q., A comparative study of state-of-the-art deep learning algorithms for vehicle detection. IEEE Intelligent Transportation Systems Magazine, 11(2), 82-95, 2019.

[30] Tan, Z., Derin öğrenme yardımıyla araç sınıflandırma, Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2019.

[31] Bui, N., Yi, H., ve Cho, J., A vehicle counts by class framework using distinguished regions tracking at multiple intersections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 578-579), 2020.

[32] Miao, Y., Liu, F., Hou, T., Liu, L., ve Liu, Y., A Nighttime Vehicle Detection Method Based on YOLO v3. In 2020 Chinese Automation Congress (CAC) (pp. 6617-6621). IEEE, 2020.

[33] Mandal V., ve Adu-Gyamfi, Y., Object detection and tracking algorithms for vehicle counting: a comparative analysis. Journal of Big Data Analytics in Transportation, 2(3), 251-261, 2020.

[34] Kilic, E., ve Ozturk, S., An accurate car counting in aerial images based on convolutional neural networks. Journal of Ambient Intelligence and Humanized Computing, 1-10, 2021.

[35] Ertel, W., Machine learning and data mining. In Introduction to Artificial Intelligence (pp. 175-243). Springer, Cham, 2017.

[36] Internet: Sharma S., Activation Functions in Neural Networks, https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, 21.01.2022.

[37] Internet: Kızrak A., Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması, https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd, 21.01.2022.

[38] Internet: Ayten B., Yapay Sinir Ağlarında Aktivasyon Fonksiyonları, https://medium.com/databulls/yapay-sinir-a%C4%9Flar%C4%B1nda-aktivasyon-fonksiyonlar%C4%B1-11002b8ac522, 21.01.2022.

[39] Misra, D., Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681, 2019.

[40] Internet: Saurabh, Backpropagation – Algorithm For Training A Neural Network, https://www.edureka.co/blog/backpropagation/, 21.01.2022.

[41] Internet: Milman O., kaybolan gradyan nedir?, https://istatistikler.narkive.info.tr/z99r2ZfX/kaybolan-gradyan-nedir, 21.02.2022.

[42] Internet: Log Kaybı ve Çapraz Entropi (Log Loss and Cross-Entropy) https://veribilimcisi.com/2018/05/04/log-kaybi-ve-capraz-entropi-log-loss-and-cross-entropy/, 21.01.2022.

[43] Internet: Brownlee J., Loss and Loss Functions for Training Deep Learning Neural Networks, https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/, 21.01.2022.

[44] Seyyayer, E., Ayata, F., Uçkan, T., ve Karcı, A., Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarinin Uygulanmasi Ve Kiyaslanmasi. Computer Science, 5(2), 90-98, 2020.

[45] Doğan, F., ve Türkoğlu, İ., Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 409-445, 2019.

[46] Internet: Tuncer E., Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?, https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad, 22.01.2022.

[47] Kızrak, M. A., ve Boalt, B., Derin öğrenme ile kalabalık analizi üzerine detaylı bir araştırma. Bilişim Teknolojileri Dergisi, 11(3), 263-286, 2018.

[48] Internet: Çarkacı N., Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler, https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4.

[49] Redmon, J., Divvala, S., Girshick, R., ve Farhadi, A., You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[50] Redmon, J., ve Farhadi, A., YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 7263-7271, 2017

[51] Redmon, J., & Farhadi, A., Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018

[52] Bochkovskiy, A., Wang, C. Y., ve Liao, H. Y. M., Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020

[53] Internet: Çakmak D. YOLOv4 ve YOLOv5 https://deryacakmak.medium.com/yolov4-ve-yolov5-9c4ad208579e, 21.01.2022

[54] Internet: Overview of model structure about YOLOv5 #280 https://github.com/ultralytics/yolov5/issues/280, 21.01.2022

[55] Shorten, C., & Khoshgoftaar, T. M., A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48, 2019

[56] Bazi, Y., Bashmal, L., Rahhal, M. M. A., Dayil, R. A., & Ajlan, N. A., Vision transformers for remote sensing image classification. Remote Sensing, 13(3), 516, 2021

[57] Internet: Ravalı S., Veri Etiketleme (Data Annotation), https://www.artiwise.com/2021/08/13/veri-etiketleme, 21.01.2022

[58] Internet: Sharma P., Understanding Transfer Learning for Deep Learning, https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/#h2_5, 21.01.2022

[59] Internet: Nova Research Lab., Öğrenme Aktarımı/Transfer Learning, https://medium.com/novaresearchlab/%C3%B6%C4%9Frenme-aktar%C4%B1m%C4%B1-transfer-learning-c0b8126965c4, 21.01.2022

[60] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M., Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13029-13038, 2021

[61] Internet: Rajput M., https://towardsdatascience.com/yolo-v5-is-here-b668ce2a4908, 21.01.2022

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Öztürk Ş. ve Uzar M., Derin Öğrenme İle Araç Tespiti: Ytü Kampüs Otopark Alanları Örneği, ISPEC 2. Ulusal Bilimsel Araştırmalar Kongresi, s. 74, 2022