T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE GÜVENLİK KAMERASI GÖRÜNTÜLERİNDEN APRONLARDA YER OPERASYONLARININ GERÇEK ZAMANLI ANALİZİ

Berna YILMAZ

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Doç. Dr. M. Elif KARSLIGİL

Temmuz, 2021

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE GÜVENLİK KAMERASI GÖRÜNTÜLERİNDEN APRONLARDA YER OPERASYONLARININ GERÇEK ZAMANLI ANALİZİ

Berna YILMAZ tarafından hazırlanan tez çalışması 08.07.2021 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. M. Elif KARSLIGİL
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. M. Elif KARSLIGİL, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Mustafa KAMAŞAK, Üye
İstabul Teknik Üniversitesi _____

Doç. Dr. Gökhan BİLGİN, Üye
Yıldız Teknik Üniversitesi _____

Danışmanım Doç. Dr. M. Elif KARSLIGİL sorumluluğunda tarafımca hazırlanan Derin Öğrenme Yöntemleri ile Güvenlik Kamerası Görüntülerinden Apronlarda Yer Operasyonlarının Gerçek Zamanlı Analizi 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.

Berna YILMAZ

İmza

Aileme ithaf edilmiştir

TEŞEKKÜR

Bu tez projesinin planlanması, yürütülmesi ve yazılması hususlarında göstermiş olduğu destek ve yardımlarından ötürü tez danışmanım Yıldız Teknik Üniversitesi Öğretim Üyesi Doç. Dr. M. Elif KARSLIGİL'e teşekkürlerimi sunuyorum.

Ayrıca çalışmam için gerekli veri kümesinin temini için Türk Hava Yolları Teknoloji AR-GE Merkezi'ne, çalışmam konusunda desteğini esirgemeyen Türk Hava Yolları yöneticilerinden Sayın Serkan Aydın'a teşekkürlerimi sunuyorum.

Yaşamımın her döneminde olduğu gibi çalışmalarım esnasında da fedekarlık gösterip her anlamda destek oldukları için ve bana duydukları güven için anneme, babama ve kardeşime en derin teşekkürlerimi sunuyorum.

Berna YILMAZ

İÇİNDEKİLER

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

TABLO LİSTESİ x

ÖZET xi

ABSTRACT xiii

1 GİRİŞ 1

1.1 Literatür Özeti . . . 2

1.1.1 Nesne Tespiti . . . 2

1.1.2 Nesne Takibi . . . 3

1.1.3 Apron Görüntülerini Kullanan Çalışmalar . . . 4

1.2 Tezin Amacı . . . 5

1.3 Hipotez . . . 6

2 NESNE TESPİTİ VE NESNE TAKİBİ 9

2.1 Derin öğrenme . . . 9

2.1.1 Evrişimsel Sinir Ağı . . . 11

2.1.2 Hiper parametreler . . . 12

2.2 Nesne Tespiti . . . 17

2.2.1 YOLO ile Nesne Tespiti . . . 18

2.2.2 Bölge Tabanlı Evrişimsel Sinir Ağı . . . 19

2.3 Nesne Takibi . . . 24

2.3.1 SiamMask Mimarisi . . . 25

3 SİSTEM TASARIMI 27

3.1 Veri Kümesi . . . 27

3.2 Sistem Mimarisi . . . 30

3.2.1 Eğitim Aşaması . . . 33

3.2.2 Omurga Ağı . . . 33

3.2.3 Nesnerin İlişkilendirilmesi . . . 34

3.2.4 Operasyonların Tespiti 34

4 DENEYSEL SONUÇLAR **41**

4.1 Performans Ölçümü 41

4.1.1 IoU 41

4.1.2 Kesinlik (Precision) ve Duyarlılık (Recall) 41

4.1.3 Ortalama Kesinlik (Average Precision, AP) Hesaplaması 43

4.2 Nesne Tespit Başarısı 43

4.3 Nesne Takip Başarısı 44

4.4 Operasyon Tespit Başarısı 45

4.5 Kullanılan Kütüphaneler 46

5 SONUÇ VE ÖNERİLER **50**

KAYNAKÇA **51**

A DERİN ÖĞRENME KÜTÜPHANELERİ **56**

B HAVACILIK TERİMLERİ **57**

TEZDEN ÜRETİLMİŞ YAYINLAR **58**

KISALTMA LİSTESİ

CCTV	Kapalı Devre Televizyon(Closed-Circuit Television)
CPU	Merkezi İşlem Birimi(Central Process Unit)
ESA	Evrişimsel Sinir Ağları
Faster R-CNN	Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı(Faster Region Based Convolutional Neural Network)
GPU	Grafik İşlem Ünitesi(Graphics Processing Unit)
IoU	Kesişimin Birleşme Oranı(Intersection Over Union)
LSTM	Uzun Kısa Vadeli Hafıza Ağları(Long Short-Term Memory)
R-CNN	Bölge Tabanlı Evrişimsel Sinir Ağı(Region Based Convolutional Neural Networks)
ReLu	Doğrultulmuş Lineer Birim(Rectified Linear Unit)
ROI	İlgi Bölgesi(Region of Interest)
RPN	Bölge Öneri Ağları(Region Proposal Networks)
RNN	Özyinemeli Sinir Ağları(Reccurrent Neural Network)
SORT	Basit Çevrimiçi ve Gerçek Zamanlı Takip(Simple Online and Realtime Tracking)
SVM	Destek Vektör Makinesi(Support Vector Machine)
YOLO	You Only Look Once
YSA	Yapay Sinir Ağı

ŞEKİL LİSTESİ

Şekil 1.1 (sol) Anlamsal Bölütleme, (sağ) Örnek Bölütleme 3
Şekil 1.2 Operasyonel Süreçlerden Bazıları 7
Şekil 2.1 Tek katmanlı yapay sinir ağı 10
Şekil 2.2 Çok katmanlı temel yapay sinir ağı mimarisi 10
Şekil 2.3 Evrişimsel sinir ağının genel mimarisi [24] 11
Şekil 2.4 Bazı optimizasyon algoritmalarının MNIST üzerinde çalışma zamanı grafiği [28] 13
Şekil 2.5 İkili basamak fonkiyonu 14
Şekil 2.6 Doğrusal aktivasyon fonksiyonu 15
Şekil 2.7 Sigmoid aktivasyon fonksiyonu 15
Şekil 2.8 Tanh aktivasyon fonksiyonu 16
Şekil 2.9 ReLu aktivasyon fonksiyonu 16
Şekil 2.10 Sızıntı aktivasyon fonksiyonu 17
Şekil 2.11 YoloV3 ağının çıkışındaki hücre bilgisi [40] 19
Şekil 2.12 NMS işlemi [41] 20
Şekil 2.13 R-CNN mimarisi [42] 20
Şekil 2.14 Seçici arama yöntemi [43] 21
Şekil 2.15 Fast R-CNN mimarisi [44] 21
Şekil 2.16 Faster R-CNN mimarisi [9] 22
Şekil 2.17 RPN mimarisi [9] 23
Şekil 2.18 Mask R-CNN genel mimarisi [45] 23
Şekil 2.19 SiamMask Mimarisi [52] 26
Şekil 3.1 Sistem Akış Diyagramı 28
Şekil 3.2 Gündüz ortamda alınan örnek görüntü 29
Şekil 3.3 Karanlık ortamda alınan örnek görüntü 30
Şekil 3.4 Oluşturulan sınıf örneği 31
Şekil 3.5 Sistem Genel Mimarisi 32
Şekil 3.6 Nesne tespiti kayıp değerleri 33
Şekil 3.7 ResNet mimarileri [36] 34
Şekil 3.8 Başlangıç aşaması 35
Şekil 3.9 Nesnelerin merkez noktalarıyla öklid mesafesi hesaplaması 36

Şekil 3.10 Nesnelerin merkez noktalarının güncellenmesi 37
Şekil 3.11 İkram Yüklemesi Olayı 38
Şekil 3.12 Yakıt aracının gelmesi ve yakıt yükleme işlemi 39
Şekil 3.13 Pushback işlemi 40
Şekil 4.1 IoU 42
Şekil 4.2 Nesne Tespit Örnekleri 44
Şekil 4.3 Farklı sınırlayıcı kutu gösterimleri[52] 45
Şekil 4.4 Nesne Takip örnek IoU değerleri 47
Şekil 4.5 Nesnelerin birbirinin önüne geçerek takibin başarısız olduğu örnek - Takip etme algoritmasından elde edilen sınırlayıcı kutu mavi renk, gerçek referans(ground-truth) kırmızı ile gösterilmiştir. 48
Şekil 4.6 Operasyon Tespiti 49

TABLO LİSTESİ

Tablo 3.1 Sınıflar ve örnek sayıları 31
Tablo 4.1 Duyarlılık değerleri 43
Tablo 4.2 VOT-2016 Veri kümesi üzerinde farklı sınırlayıcı kutu gösterim performansları[52] 44
Tablo 4.3 Nesnelerin takip başarısı 45
Tablo 4.4 Olayların tespitindeki gecikme süresi 46
Tablo A.1 Bazı derin öğrenme kütüphaneleri ve ek bilgileri 56
Tablo B.1 Bazı havacılık terimleri 57

ÖZET

Derin Öğrenme Yöntemleri ile Güvenlik Kamerası Görüntülerinden Apronlarda Yer Operasyonlarının Gerçek Zamanlı Analizi

Berna YILMAZ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. M. Elif KARSLIGİL

İstanbul Havalimanı'ndan günlük olarak yaklaşık 1000 uçuş gerçekleşmektedir. Her uçuş için uçakların havalimanı apron alanında kaldıkları süre boyunca sürdürülen operasyonel faaliyetleri mevcuttur. İnişten sonra veya kalkıştan önce yapılması gerekli işlemler bulunmaktadır. İşlemler uçağın apron park alanına gelmesi ile başlayıp, geri itme işleminin bitmesiyle sona erer. Mevcut durumda, bu süreçler farklı sistemlerden gelen bilgiler ve gözle yapılan değerlendirmeler sonucu takip edilmekte ve gerek güvenlik, gerekse operasyon süreçlerinin optimizasyonu için değerlendirilmektedir. Yapılan operasyonların kontrolünün, takibinin ve standardizasyonunun sağlanabilmesi için operasyon işlemlerinin otomatize edilmesi önemli bir fayda sağlayacaktır. Bu tez çalışmasında, video görüntülerini değerlendirerek, uçağın apronda bulunduğu süre boyunca yapılan operasyonel faaliyetlerin tespit, takip ve raporlamasını yapan derin öğrenme tabanlı bir sistem tasarlanmış ve gerçeklenmiştir.

Konu ile ilgili genel kullanıma açık bir veri kümesi bulunmadığı için, önce güvenlik kamerasının farklı zamanlara ait görüntüleri kullanılarak veri kümesi hazırlanmıştır. Operasyonel işlemlerin değerlendirilmesi görüntüde uçağın tespiti ile başlamaktadır. Uçak iniş yaptıktan sonra körük yanaşması, ikram yükleme, yakıt yükleme gibi farklı işlemlerin tespiti yapılmaktadır. Uçak ve operasyonlarda yer alan araçların tespiti için bölge tabanlı evrişimsel sinir ağı modellerinden biri olan Mask R-CNN yöntemi kullanılmıştır. Yapılan işlemlerin oryantasyonunun uygunluğunun anlaşılabilmesi

için uçağın kapıları ve kuyruk bölümü de yine Mask R-CNN kullanılarak tespit edilmiştir. Uçak ve diğer araçların takibi için farklı nesne takip yöntemleri incelenmiş ve derin öğrenme tabanlı SiamMask algoritması kullanılmıştır. Operasyonda yer alan araçların uçak ile bağlantısı başladıktan sonra aracın konumu ve hareket yönü bilgileri kullanılarak yapılan işlemler hakkında çıkarımlarda bulunularak operasyonun türü, başlangıç ve bitiş zamanı bilgileri çıkartılmıştır.

Olayların başarısını nesne tespiti ve takibi doğrudan etkilemektedir. Bu sebeple her aşamada performans değerlendirmesi yapılmıştır. Sistemin genel başarısının gerçek zamanlı kullanım için yeterli olduğu görülmüştür.

Anahtar Kelimeler: Nesne tespiti, Çoklu nesne takibi, Video analizi, Derin öğrenme, Apronda yer operasyon tespiti

ABSTRACT

Real-Time Analysis of Ground Operations on Aprons from Security Camera Images with Deep Learning Methods

Berna YILMAZ

Department of Computer Engineering
Master of Science Thesis

Advisor: Assoc. Prof. Dr. M. Elif KARSLIGİL

There are approximately 1000 flights daily from Istanbul Airport. For each flight, there are operational activities that continue during the aircraft's stay in the airport apron area. There are procedures to be done after landing or before take-off. Operations begin with the aircraft's arrival at the apron parking area and end with the end of the push-back operation. Currently, these processes are monitored as a result of the informations coming from different systems and visual evaluations, and evaluated for both security and optimization of operational processes. Automating processes of the operation will provide a significant benefit in order to ensure the control, follow-up and standardization of the operations. In this thesis, a deep learning-based system that evaluates videos and detects, monitors and reports operational activities while the aircraft is on the apron has been designed and implemented.

Since there is no publicly available dataset on the subject, a dataset was prepared with security camera footages taken from various times. Evaluation of operational processes begins with the detection of the aircraft. After the aircraft has landed, different operations such as fuel, catering loading are detected. Mask R-CNN method, which is one of the region-based convolutional neural network models, was used to detect aircraft and vehicles in operations. In order to understand the operations performed, the doors, front wheel and tail part of the aircraft were determined using Mask R-CNN. Different object tracking methods have been examined for tracking aircraft and other vehicles and the deep learning based SiamMask algorithm has been used. After the connection of the vehicles involved in the operation with the aircraft

began, inferences were made about the operations performed using the vehicle's position and direction of movement information, and the type of operation, start and end time information were obtained.

Object detection and tracking directly affect the success of these events. For this reason, performance evaluation was made at each stage. It has been seen that the overall success of the system is sufficient for real-time use.

Keywords: Object detection, Multi object tracking, Video analysis, Deep Learning, Ground operation detection on apron

1
GİRİŞ

Apron alanları güvenlik sebebiyle 7/24 gözetim kameraları ile izlenmektedir. Uçağın inişi ve kalkışı arasında geçen süre içerisinde uçağa verilen hizmetler olarak tanımlanan yer hizmetleri belli bir sistem içerisinde yürütülmektedir. Bu işlemler esnasında apron alanları güvenlik kameraları ile gözlenmektedir. İşlemler, uçağın apron alanına gelmesi ile başlamakta, kalkış hazırlığı için uçağın geri itme (pushback) hizmeti ile sona ermektedir. Yapılan işlemler şu şekilde sıralanabilir:

- Uçağı karşılama ve park ettirme
- Takozların yerleştirilmesi
- Uçak etrafına konilerin yerleştirilmesi
- Enerjilendirme
- Gövde hasar kontrolü
- Körük ve merdiven hizmetinin sağlanması
- Uçak kapılarının açılması
- Yolcuların güvenli bir şekilde tahliyesi
- Kargo ve bagajların emniyetli bir şekilde boşaltılması
- Yakıt alımı
- Temizlik ve ikram hizmetlerinin sağlanması
- Tuvalet ve su ikmal işlemi
- Uçağa yolcu alım işleminin (boarding) başlaması
- Kargo ve bagajların emniyetli bir şekilde yüklenmesi

- Uçak kapılarının kapatılması
- Pushback hizmetinin sağlanması

Bu operasyonel çevrim süreçleri belli prosedürlerle belirlenmiş ve bu prosüderlere uyulması otoriteler tarafından zorunlu hale getirilmiştir. Operasyonel sürecin herhangi bir kısmında yaşanan gecikme uçtan uca bütün süreci etkileyerek uçuş planında gecikmeye sebep olabilir. Ayrıca bu durum sadece ulaştırma hizmetindeki bir aksaklık olmayıp, bunun yanında yolcular için seyehat deneyimindeki memnuniyetini de olumsuz olarak etkilemektedir. Bu süreçler halihazırda farklı sistemlerden gelen bilgiler ve gözle yapılan değerlendirmeler ile takip edilmektedir. Gözle gözetim çoğu zaman zahmetli ve zaman alıcı bir işlemdir. Bu sebeple bu işlemlerin otomatik bir şekilde yürütülmesi birçok açıdan avantaj sağlayacaktır.

Bu bölümün devamında sırasıyla tezin amacı,literatür çalışmaları ve hipotez anlatılacaktır.

1.1 Literatür Özeti

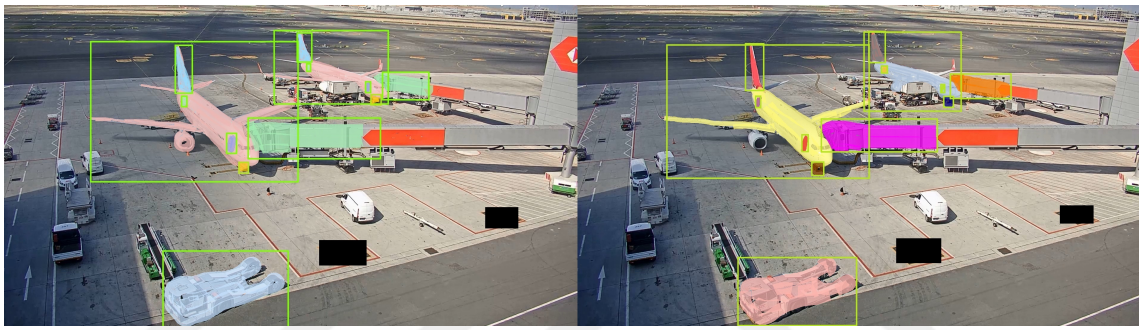
Tez çalışması yapılırken nesne tespiti, nesne takibi ve apronlarda uçak operasyonları ile ilgili yapılmış önceki çalışmalar için literatür taraması yapılmıştır. Nesne takibinde genellikle nesne tespiti alt görev, olduğu için nesne takip çalışmaları içerisinde nesne tespiti çalışmaları da bulunmaktadır.

1.1.1 Nesne Tespiti

Nesne tespitinin yanı sıra bu çalışmada uçak ve uçak parçaları da belirlendiği için nesne ve nesne parçalarının tespiti konusundaki çalışmalarda incelenmiştir. Ancak parça özeline bir geliştirme yapılmamış olup genel nesne tespiti modülüne göre uçak parçaları da tespit edilmiştir.

Parça veya nesne tespiti için çeşitli yöntemler incelenmiştir. [1] çalışmasında bir nesnenin anlamsal parçalarını tespit etmeye yönelik bir yaklaşım sunulmuştur. 3B CAD modelleri kullanılarak büyük veri kümeleri oluşturmak yerine küçük etiketlenmiş bir veri kümesinden öğrenebilen bir sistem önerilmiştir. Araba verisi üzerinde tekerlek, kapı, cam, plaka gibi bölümler etiketlenmiştir. Sistemdeki en önemli bileşen eşleştirme algoritmasıdır. Eğitim sırasında bölgesel öznitelikler çıkarılır, gerçek görüntü 3B CAD model ile benzer bakış açısına göre eşleştirilmektedir. Test aşamasında ise yine eşleştirme algoritması kullanılarak 3B modeldeki anlamsal parçalar 2B resimler ile eşleştirilmektedir. Evrişimsel sinir ağı modellerinden VGGNet [2] kullanılmıştır.

Benzer çalışmalardan biri [3], 2B görüntüleri bölümlere ayıran ve birden fazla bakış açısından gelen bilgileri bitkinin 3B nokta bulutu temsilinde birleştiren çoklu görüş kamera sistemi ile birlikte derin öğrenmeye dayalı bir bitki bölümü bölümleme yöntemi önerilmiştir. Bitkinin gövde, yaprak ve düğüm parçaları etiketlenmiştir. Bu çalışma bölütleme işlemi Mask R-CNN [4] kullanılarak yapılmıştır. Bölütleme işlemi için hem anlamsal (semantic) hem de örnek (instance) bölütleme kullanılmıştır. Anlamsal bölütleme, görüntüdeki tüm nesneleri piksel düzeyinde algıyarak tüm nesneleri ayrı ayrı gruplamaktadır. Örnek bölütlemede anlamsal bölütlemeye benzemektedir. Anlamsal bölütleme pikselleri nesnelerin sınıflarına göre gruplandırırken, örnek bölütleme her bir nesneyi sınıfından bağımsız olarak ayrı şekilde gruplandırır. Bu sebeple örnek bölütlemede aynı sınıftaki nesneler birbirinden farklı renge sahiptir (Şekil 1.1).



Şekil 1.1 (sol) Anlamsal Bölütleme, (sağ) Örnek Bölütleme

CCTV kameraları ile yapılan çalışmalardaki yöntemler incelenmiştir. Gözetleme kameralarında alınan görüntülerin analizi ve anlamlı hale getirilmesi gözle değerlendirildiğinde iş gücü ve zaman açısından dezavantaj oluşturmaktadır. Buna yönelik bu çalışmada [5] trafikten alınan görüntüler ile çeşitli kaza tiplerini tespit eden bir model önerilmiştir. Modelde nesne tespiti için Yolo v3 [6] kullanılmıştır. Ek olarak düşük görüşlü şartlar için resimleri iyileştirmeye yönelik Retinex kullanılarak tespit etme algoritmasına iyileştirilmiş görüntü girdi olarak verilmiştir. Ayrıca sonuç bölümünde de belirtildiği üzere tespit etme işleminin doğruluğu, modelin performansını etkilediği belirtilmiştir.

Tezin 2. bölümünde nesne tespiti için kullanılan yöntemler detaylı olarak incelenmiştir.

1.1.2 Nesne Takibi

Hareket tespitine yönelik bu çalışmada [7] toplantıların kaydedilmesi ve analiz edilmesi konusu üzerinde çalışılmıştır. Toplantı sırasında tepede yer alan kamera ile görüntüler üzerinden anlamlı çıktılar elde edilmeye çalışılmıştır. Bunun için

toplantıda yer alan kişilerin hareketleri izlenmiştir. Hareket tespitinde temelde çerçeveler arası fark ve Shi-Tomasi köşe tespitini birlikte kullanan yeni bir yöntem önerilmiştir. Öncelikli olarak önerilen yöntemle hareket bölgesi çıkartılmış ve sonrasında köşe tespiti uygulanmıştır. Önerilen yöntemde yavaş hareket eden nesneler için büyük pencere boyutları kullanılırken, hızlı hareketler için küçük pencere boyutları kullanılması yaklaşımı önerilmiştir.

Bu çalışmada [8] insan takibi derin öğrenme yöntemi ile yapılmıştır. Veri kümesine insan vücut parçaları eklenerek Faster R-CNN [9] ile tekrar eğitilmiş ve veri kümesi çeşitlendirilmiştir. Derin öğrenme tabanlı nesne tespit ağı izleme algoritması ile birleştirerek çoklu nesne izleyici yapılmıştır. Takip etme için temel olarak SORT [10] ve DeepSORT [11] modelleri esas alınmıştır. Bundan dolayı sistem içerisinde Kalman filtresi ve nesne ilişkilendirmek için ise Hungarian algoritması kullanılmıştır. İzleme doğruluğunu artırmak için en yakın renk ve histogram gibi diğer özelliklerden de yararlanılmıştır. İnsan üzerinde önerilen yöntemin SORT ve Deep SORT'dan daha başarılı sonuçlar aldığı belirtilmiştir.

Bunların yanında son yıllarda kullanılan takip etme ve tespit etme yöntemleri anket çalışmaları da incelenmiştir. Nesne tespitinde SSD [12] (Single Shot Detector) ve YOLO [13] gibi farklı yöntemler bulunmaktadır, ancak çoğunlukla Faster R-CNN ve türevlerinin kullanıldığı belirtilmiştir [14]. Nesne takibinde de yine SORT, SORT'un eksikliklerine yönelik geliştirilen Deep SORT, Siyam ağları ve LSTM gibi mimarilerden bahsedilmiştir. SSD tespiti kullanan algoritmalar hızlı olmasına karşın daha kötü performans eğilimde olduğu belirtilmiştir. Nesne takibi için gerekli özellik çıkarımı için ise en iyi performansın görünüm özellikerini çıkaran CNN mimarileri olduğu belirtilmiştir. Ancak yine de sadece görsel özelliklerin yeterli olmadığı bunun yanında hareket (motion) ile ilgili bilgilerin kullanılabileceğini belirtmektedir. LSTM ve Kalman filteresi gibi algoritmalar ile hedefin bir sonraki konumunu tahmin edilerek ilişki kurmaya iyileştirici katkı sağlayabileceği belirtilmiştir. Çeşitli Bayes filtreleri kullanılarak da hedefin hareketinin tahmin edilebileceği belirtilmiştir. Bununla birlikte görsel olmayan özelliklerle birlikte kullanıldığında bile görünüm özelliklerinden yararlanmanın önemli bir rol oynadığı belirtilmiştir.

1.1.3 Apron Görüntülerini Kullanan Çalışmalar

Havaalanlarının veya apronların izlenmesine yönelik az sayıda çalışma bulunmaktadır. Bu çalışmada [15] pist üzerinde uçakların konumlarını tespit eden ve ayrıca güvenlik yönetimini sağlamak amacıyla uçakların gerçek zamanlı olarak hız ve mesafe analizini sağlayan bir model kurulmuştur. George Bush Kıtalararası Havaalanı'ndaki dijital kuleden alınan videolarının doğrulamaları ile simüle edilmiş veriler kullanılmıştır.

Uçakların tespiti için özelleştirilmiş evrişimsel sinir ağı mimarisi kullanılmıştır. Nesne takibi öncesinde nesnenin el ile işaretlenmesi veya harici bir algoritma ile bulunması gerekmektedir. Bu nedenle takip etmenin bir diğer yoluda tespite dayalı takiptir. Kuleden alınan görüntülerde görüş açısından dolayı nesneler çakışmadığından basit bir takip etme yöntemi kullanılmıştır. Her çerçevedeki uçaklar tespit edilip, sınırlayıcı kutuların bir önceki çerçeve ile karşılaştırılarak eşlenmesi yöntemi ile takip yapılmaktadır. Özelleştirilmiş evrişimsel sinir ağı, AirNet olarak adlandırılmıştır. Evrişimsel sinir ağı ile benzer performansa sahip ancak daha hızlı bir model sunulmuştur. Oluşturulan modele iki yönlü özellik piramit ağı (bidirectional feature pyramid network, BiFPN) yapısı entegre edilmiştir. Evrişimsel sinir ağı modellerinden olan Retina, YOLOv3, EfficientDet ile sunulan yöntemin çalışma zamanı, parametre sayısı ve ortalama tahmin süreleri karşılaştırılmıştır. Bu değerlendirmeye göre sunulan modelin parametre sayısının az ve daha hızlı olması avantaj olarak sunulmuştur.

Bu çalışmada [16] birden çok kamera kullanılarak apronu izleme ve olay tanıma işlemleri yapan uçtan uca bir sistemden bahsedilmiştir. Aprondaki görüntüleri kullanarak iki ana modül geliştirilmiştir. Birincisi, apron üzerindeki nesnelerin tespiti, takibi ve sınıflandırılmasından sorumlu iken, diğeri apron faaliyetlerini anlamaktan sorumludur. Her çerçevede nesne takip ve kategorize edilmektedir. Nesne sınıflandırılmasında çok aşamalı sınıflandırma yaklaşımı benimsenmiştir. Yani öncelikle insan, yer araçları, uçak şekilde sınıflandırma yapılıp, sonrasında yakıt aracı, kargo aracı gibi alt tip araçlar belirlenmektedir. Olaylar apronun belirli bölgelerinde yapıldığından bölge-araç-insan durumları ve etkileşimleri analiz edilerek çıktı üretilmektedir. Görüntülerdeki enerjilendirme (GPU) aracı, yakıt aracı ve yükleme aracı ile ilgili operasyonlar takip edilmiştir. Nesne takibi için KLT [17] (Kanade-Lucas-Tomasi) algoritması kullanılmıştır. Olayları tanıma modülü ise önceden tanımlanmış bölge-araç-insan bilgilerini kullanarak yorumlamaktadır. Apron bölgeleri, operasyonun yapıldığı bölgeye göre ayrılmıştır. Böylece ilgili nesnenin o bölgeye girişi ve çıkışı ile ilgili bilgi edinilmiştir. Olayların belirlenmesinde başarılı sonuçlar elde edilmiştir, ancak aprondaki farklı durumlara karşı duyarlı olduğu görülmüştür.

1.2 Tezin Amacı

Apronda uçaklar iniş yaptıkları andan itibaren bahsedilen operasyonlara tabi tutulurlar. Yapılan işlemlerin ne zaman yapılmaya başlandığı, ne zaman bittiği, ne kadar sürdüğü gibi bilgiler hem güvenlik hem de sonrasında yapılan değerlendirme hususunda önem arz eden bilgilerdir.

Yer hizmetleri operasyonel çevrim süreclerinin herbiri için ne şekilde yapılması gerektiği, hangi durumlarda ne yapılması gerektiği gibi bilgiler ayrıntılı olarak prosedürlerle belirlenmiştir. Özellikle eş zamanlı gerçekleşen aktivitelerde kurallara uyulması çok önemlidir. Aksi takdirde ölümle veya yaralanmayla ve maddi kayıplarla sonuçlanabilecek tehlikere yol açabilmektedirler [18]. Emniyetli bir operasyon için bu süreçler park pozisyonunda belli alanlardan yapılmaktadır. Bu süreçler belli sıralarla yapılacağı gibi aynı anda yapılabilen işlemler de bulunmaktadır. Uçağın hizmet sürecinde, yer hizmeti ekipmanı ile yapılan birçok aktivite vardır. Uçak indikten sonra temel olarak park pozisyonuna uçak karşılama (marshalling) veya otomasyonlu sistem ile gelmektedir. Daha sonra takozlama işlemi yapılmaktadır. Yolcuların merdiven veya körük ile terminale ulaşması sağlanmaktadır. Gerekiyorsa enerjilendirme ve havalandırma işlemi ve kargoların boşaltılması işlemi gerçekleşmektedir. Yakıt ikmali, ikram yüklemesi, temiz su ve uçak tuvalet hizmeti sağlanmaktadır. Gerekli işlemler bittikten sonra yeni sefer için yolcuların binmesi ve kargoların yüklenmesi sağlanıp pushback aracı ile geri itme hizmeti sonrasında kalkış için taksi durumuna geçmektedir. Genel olarak bu süreçlerin bazıları şekil 1.2'da gösterilmiştir. Sıralı işlemlere örnek olarak, uçağın parketmeden takoz yerleştirilememesi veya temizlik işlemi tamamlanmadan yolcuların uçağa alınmaması verilebilir. Fakat örneğin yakıt alımı yapılırken ikram hizmetleri sağlanabilir veya kargo işlemleri gerçekleşebilmektedir.

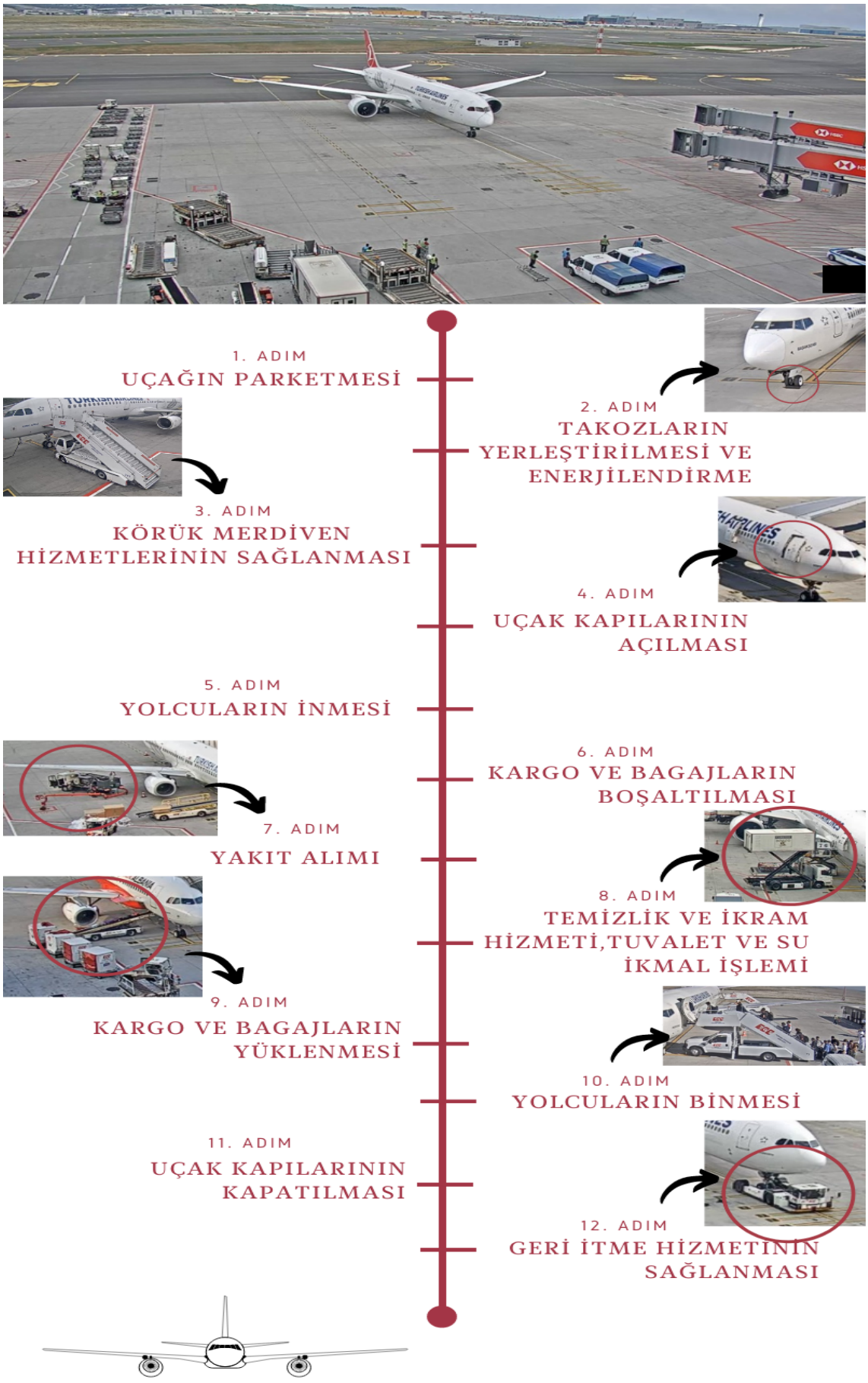
Bu tez çalışmasında Türk Hava Yolları Teknoloji AR-GE merkezinden alınan apron görüntüsü verileri kullanılarak operasyonel çevrim süreçlerinin doğru ölçümü ve optimizasyonunu amaçlayan, video görüntülerini değerlendirerek tespit, takip ve kontrol eden derin öğrenme tabanlı bir sistem tasarlanmış ve gerçeklenmiştir. Bu kapsamda ortak kullanıma açık bir veri kümesi olmadığı için operasyonların tamamı değil, sadece sağlanan görüntülerde bulunan operasyonlar değerlendirilmiştir.

1.3 Hipotez

Literatürde apronlarda uçaklara verilen hizmetlerin değerlendirilmesi ile ilgili yapılan çalışmalar değerlendirildiğinde, önerdiğimiz sistemin mevcut çalışmalardan farklı olarak derin öğrenme tabanlı yöntemlerle nesne tespiti, takibi ve hareket değerlendirmesi yapan ilk sistem olduğu görülmektedir. Ayrıca gerçekleştirilen sistem gerçek zamanlı işlem yapabilmektedir.

Bu tez çalışmasında diğer çalışmalardan farklı olarak yapılan genel işlemler şunlardır:

- Örnek bölütleme kullanılarak uçak parça bölümlerini ve operasyonda yer alan bazı araçları derin öğrenme ile tespit eden



Şekil 1.2 Operasyonel Süreçlerden Bazıları

- Tespit ettiği nesneleri derin öğrenme yöntemi ile takip eden
- Tespit veya takip ettiği nesnelerin durumları ve konumlarına göre çıkarımlarda bulunabilen bir sistem tasarlanmış ve gerçeklenmiştir.

2
NESNE TESPİTİ VE NESNE TAKİBİ

Bu çalışmada uçağa verilen yer hizmetlerinin değerlendirilmesi için video görüntüleri kullanılarak nesne tespiti ve nesne takibi yapılmıştır. Bu işlemleri gerçekleştirebilmek için evrişimsel sinir ağı temelli yöntemler kullanılmıştır. Bu sebeple bu bölümde önce evrişimsel sinir ağı anlatılacak ardından tez çalışmasında kullanılan başarılı nesne tespiti ve nesne takibi yöntemlerinden bahsedilecektir.

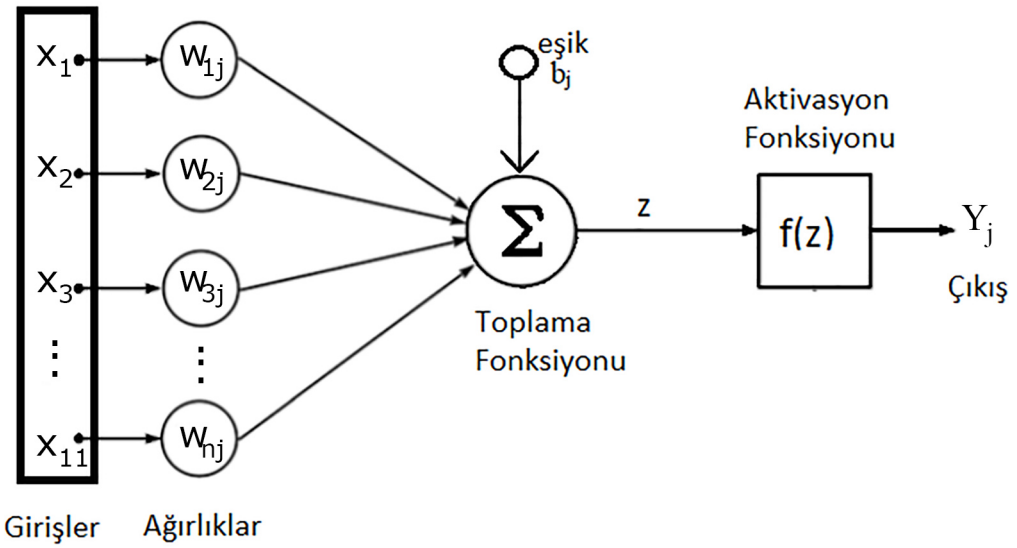
2.1 Derin öğrenme

Yapay Sinir Ağları (YSA), beynin çalışma yönteminden yola çıkarak biyolojik nöronların basitleştirilmiş şekilde matematiksel olarak modellemesine dayanmaktadır [19]. YSA'nın temel birimi nöron adı verilen işlemci elemanlarıdır. Nöron yapısı şekil 2.1'de gösterilmektedir. Girdiler X sembolü ile, bilginin önemi ve nöron üzerindeki etkisini gösteren ağırlıklar ise W ile gösterilmektedir. Toplama fonksiyonu, girdi ve ağırlık değerlerinin çarpımlarının toplamı ile bulunmaktadır (Eşitlik 2.1) [20]. Toplama fonksiyonu sonucunda doğrusal veya doğrusal olmayan türevlenebilir aktivasyon fonksiyonundan geçirilerek çıktı hesaplanmaktadır (Eşitlik 2.2). Derin öğrenme, yapay sinir ağlarını temel alan makine öğrenmesinin bir alt kümesidir ve YSA'lardaki gizli katman sayılarının arttırılması sonucu derinleşen ağlardır [21].

$$Z_i = \sum_{i=1}^{n} (W_{ij} X_i + b_j) \tag{2.1}$$

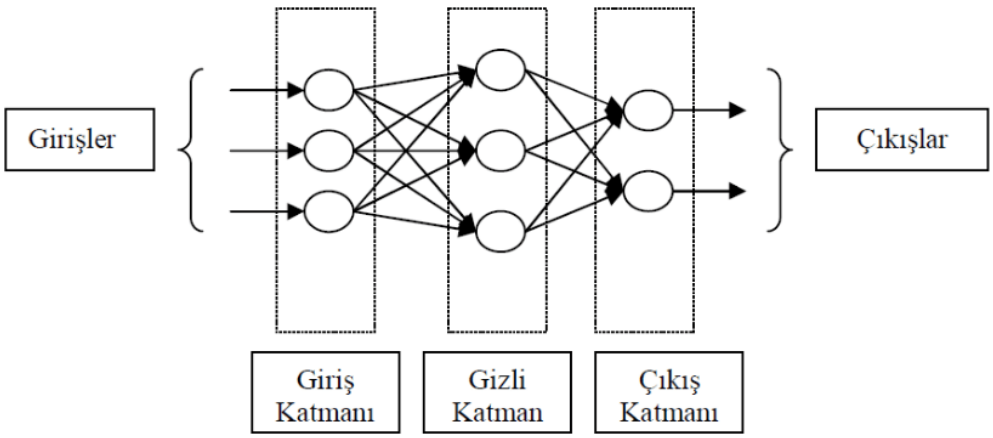
$$y_j = f(Z_i) = f\left(\sum_{i=1}^{n} (W_{ij} X_i + b_j) \right) \tag{2.2}$$

Çok katmanlı YSA mimarisi şekil 2.2'da gösterilmiştir. Her sıralı katman önceki



Şekil 2.1 Tek katmanlı yapay sinir ağı

katmandaki çıktıyı girdi olarak alır [22]. Çıkış işlemi sonucunda elde edilen çıkış değer ile gerçek değer karşılaştırılır ve kayıp değeri elde edilir. Eşitlik 2.3'de [23] kayıp değerinin hesaplaması gösterilmektedir. Elde edilen değerler, geriye doğru diğer katmanlara yayılır. Hatayı (kaybı) en aza indirmek amacıyla, ağırlıkları güncellemek için optimizasyon yöntemi kullanılır. Bu işlemde hatanın ağırlıklara göre kısmı türevi, zincir kuralı ile yapılır (Eşitlik 2.4). Ağırlıkların güncelleştirme işlemi, modelin öğrenme süreci olarak ifade edilmektedir (Eşitlik 2.5). Model eğitilirken iterasyon ilerledikçe kayıp değerinin düşmesi modelin öğrendiği olarak anlaşılabilemktedir.



Şekil 2.2 Çok katmanlı temel yapay sinir ağı mimarisi

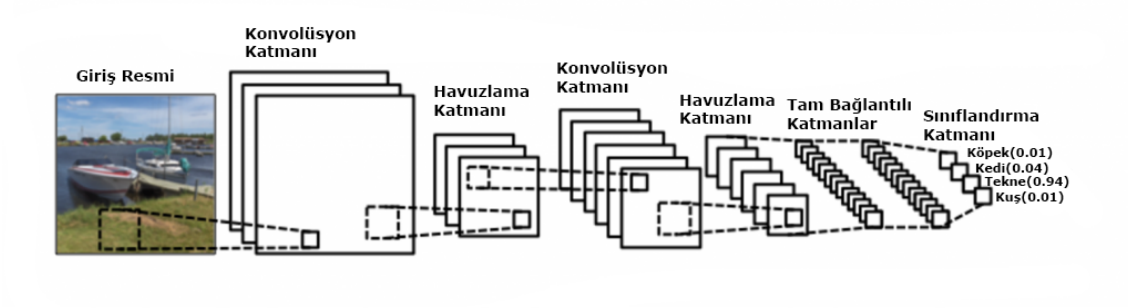
$$E = \frac{1}{2}\sum_{i=1}^{n}(y_{gercek} - y_{tahmin})^2 \tag{2.3}$$

$$\frac{\partial E(t)}{\partial W_{kj}^2(t)} = \frac{\partial E(t)}{\partial n_k^2(t)} \frac{\partial n_k^2(t)}{\partial W_{kj}^2(t)} \tag{2.4}$$

$$W^{l+1} = W^l - \alpha \frac{\partial E}{\partial W^l} \tag{2.5}$$

2.1.1 Evrişimsel Sinir Ağı

Evrişimsel Sinir Ağı (ESA) mimarisi, girdi olarak görselleri alan sonrasında bunlara çeşitli filtreler uygulayarak özellikler elde eden ve bunları sınıflandıran bir algoritmadır. Katmanlı bir yapısı vardır (Şekil 2.3).



Şekil 2.3 Evrişimsel sinir ağının genel mimarisi [24]

Evrişimsel katman, ESA'nın temel katmanlarından biridir [21]. Bu katmanda orjinal görsel boyutlardan daha küçük farklı boyutlarda filtreler uygulanmaktadır. Birden fazla özelliğin tespit için genelde birden fazla evrişimsel katmanı uygulanmaktadır. Sonucunda özellik haritası (Feature Map) elde edilir. Genelde evrişimsel katmanlarından sonra doğrusal olmayan katman gelmektedir. Bu katman, ağı doğrusal olmayan bir yapıya sokmak için kullanılmaktadır. Kullanılan fonksiyon genelde ReLu (Rectified linear unit) fonksiyonudur. ReLu fonksiyonu özellik haritasına uygulandığında özellik haritasındaki sıfırdan küçük değerleri sıfıra getirmekte ve ağın daha hızlı öğrenmesini sağlamaktadır. Havuzlama (Pooling) katmanı, evrişimsel katmanlar arasına daha küçük çıktılar elde etmek amacıyla yerleştirilmektedir. Düzleştirme (flattening) katmanı, evrişimsel ve havuzlama katmanından gelen matrisleri tek boyutlu diziye çevrilmektedir. Tam Bağlantılı Katmanında (Fully Connected Layer), seçilen matris boyutuna göre ağırlık matrisi oluşmaktadır. Düzleştirme sonucu üretilen her bir nöron buradaki seçilen matris boyutu kadar nörona bağlanmaktadır. Dropout Katmanında, ağın ezberleme yapmasının önüne geçmesi için bazı nöronlar kaldırılmaktadır. Girdi resmi, sinir ağları ile eğitebilecek

şekile geldikten sonra klasik sinir ağları yöntemiyle ağırlık ve eşik (bias) değiştirilerek öğrenme sürecine girilmektedir.

2.1.2 Hiper parametreler

Derin öğrenmede çok katmanlı yapıların tasarım mimarisi, katman sayısı, kullanılacak olan optimizasyon veya aktivasyon fonksiyonları, nöron sayısı gibi parametreler başarıyı etkileyen parametrelerdir [25]. En iyi sonucu elde edebilmek için optimum şekilde çok katmanlı ağ yapısı tasarlamak önemlidir. Burada dikkat edilecek en önemli noktalardan biri kullanılan hiper parametrelerdir.

Hiper parametreler, sabit kullanılan değerler değildir. Kullanılan veri kümesine ve probleme göre sezgisel olarak başlanabilir. Çünkü başlangıçta bu parametrelerin değerleri açık değildir. Model birbirinden farklı hiper parametrelerle başarılı olabilir, ancak en uygun parametrelerin seçilmesi modelin başarısını etkilemektedir.

Veri kümesi, mini küme (mini-batch) boyutu, öğrenme hızı, optimizasyon algoritması, iterasyon sayısı, aktivasyon fonksiyonu hiper parametreler arasındadır.

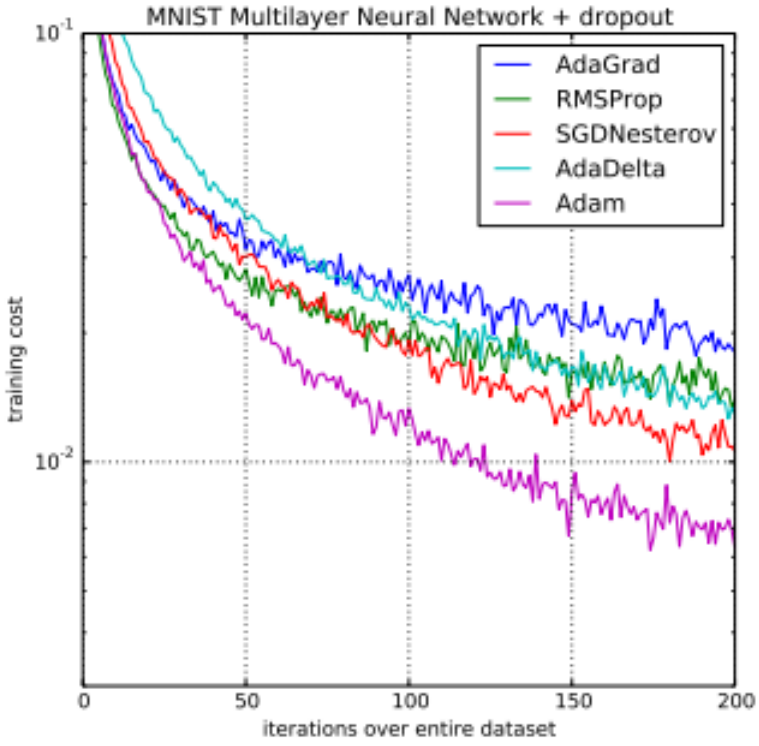
Veri kümesinin büyüklülüğü ve çeşitliliği modelin başarısı için çok önemlidir. Ancak veri kümesi büyüklüğü yanında depolama ve zaman açısından kaybı bulunmaktadır. Probleme göre uygun veri kümesi oluşturulmalı ve model bu veri kümesiyle eğitilmelidir.

Veri kümesindeki tüm verilerin aynı anda işlenmesi maliyet açısından problemlere yol açmaktadır. Her **iterasyondan** geri yayılım ile ağırlık değerleri güncellenmektedir. Bunu tüm veri için yapmak maliyetli bir işlemdir. Dolayısıyla eğitim örneklerinin küçük parçalar haline işlenmesi **mini-küme** olarak adlandırılmaktadır. Yani modelin aynı anda kaç veriyi işleyeceği bu parametre ile belirtilmektedir. Mini-küme kullanıldığında her iterasyonda farklı veri işleneceğinden ve bazı iterasyondaki veriler için parametreler daha uygun olurken bazıları için tam uygun olmayacağından gürültü durumu yani grafikte zikzaklar olarak görülebilecek bir durum ortaya çıkabilmektedir. Gürültüyü azaltmak için öğrenme değeri (learning rate) düşürülebilinir. Ancak iterasyonlar ilerledikçe kayıp değerinin düşmesi beklenmektedir. Kayıp değerinin düşmesi başarının arttığı anlamına gelmektedir. Dolayısıyla iterasyon arttıkça modelin başarısının artması beklenmektedir. Başarı belli bir adımdan sonra modelin öğrenme durumu çok azalacaktır. Bu noktada eğitim sonlandırılabilir.

Öğrenme hızının yüksek olması veriden fazlaca etkilenildiğini belirtmektedir. Genelde belli bir iterasyon sonrasında düşürülerek gidilir. Eğer öğrenme hızı çok yüksek olursa salınım fazla olacaktır, tersi durumda ise çok küçük adımlarla

ilerleyeceğinden zaman açısında dezavantaj oluşturacaktır. Ayrıca çok küçük olması durumunda istenilen global optimum noktasına hiç ulaşılmadan lokal optimum değerde takılmasına sebep olabilmektedir.

Ağırlıkların doğru şekilde güncellenmesi öğrenme için önemlidir. Elde edilen hataya göre geri yayılım ile ağırlıklar güncellenir. Geri yayılım ile ağdaki her bir ağırlığın hesaplanan hataya olan etkisi için yaygın olarak stokastik gradyan azalma (stochastic gradient descent) [26], adagrad, adadelta [27], adam [28], adamax gibi **optimizasyon algoritmaları** kullanılmaktadır [29]. Bunlar arasında hız, başarım, bellek gereksinimi bakımından farklılıklar bulunmaktadır. MNIST veri seti üzerinde bazı optimizasyon algoritmalarının çalışma zamanı şekil 2.4'da gösterilmiştir.



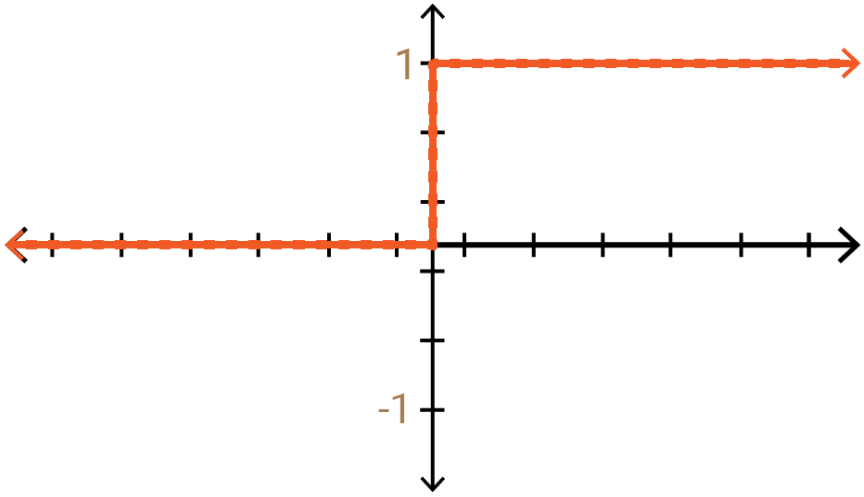
Şekil 2.4 Bazı optimizasyon algoritmalarının MNIST üzerinde çalışma zamanı grafiği [28]

Geri türev alınabilmesi için gizli katmanda nöronların ağırlığı hesaplandıktan sonra gizli katmanların çıktısı aktivasyon fonksiyonları ile normalize edilmektedir. Bu fonksiyonlar, çok katmanlı ağlarda doğrusal (linear) ifadeleri doğrusal olmayan ifadelere dönüştürmek için kullanılır. Kullanılan **aktivasyon fonksiyonları** ikili basamak fonksiyonu[30], doğrusal aktivasyon fonksiyonu, sigmoid[31, 32], Tanh, ReLU, Leaky ReLU, softMax olarak listelenebilir.

İkili basamak fonksiyonu, eşik değer temelli sınıflandırıcı olarak ve genellikle çıkış

katmanında kullanılmaktadır. Birden fazla sınıflandırma durumlarında yapısı gereği kullanılamamaktadır. Ayrıca geri yayılım esnasında f(x)'in x'e göre türevi 0'a eşit olacağından bu durum geri yayılımda soruna sebep olabilmektedir [30]. İkili basamak fonksiyonunun, matematiksel eşitliği, eşitlik 2.6'da, grafiği ise şekil 2.5'de gösterilmiştir.

$$f(x) = \begin{cases} 1, & x \geq 0 \text{ ise} \\ 0, & x < 0 \text{ ise} \end{cases} \tag{2.6}$$

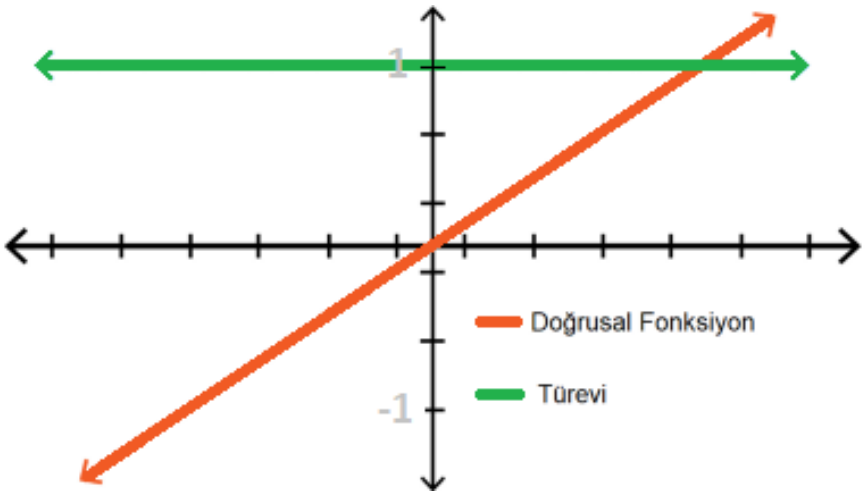


Şekil 2.5 İkili basamak fonkiyonu

Doğrusal aktivasyon fonksiyonu, grafiği şekil 2.6'da gösterilmektedir. En önemli sorun eşitlik 2.7'ye bakıldığında x'e göre türev alındığında sabit bir c elde edilmesidir. Dolayısıyla türevi hep sabit bir değer çıktığından ağırlıklar doğru şekilde güncellenmeyeceği için öğrenme işlemi performanslı bir şekilde yapılamamaktadır. Basit görevler dışında kullanılması çok tercih edilmezler.

$$f(x) = ax \tag{2.7}$$

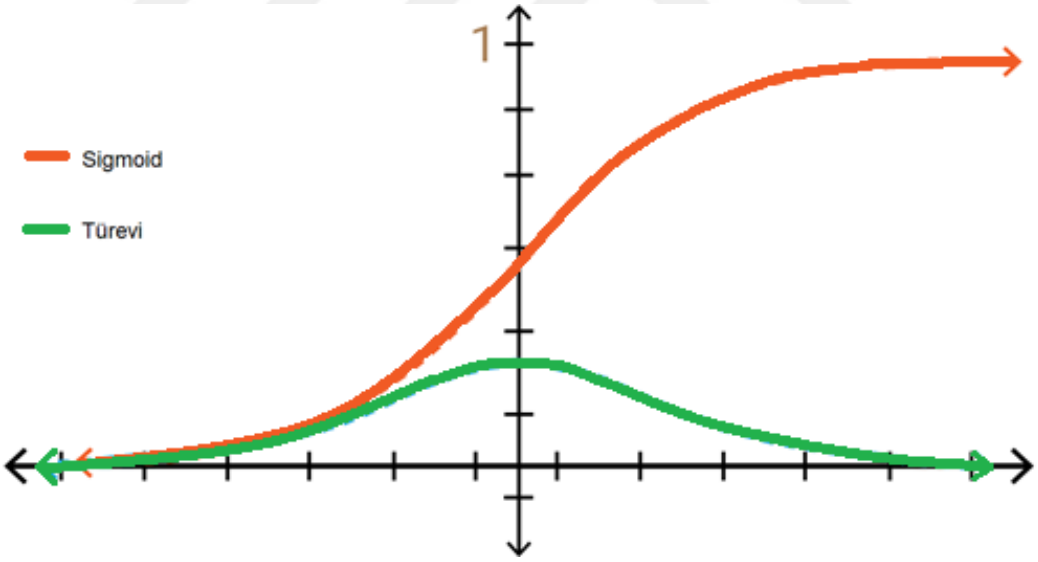
Sigmoid aktivasyon fonksiyonu, şekil 2.7'da ki grafikten görüldüğü üzere x değeri -2 ve 2 arasındayken y değerleri hızlı bir şekilde değişmektedir. Bu iyi bir sınıflandırıcı olarak kullanılabileceği anlamına gelmektedir. Türevlenebilir bir fonksiyon olması sebebiyle de öğrenme esnasında sık kullanılan yöntemlerden biridir. Bu fonksiyonun sorunu, x değerinin büyük olduğu yerlerde geri yayılım (back propagation) esnasında türev 0'a yakınsayacağı için gradyan kaybolması sorunu olabilmektedir. Böylesi bir durumda öğrenme minimum düzeyde gerçekleşeceği için modelden beklenen



Şekil 2.6 Doğrusal aktivasyon fonksiyonu

performans alınanamaktadır. Sigmoid aktivasyon fonksiyonunun eşitliği 2.8'de gösterilmiştir.

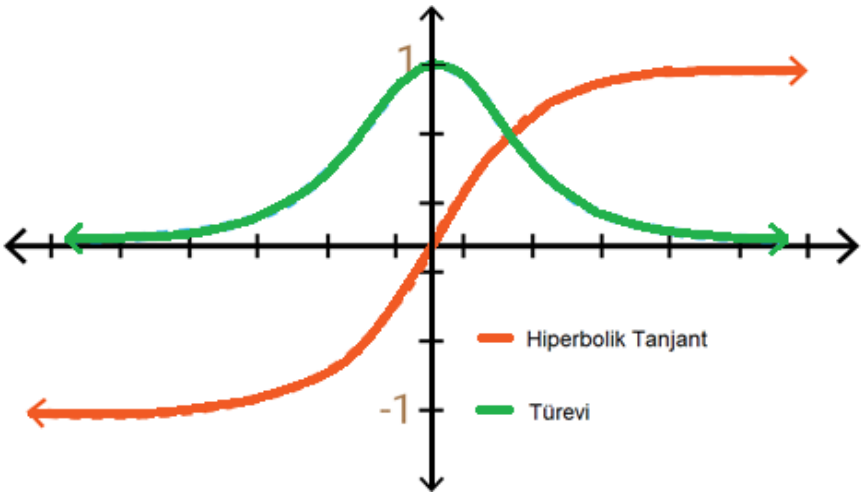
$$f(x) = \frac{1}{1+e^{-x}} \tag{2.8}$$



Şekil 2.7 Sigmoid aktivasyon fonksiyonu

Tanh fonksiyonu, sigmoid fonksiyonu ile benzer bir yapıya sahiptir. Tanh fonksiyonunun, grafiği şekil 2.8'da, matematiksel gösterimi ise eşitlik 2.9'da gösterilmektedir. Değer aralığı genişlediği için daha hızlı öğrenme sağlanabilecektir. Ancak gradyan kaybolması sorunu, tanh fonksiyonunda devam etmektedir.

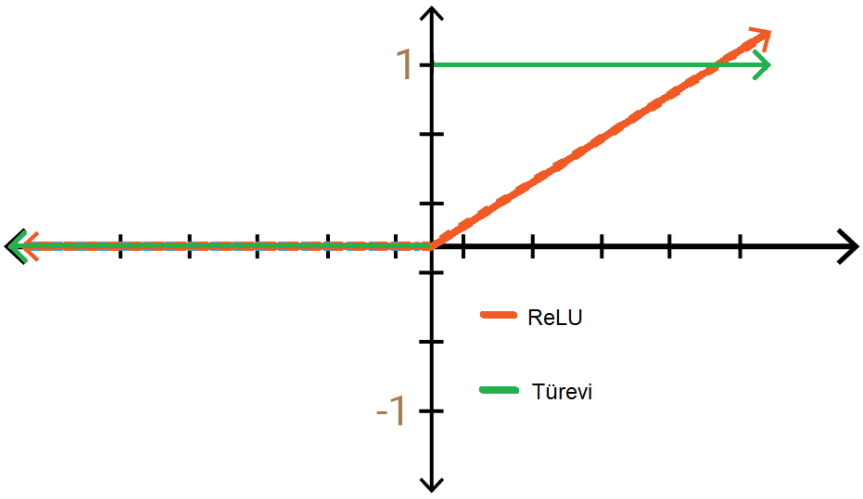
$$tanh(x) = 2sigmoid(2x) - 1 \tag{2.9}$$



Şekil 2.8 Tanh aktivasyon fonksiyonu

Rectified Linear Unit (ReLu) fonksiyonu, tüm nöronlar aynı anda etkinleştirilmeyip tek seferde belirli sayıda nöron etkileştirildiğinden dolayı diğer fonksiyonlardan daha etkilidir. x'in negatif değerleri için 0 değerini alması ağın hızlı çalışmasını sağlar, ancak bu durumun dezavantajı bulunmaktadır. Değerlerin 0 olması durumunda türev 0 olacaktır, yani geri yayılım sırasında ağırlık güncelleme yapılamayacağı için öğrenme bu bölgelerde yapılamayacaktır. ReLu fonksiyonunun grafiği şekil 2.9'da, matematiksel gösterimi ise eşitlik 2.10'da gösterilmektedir.

$$f(x) = max(0, x) \tag{2.10}$$

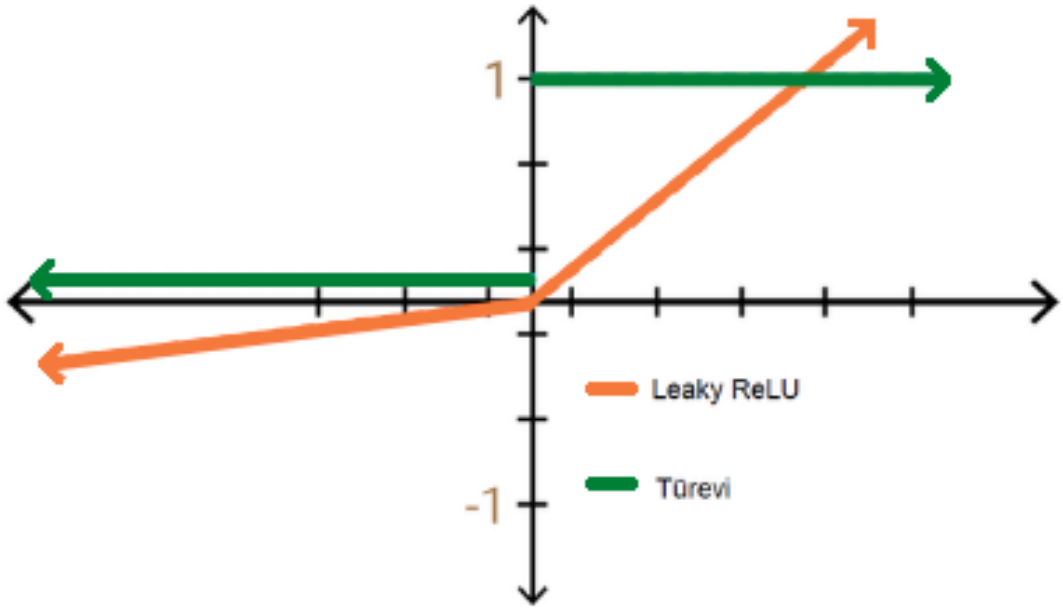


Şekil 2.9 ReLu aktivasyon fonksiyonu

Sızıntı(Leaky) ReLU fonksiyonu, ReLu fonksiyonun biraz daha geliştirilmiş bir versiyonudur. x'in negatif değerleri için 0 yerine çok küçük x doğrusal bileşeni olarak tanımlanır. Böylece o bölgelerde de öğrenme sağlanır. Sızıntı ReLU fonksiyonunun

grafiği şekil 2.10'da, matematiksel gösterimi ise eşitlik 2.11'de gösterilmektedir.

$$f(x) = \begin{cases} 0.01x, & x < 0 \text{ ise} \\ x, & x \geq 0 \text{ ise} \end{cases} \tag{2.11}$$



Şekil 2.10 Sızıntı aktivasyon fonksiyonu

Softmax fonksiyonu, softmax genellikle bir sınıflandırma ağının son katmanı için aktivasyon olarak kullanılmaktadır. Çünkü girdinin sınıf olasığını 0-1 aralığında bir değer olacak şekilde olasılık dağılımını vermektedir [30]. Bu sebeple olasılık değerlerinin toplamı 1'dir. Softmax fonksiyonunun, matematiksel gösterimi eşitlik 2.12'de gösterilmektedir.

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.12}$$

2.2 Nesne Tespiti

Geleneksek nesne tespiti modellerinde özellik çıkarmak için alçak seviyeli özellkler olan Haar-benzeri öznitelikler [33] (Haar-Like features), yönlü grandyan histogramı [34] (Histogram of oriented gradients, HOG), ölçek-değişmez özellik dönüşümü [35] (Scale-Invariant Feature Transform, SIFT) gibi yöntemlerden faydalanılmaktadır. Geleneksel yaklaşımlarda özellik vektörünün çıkarılması için alanında uzman kişilere

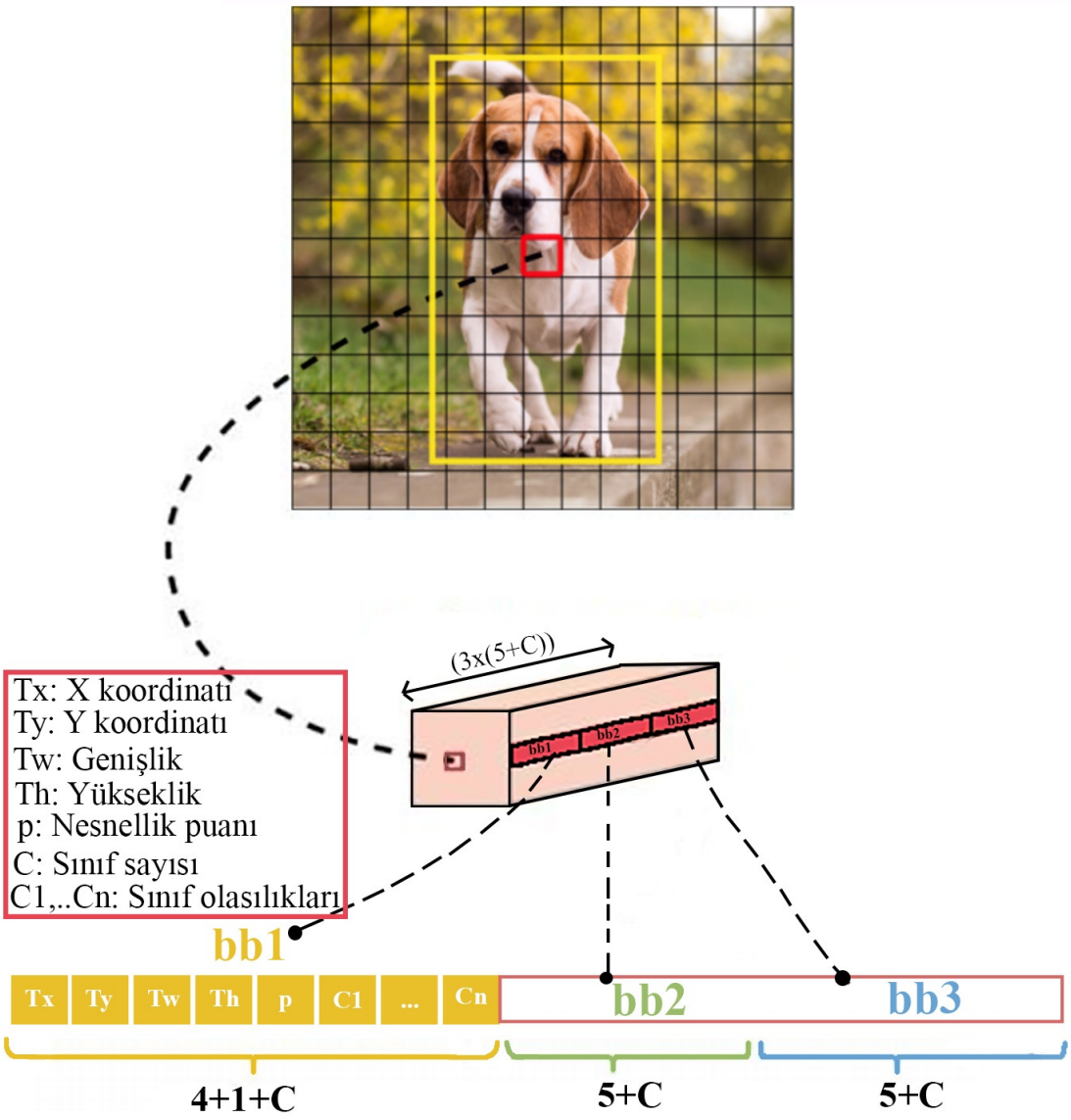
ihtiyaç duyulmakta [21], özellik çıkarma ve sınıflandırma işlemleri ayrı ayrı yapılmaktadır. Derin öğrenme yöntemlerinin devreye girmesi ve evrişimsel sinir ağlarının kullanılması ile etkileyici başarılar elde edilmiştir [36].

Nesne tespit modelleri genellikle tek aşamalı ve iki aşamalı olarak iki kategoriye ayrılabilmektedir. Tek aşamalı modeller içerisinde YOLO [13] ve SSD [12], iki aşamalı modeller arasında ise bölge tabanlı evrişimsel ağlarından Faster R-CNN [9] gibi algoritmalar değerlendirilmektedir. İki aşamalı nesne tespit modelleri yüksek doğruluk oranına sahipken, tek aşamalı tespit ediciler ise yüksek tespit hızına sahiptirler [37].

2.2.1 YOLO ile Nesne Tespiti

CNN kullanarak nesne tespiti yapan algoritmalardan biri You Only Look Once (YOLO) [13] algoritmasıdır. Resme tek bir CNN uygulamasından dolayı gerçek zamanda işlemeye uygundur. Girdi resmini $N \times N$'lik ızgaralara bölerek, her hücre için sınırlayıcı kutu ve nesnellik (objectness) puanı tahmini yapmaktadır. Nesnellik puanı, hücre içerisinde nesne olup olmadığını gösteren değerdir. YOLO'da herbir hücrede sadece bir nesne tanımlanabilmekte acak bir hücrede birden fazla nesne olması durumunu için YOLOv2'de [38] çapa (anchor) kutuları dahil edilmiştir. Çapalar ile nesnenin etrafındaki sınırlayıcı kutu tahmin edilmektedir. YOLOv2'de darknet-19 olarak adlandırılan mimari kullanılmaktadır. Ancak küçük nesnelerin tespiti özelliklerin kaybedilmesinden dolayı işlenenememektedir. Yolov3 [6]'de bu eksikliklere yönelik iyileştirmeler yapılmıştır. YOLOV3'de darknet'in türevlerinden biri olan darknet-53 mimarisi kullanılmaktadır. YOLOv3 ağının çıkışında 3 farklı ölçekte çıkış üretilmektedir. Bunun sebebi büyük ve küçük nesneleri belirleyebilmektir. 13, 26, 52 ölçekleri ele alındığında 13 büyük nesneleri, 26 orta büyüklükteki nesneleri, 52 ise küçük nesneleri tespit etmekten sorumludur. 13, 26, 52 farklı ölçekler ile her hücre (grid) için 3 sınır kutusu tahmini yapıldığında $3 \times (13^2 + 26^2 + 52^2) = 10647$ tane sınırlayıcı kutu elde edilmektedir. Her sınırlayıcı kutu 5+C (sınıf sayısı) özelliğe sahiptir (Şekil 2.11). Tek ölçek için her hücrede 3 sınırlayıcı kutu tahmini yapıldığı için $(3 \times (5 + C))$ değer elde edilmektedir. C sınıf olasılık değerlerinden maximum olanlar her sınırlayıcı kutu için belirlenmektedir. Bu değerler ile sınır kutularının nesnellik puanı çarpılır. Nesnellik puanı belirlenen eşik değerinden küçük olan sınır kutuları elenmektedir. Nesnellik puanına göre büyükten küçüğe doğru sıralandıktan sonra bir nesnenin birden fazla sınır kutusu olmaması amacıyla NMS (Non-Maximal Supression) uygulanmaktadır (Şekil 2.12). Non-max Suppression algoritması, görüntü üzerinde tespit edilen nesneler için çizilen sınırlayıcı kutulardan güven değeri en yüksek olanı ekrana çizmektedir. Hız ve doğruluk

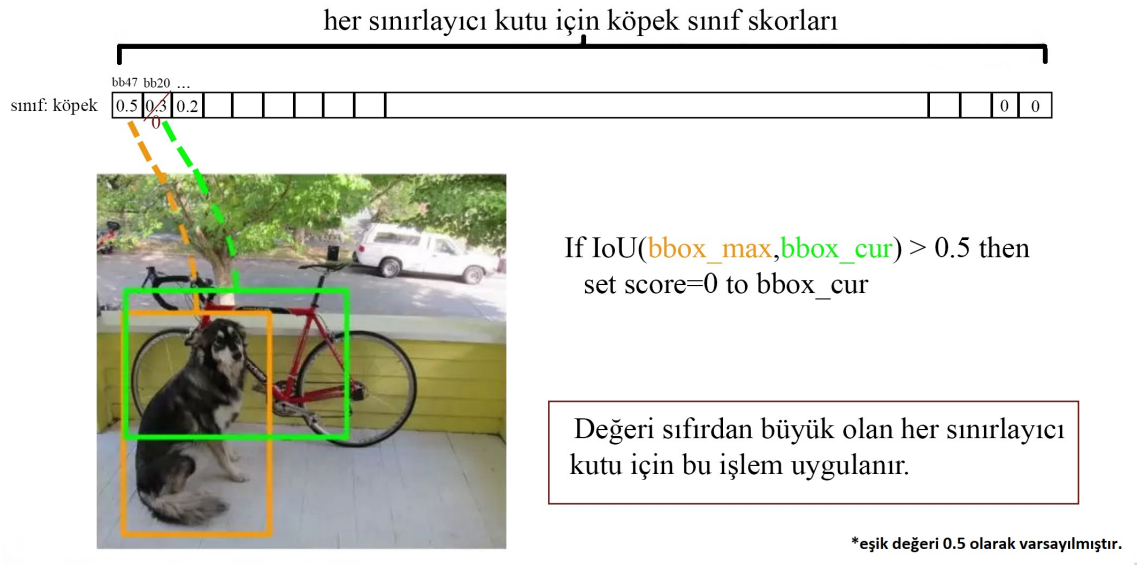
açısından geliştirilen YOLO4'de [39], YOLOv3'ten farklı olarak farklı bir omurga ağı kullanılmıştır. YOLOv4'de CSPDarknet53 omurga ağı kullanılmıştır.



Şekil 2.11 YoloV3 ağının çıkışındaki hücre bilgisi [40]

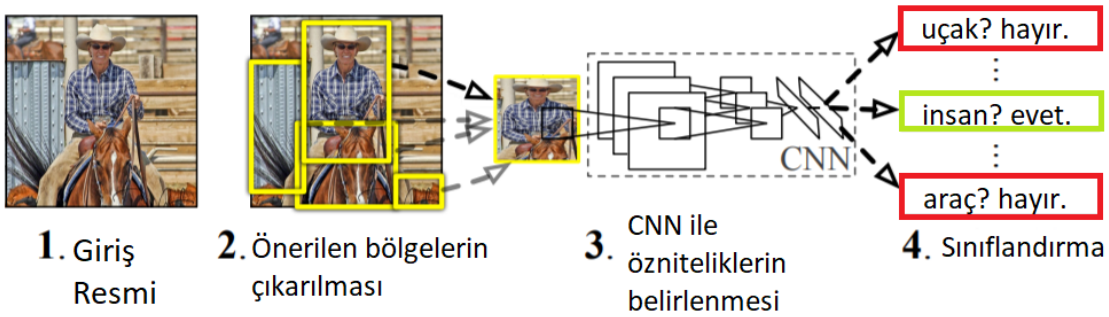
2.2.2 Bölge Tabanlı Evrişimsel Sinir Ağı

Bölge Tabanlı Evrişimsel Sinir Ağı (Region Based Convolutional Neural Networks, R-CNN) [42] olarak adlandırılan model, CNN tabanlı bir nesne tespit modelidir. R-CNN modelinde (Şekil 2.13) çok sayıda bölge seçmek yerine önerilen bazı görüntü parçaları belirlenir ve bunların nesneye karşılık gelip gelmediği kontrol edilir. Bunu belirleyebilmek için seçici arama algoritması [43] kullanılmaktadır. Elde edilen her bölge için konvolüsyon sinir ağı kullanılarak öznitelikler çıkarılmaktadır. Sınıfları belirlemek için destek vektör makine algoritması (Support Vector Machine, SVM), sınırlayıcı kutular için ise regresyon modeli kullanılmaktadır. Ancak bu mimarinin



Şekil 2.12 NMS işlemi [41]

dezavantajı büyük boyuttaki veri kümeleri için çok fazla zaman alması ve maliyetli olmasıdır.

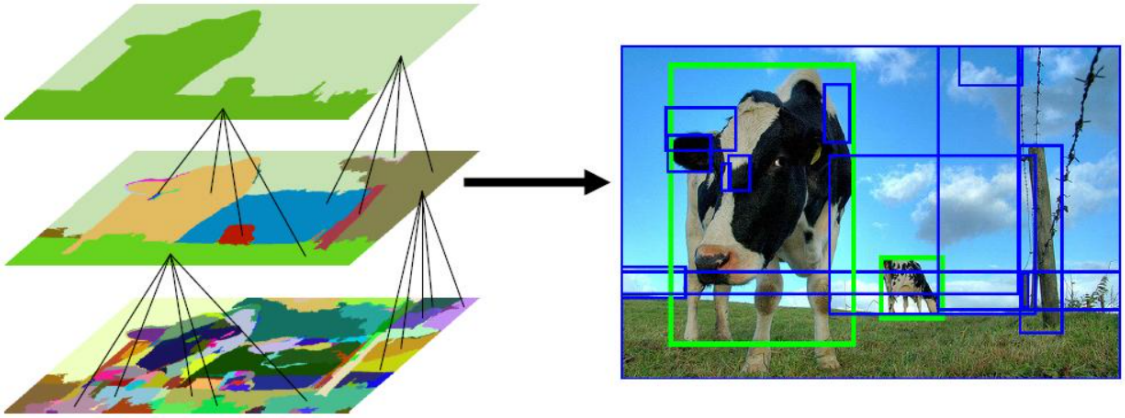


Şekil 2.13 R-CNN mimarisi [42]

Nesnenin yerinin belirlenmesi nesne tespiti için önenmli bir adımdır. Bunu yapabilmek için kullanılan yaklaşımlardan biri kayan pencere metodudur. Bu metod tam arama (exhaustive search) olarak adlandırılmaktadır. Bu yöntemde küçük boyutlu resimler için bile binlerce pencerede nesne araması yapılması nedeniyle maliyetli bir yöntemdir. Buna yönelik bölge öneri için tam arama ve segmentasyonunu birlikte kullanan seçici arama algoritması önerilmiştir [43].

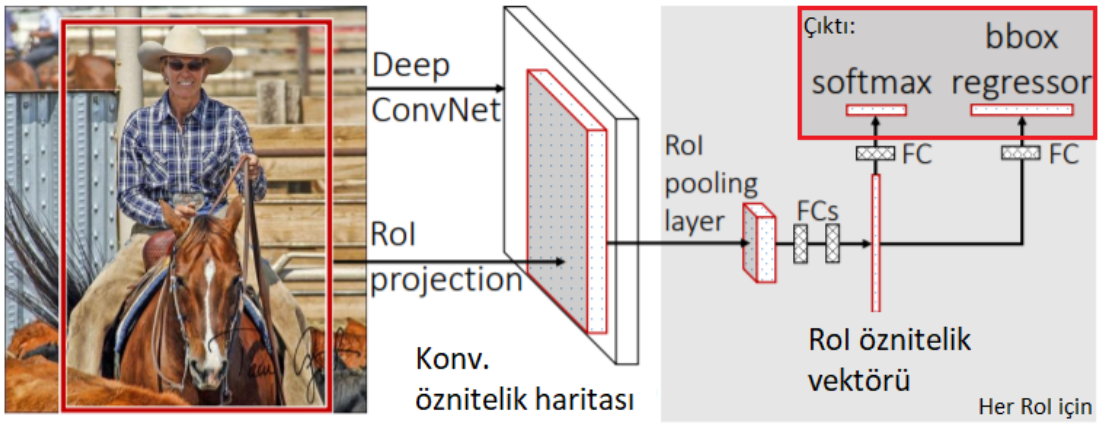
Seçici arama algoritması, bölgelerin benzerliğini renk, boyut, şekil, doku bilgilerini hesaplayarak karar vermektedir. Küçükten büyüğe doğru bir hiyerarşi içermektedir. Önce küçük bölgeler belirlenir, sonrasında benzer olan iki bölge birleştirilir ve işlem tekrarlı olarak devam eder. Sonucunda nesne içerebilecek olan bölgeler belirlenmektedir (Şekil 2.14).

R-CNN mimarisinde nesne olmaya aday bölgeler seçici arama ile belirlendikten sonra



Şekil 2.14 Seçici arama yöntemi [43]

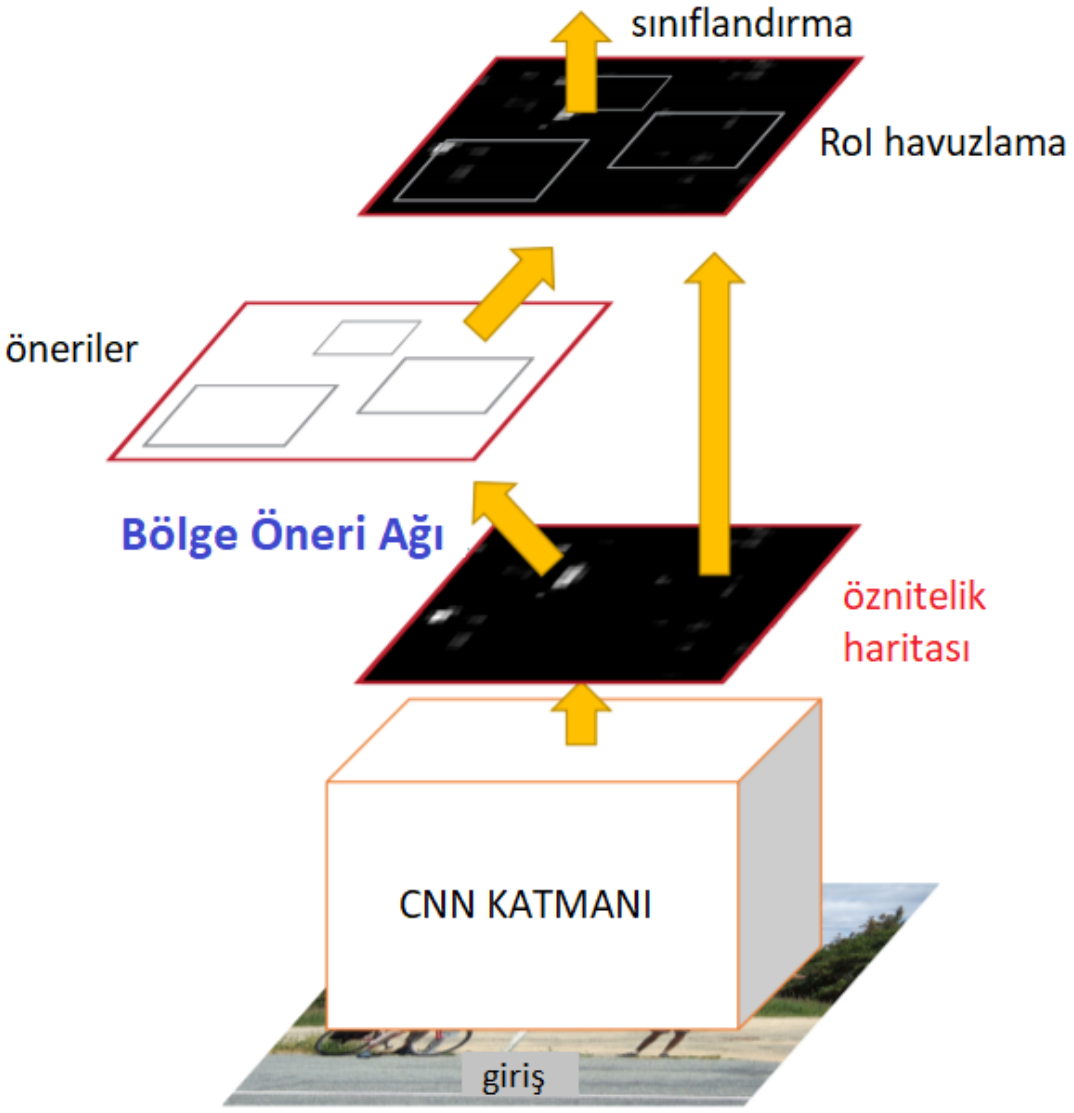
her bir bölge ayrı ayrı CNN modeline verilir ve bu durum işlem karmaşıklığına sebep olmaktadır. İşlem karmaşıklığına yönelik Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Fast R-CNN) önerilmiştir [44]. Burada R-CNN den farklı olarak görüntü ilk olarak konvolüsyon katmanına aktarılır ve daha sonra önerilen bölgeler belirlenmektedir. Bu sayede tek bir özellik haritası oluşturulacağı için bellekte fazla yer kaplamamaktadır. Görüntüleri yapay sinir ağına aynı boyutlarda girdi olarak verebilmek için ilgili bölgeler (Region of Interest, ROI) havuzlama (pooling) katmanından geçirilmektedir. Daha hızlı ve doğru tahminler yaptığı için SVM yerine softmax katmanı kullanılmaktadır (Şekil 2.15).



Şekil 2.15 Fast R-CNN mimarisi [44]

Daha sonra Daha Hızlı Bölge Tabanlı Evrişimsel Sinir Ağı (Faster R-CNN) önerilmiştir [9]. Faster R-CNN modeli (Şekil 2.16) iki aşamadan oluşmaktadır. İlk aşama, aday bölgelerin oluşturulmasıdır. Aday bölgeler oluşturulurken seçiçi arama algoritması yerine bölge öneri ağı (Region Proposal Network, RPN) algoritması kullanılmaktadır. Bu algoritma ile 3 × 3'lük kayan pencere, konvolüsyon katmanında oluşturulan öznitelik haritası üzerinde gezerek k tane çapa kutu üretmektedir. RPN, her çapa

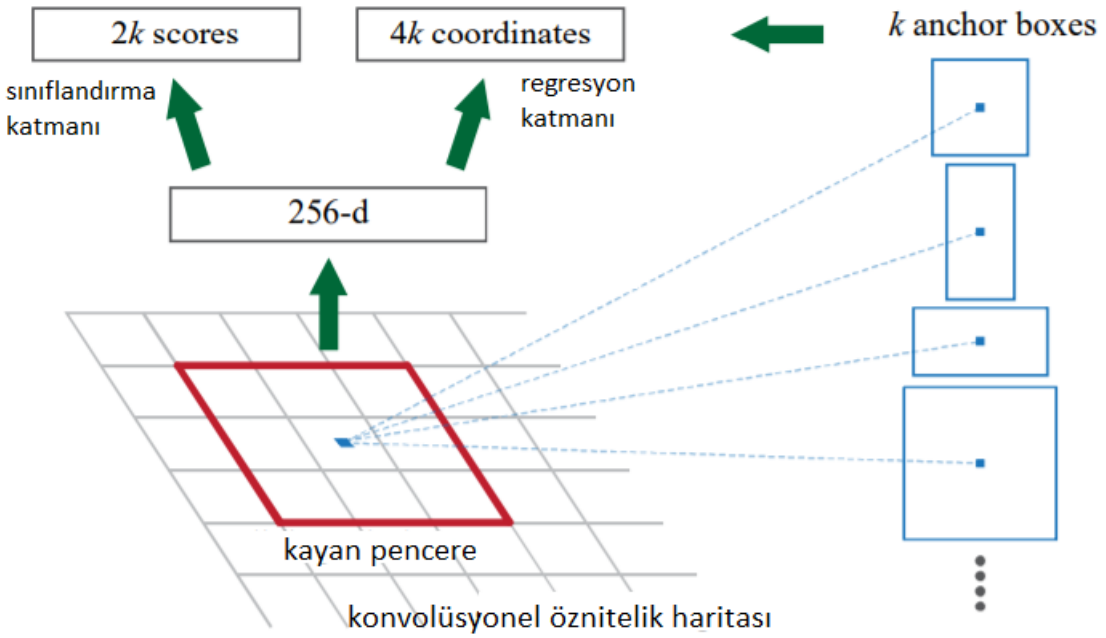
kutu için nesne olma ihtimali (objectness) ve nesne bulunması beklenen sınırlayıcı kutu tahmin etmektedir (Şekil 2.17). İkinci aşama, bölge öneri ağı çıktısı kullanılarak, regresyon ile sınırlayıcı kutu ve sınıflandırıcı ile nesnenin sınıfı tahmin edilmektedir.



Şekil 2.16 Faster R-CNN mimarisi [9]

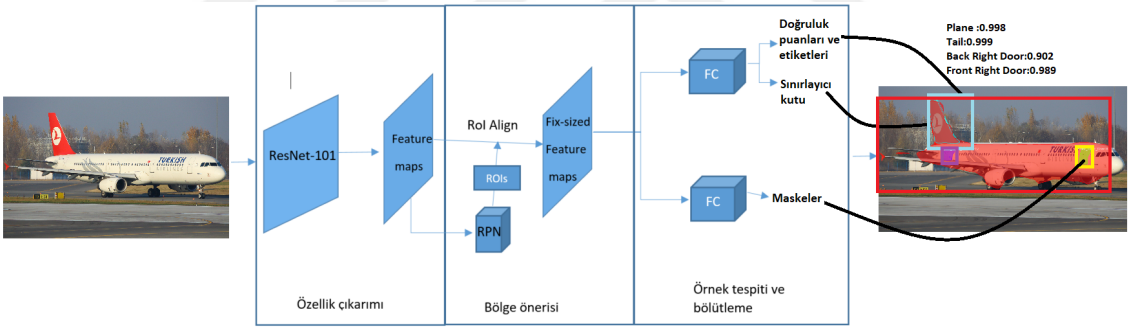
2.2.2.1 Mask R-CNN mimarisi

Mask R-CNN [4], Facebook Yapay Zeka Araştırması (FAIR) ekibi tarafından geliştirilen örnek nesne bölütlemesi (instance object segmentation) algoritmasıdır [4]. Temel olarak Faster R-CNN [9] algoritması üzerine geliştirilmiştir. Mask R-CNN çıktısında sınırlayıcı kutu tespitine paralel olarak nesnelerin maskelelerini de tahmin etmektedir. Böylece nesne bölütlemesi için esnek ve genel bir çerçeve sunulmuştur. Önerilen yöntemle tespit edilen nesnelerin her biri için yüksek kaliteli bölütleme maskesi oluşturulur.



Şekil 2.17 RPN mimarisi [9]

Mask R-CNN genel mimarisi şekil 2.18'de gösterilmiştir.



Şekil 2.18 Mask R-CNN genel mimarisi [45]

Öncelikle giriş görüntüsünün CNN ile öznitelik haritası çıkarımı yapılır. Bu öznitelik haritası bölge öneri ağına (Region Proposal Networks, RPN) giriş olarak verilir. RPN, bu öznitelik haritası üzerinde kayan pencere metoduyla nesne olabilecek bölgeleri tespit eder. Bu sayede nesne olabilecek ankorlar oluşturulur. Bu ankorların birbiriyle ne kadar örtüştüğü kesişimin birleşme oranı (Intersection over Union, IoU) ile gösterilir. Bu değerin en yüksek değerine sahip olanı seçilir. En yüksek birleşme oranı seçilerek, sonucunda kesişim bölgeleri (ROI) elde edilir.

RPN sonucunda farklı boyutlarda ROI 'ler üretilir. Faster R-CNN ve öncesinde farklı boyutlardaki kutuları sabit giriş boyutuna getirme işlemi ROI Pooling işlemi ile yapılmaktadır. Mask R-CNN ile ROI Align adlı, bir özellik haritasını farklı noktalarda örnekleyen ve bilineer enterpolasyon uygulayan bir yöntem önerilmektedir [46].

Aynı boyuta getirilen öznitelik haritaları veya ROI ler sınıflandırma ve regresyon için ilerletilir. Bu adım paralel olarak iki aşamayı çalıştırır. Sonuç olarak maske, tam bağlantılı katmandan gelen sınıf etiketi, puanı ve sınırlayıcı kutu elde edilir. Sınıflandırma, regresyon ve bölütleme için kayıp fonksiyonu hesaplanır. Sınıflandırma sonucu her bir kategori için sınıf puanı ve regresyon ile sınırlayıcı kutu üretilir.

Bu çalışmada bölge tabanlı nesne tespiti algoritmalarının en gelişmişi olan Mask R-CNN mimarisi kullanılmıştır.

2.3 Nesne Takibi

Bu çalışmada nesne takibi için derin öğrenme tabanlı bir sistem geliştirilmiştir. Nesne takibi temel anlamda hedef nesne veya nesnelerin konumunun önceki/sonraki bilgiler kullanılarak belirlenmesidir. Nesne tespitinden farklı olarak, nesne takip etme her çerçevede bir nesnenin bilgisi çerçeveden çerçeveye kaybedilmeden kullanılmaktadır. Tek nesne takibi (Single Object Tracking, SOT) ve çoklu nesne takibi (Multi Object Tracking, MOT) yapan takip yöntemleri bulunmaktadır. Nesne takibi güvenlik ve gözetim, trafik kontrolü [47], insan etkileşimleri, otonom araç sektörü, yaya sayımlarında [48], video sıkıştırma işlemleri [49] gibi birçok alanda kullanılabilmektedir.

Görsel nesne izlemenin amacı, bir videonun ilk karesindeki herhangi bir hedef nesnenin konum bilgisini kullanarak, sonraki tüm karelerde en iyi doğrulukla konumunu tahmin etmektir [50].

Birden fazla nesne takip etme konusunda literatürde çok çeşitli yaklaşımlar olmasına rağmen genel olarak şu adımları içerir [14]:

- Tespit aşaması : Bir veya birden fazla nesnenin yeri tespit edilir. Özellik çıkarımı/Hareket tahmini : Bir veya daha fazla özellik çıkarma algoritması görünüm, hareket özelliklerini çıkarır. İsteğe bağlı olarak, izlenen her hedefin bir sonraki konumları tahmin edilir.
- Benzerlik (Affinity) aşaması: Özellikler ve hareket tahminleri üzerinde benzerlik/uzaklık puanları hesaplanır.
- İlişkilendirme : Benzerlik/Uzaklık ölçüleri kullanılarak nesneler arasında ilişki kurulur.
- Analiz : Nesnelerin davranışları incelenir.

Nesne tespit etme işlemi nesne takip etme yöntemlerinin alt parçası olarak düşünülebilir. Tespit etme işlemlerinde nesnelerin konumları belirlenirken, takip etme işlemlerinde bir dizi çerçeve üzerinde ilişkilendirilerek işlemler yapılır. Takip işleminde önceki çerçevede tespit edilen "A" kimlikli nesne, çerçevede kaldığı süre boyunca aynı "A" kimliği ile devam edebilmelidir.

Takip yöntemlerinde zaman efektifliği, kimlik değiştirilmesi gibi zorluklar bulunmaktadır. Tespit etme işleminde olan eksiklikler takip etme sistemini de etkileyerek yörünge izleme gibi durumlarda kesinti durumları oluşturur. Bu sebeple tespit etmenin başarısı takip etmenin başarısını da etkilemektedir. Bunun yanında nesnelerin birbirinin önüne geçmesi durumda (occlusion) da takip etme algoritmaları için bir zorluktur.

Birçok uygulamada izlemenin çevrimiçi olması önemli bir kriterdir. Diğer bir deyişle, çevrimiçi takip yöntemleri nesne konumunu tahmin ederken gelecekteki çerçeveleri kullanmazlar. Gelecekteki çerçeveleri kullanan offline takip yöntemleri, genellikle tıbbi tabanlı uygulamalarda kullanılabilmektedirler [51].

Bu tez çalışmasında nesne takibi çevrimiçi olarak, farklı sahnelerdeki tek kamera görüntüleri kullanılarak, birden fazla nesne için yapılmaktadır. Nesne takibi için derin öğrenme yöntemlerinden SiamMask kullanılmıştır.

2.3.1 SiamMask Mimarisi

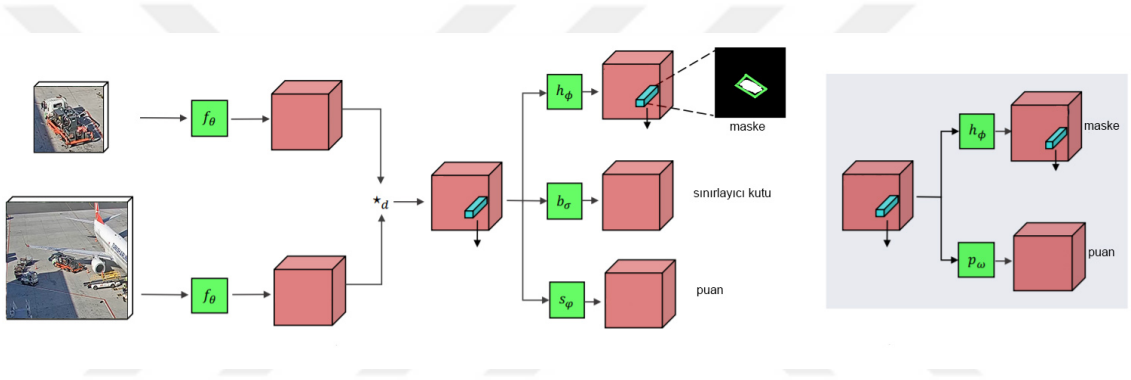
SiamMask [52] olarak adlandırılan bu yöntem nesne takibi için popüler tam evrişimli siyam [53] yaklaşımını temel almaktadır. SiamMask, başlangıçta sınırlayıcı kutu konumlarını alır ve gerçek zamanlı olarak çalışmaktadır. Saniyede 55 kare hızında nesne bölütleme maskeleri ve döndürülmüş sınırlayıcı kutular üretmektedir. Çok yönlülük ve hız sebebiyle VOT-2018 [54] gerçek zamanlı takip etme yöntemleri arasında gelişmiş bir yöntem sunmaktadır.

SiamMask yöntemi nesne takibi ve video nesne bölütleme yapabilme kapasitesine sahiptir. Milyonlarca çift video karesi üzerinde çevrimdışı olarak eğitilmiş tamamen evrişimsel (fully convolutional) siyam ağlarına [53] dayalı olarak geliştirilen hızlı izleme yaklaşımlarının başarısı dikkate alınarak tasarlanmıştır. SiamMask ağı 3 görev için eş zamanlı olarak eğitilmiştir. Birincisi tamamen evrişimsel yaklaşımı kullanan bu çalışmadaki gibi [53], hedef nesne ve birden fazla aday üzerinde benzerlik ölçüsünü öğrenmektir. Çıktısında nesnenin konumunu gösteren yoğun cevap haritası elde edilmektedir. Yoğun cevap haritası elde etmek için örnek bir z görüntüsünü daha büyük bir arama görüntüsü olan x ile karşılaştıran, çevrimdışı eğitilmiş tam

evrişimli bir siyam ağının kullanılmasını önermektedir. z, wh boyutunda hedef nesne ortalanacak şekilde kırpılmış halini x ise hedefin son tahmin edilen pozisyonu ortalanacak şekilde daha büyük boyutta kırpılmış halini ifade etmektedir. İki girdi, aynı CNN tarafından işlenerek çapraz ilişkili özellik haritası elde edilmektedir (Eşitlik 2.13).

$$g(z,x) = f(z) * f(x). \tag{2.13}$$

İki kollu ve üç kollu varyasyonlu mimarileri vardır. Üç kollu varyasyonluda maske, sınırlayıcı kutu ve skor elde edilir. İki kollu varyasyonlu da ise maske, ve skor bilgisi çıktısı elde edilir. Burada sınırlayıcı kutu maske üzerinden elde edilir. Mimarisi şekil 2.19'de gösterilmiştir.



Şekil 2.19 SiamMask Mimarisi [52]

3
SİSTEM TASARIMI

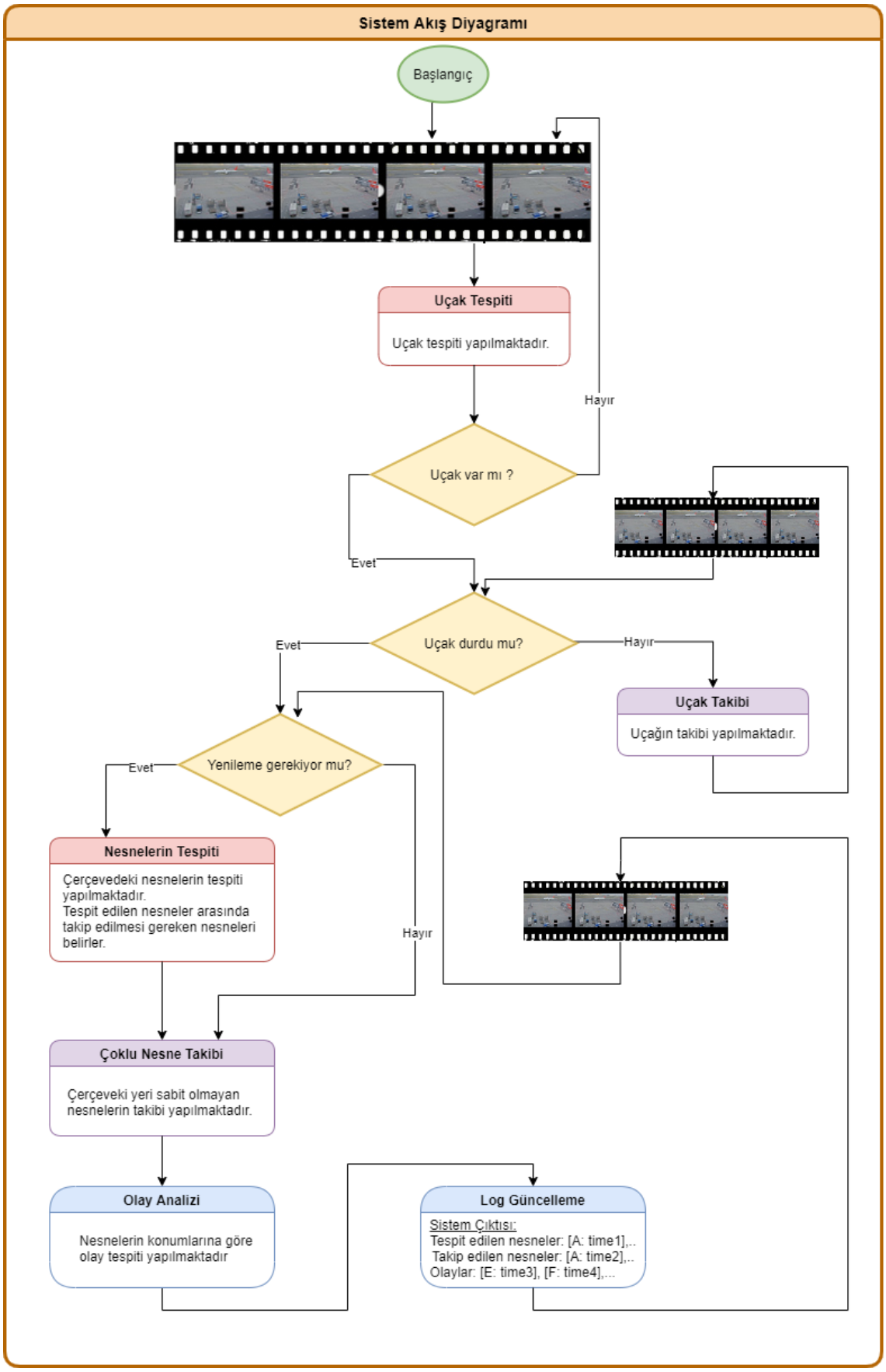
Bu çalışmada geliştirdiğimiz sistemde, uçak görüntüde tespit edildiği zaman, yani aprona geldiğinde operasyon işlemlerinin izlenmesine başlanmaktadır. Önce uçak iniş yapıp durana kadar takip edilir. Operasyonlar uçak ile etkileşimli olarak yapılmaktadır. Bu sebeple operasyon ile ilgili nesnelerin uçak ile olan IoU değerlere bakılmaktadır. Nesnelerin konum bilgisi, hareket yönleri bilgilerinden yola çıkarak hareket bilgisi elde edilmektedir. Sistem, bu bilgilerden yola çıkarak operasyonlarla ilgili çıkarımlarda bulunup sistem çıktısı üretmektedir. Ayrıca bir olay ardından beklenen bir olay varsa ve belirlenen süre içerisinde gerçekleşmediyse sistem bunu anomali durum olarak değerlendirerek uyarı mesajı da vermektedir. Örneğin; İkram aracı gelmişse ve belli bir süre yükleme başlamadıysa ikramın başlamadığına dair uyarı çıktısı verilmektedir.

Yapılan işlemler nesne tespiti, nesne takibi ve analiz bölümleridir. Bu işlemler nesneler için ortak yapılan temel işlemlerdir. Bu sebeple sistem modüler bir yapıda tasarlanmıştır. Sistemin akış diyagramı şekil 3.1'de gösterilmektedir. Modüler yapı sayesinde yeni bir nesneyi sisteme entegre etmek zor olmayacaktır.

Bu bölümde tez çalışması için genel sistem tasarımı, veri kümesi ve kullanılan kütüphaneler anlatılacaktır.

3.1 Veri Kümesi

Veri kümesi, Türk Hava Yolları Teknoloji AR-GE merkezinden temin edilmiştir. 24 saatlik videolar üzerinde operasyonların olduğu bölümler seçilmiştir. Operasyonları tamamen görebilecek olan kamera görüntüleri üzerinde çalışılmıştır. Açısı uçağı bütünüyle gören görüntüler üzerinden analiz yapılacağı için açısı uymayan görüntüler elenmiştir. Videoların bazılarında park pozisyonuna hiç uçağın gelmediği veya uçak üzerinde hiç operasyon yapılmadığı durumlar gözlemlenmiştir. Bu görüntüler atıl oldukları için değerlendirilmmemiştir. Eğitim veri kümesinden alınan örnek resimler şekil 3.2 ve şekil 3.3' de gösterilmiştir. Park pozisyon numaralarının görünmemesi için maskelenmiştir.



Şekil 3.1 Sistem Akış Diyagramı

Tez çalışmasında, video görüntülerde bulunan körük/merdiven yanaşması, ikram yüklemesi, yakıt yüklemesi ve pushback operasyonları değerlendirilmiştir.



Şekil 3.2 Gündüz ortamda alınan örnek görüntü

Nesne tespiti için;

Sisteme uçak ve uçak bölümlerini öğretebilmek için öncelikle apron görüntülerinden alınan resimlerle bir veri kümesi oluşturulmuştur. Tespit edilmek üzere işaretlenen bölümler uçağın kuyruk, kapılar ve ön tekerlek bölümleridir. Bunlar dışında operasyon sürecini takip edebilmek için körük/merdiven, ikram, pushback ve yakıt araçları da etiketlenmiştir. Şekil 3.4'de verilen görüntüde, sınıflar gösterilmektedir. Aynı görüntüde görülebilmesi için pushback aracı el ile eklenmiştir. Örnek sayısını artırmak için PASCAL VOC [55] veri kümesinde bulunan uçak resimleri de veri kümesine dahil edilmiştir. Bu verilerin bir kısmı eğitim bir kısmı validasyon için ayrılmıştır. Etiketleme için VGG Image Annotator [56] kullanılmıştır. Eğitimde kullanılan sınıf ve örnek sayıları tablo 3.1'de gösterilmiştir. Veri kümesindeki bu resimlerin ortam, aydınlatma şartları, görüş açısı ve görüş miktarı gibi farklılıkları sistemin başarısını etkilemektedir. Veri kümesinde sol taraf örnek sayısının sağ taraftan az olmasının sebebi kameranın açısından kaynaklanmaktadır. Genelllikle uçağın sağ açısından gören kameralar olduğu için bu durum oluşmuştur. Ayrıca ön kapıların arka kapılardan fazla olma sebebi ise kameraların genellikle uçagın burun kısmında olmasından kaynaklanmaktadır.

Nesne takibi ve olay analizi için;

Bütün operasyonun tamamlanması yaklaşık olarak 2 saati bulabilemektedir.



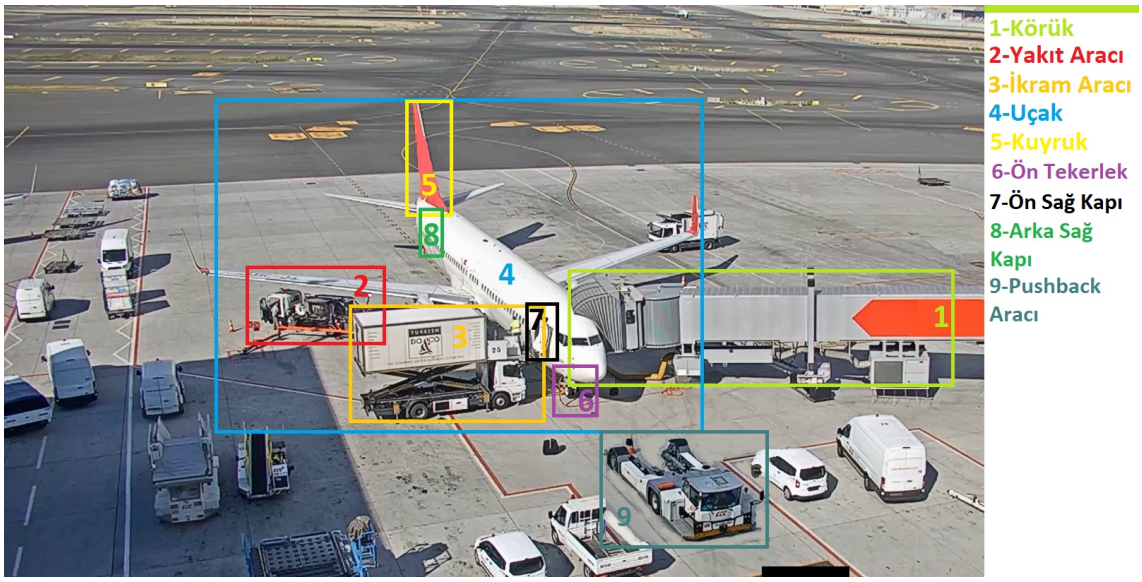
Şekil 3.3 Karanlık ortamda alınan örnek görüntü

Dolayısıyla maliyet ve donanım kısıtları nedeniyle öncelikli olarak olaylar parçalara ayrılmıştır. Bunun için analizi yapılan olaylar çerçevesinde videolar parçalara ayrılmıştır. Toplamda gece, gündüz ve farklı apron alanlarındaki videolar kullanılarak toplamda 19568 çerçeve üzerinden test işlemi gerçekleştirilmiştir.

3.2 Sistem Mimarisi

Sistem, uçağın apron alanına yanaşması ile başlamakta, pushback aracıyla geri itme işleminin başlamasıyla işlem sonlanmaktadır. Nesne tespit etme işlemleri nesne takip etme işlemlerine göre daha maliyetli olduğundan her çerçevede tespit yapmak yerine belirli aralıklarla tespit yapılmaktadır. Uçağın durmasının ardından uçak bölümlerinin takip edilmesine gerek yoktur, bu sebeple takip algoritmasına hiç verilmezler. Yer operasyonu için belirlenen araçlar tespit edildiği zaman takip algoritmasına girerler.

Sistem üç ana bölümden oluşmaktadır. İlki görüntülerde nesnelerin tespitini yapan bölümdür. Eğer çerçevede uçak tespit edilmemişse herhangi bir işlem yapmadan uçak tespit edilene kadar tespit etme algoritması çalışır. Operasyonların başlayabilmesi için uçağın park pozisyonunda olması gerekmektedir. Bu sebeple uçak durana kadar takip edilir. Ayrıca belirli aralıklarla tespit etme işlemi çalıştırılır ve çerçevedeki nesneleri tespit eder. Ana bölümlerden bir diğeri nesne takip modülüdür. Son bölüm ise tespit etme ve takip etme sonuçlarını değerlendiren bölümdür. Bu bölüm sistemin davranışını raporlamaktadır. Gerçekleştirilen sistemin genel mimarisi şekil 3.5 gösterilmiştir. Şekilde gösterilen ikram aracı olan doco aracı ve körük ile ilgili olay



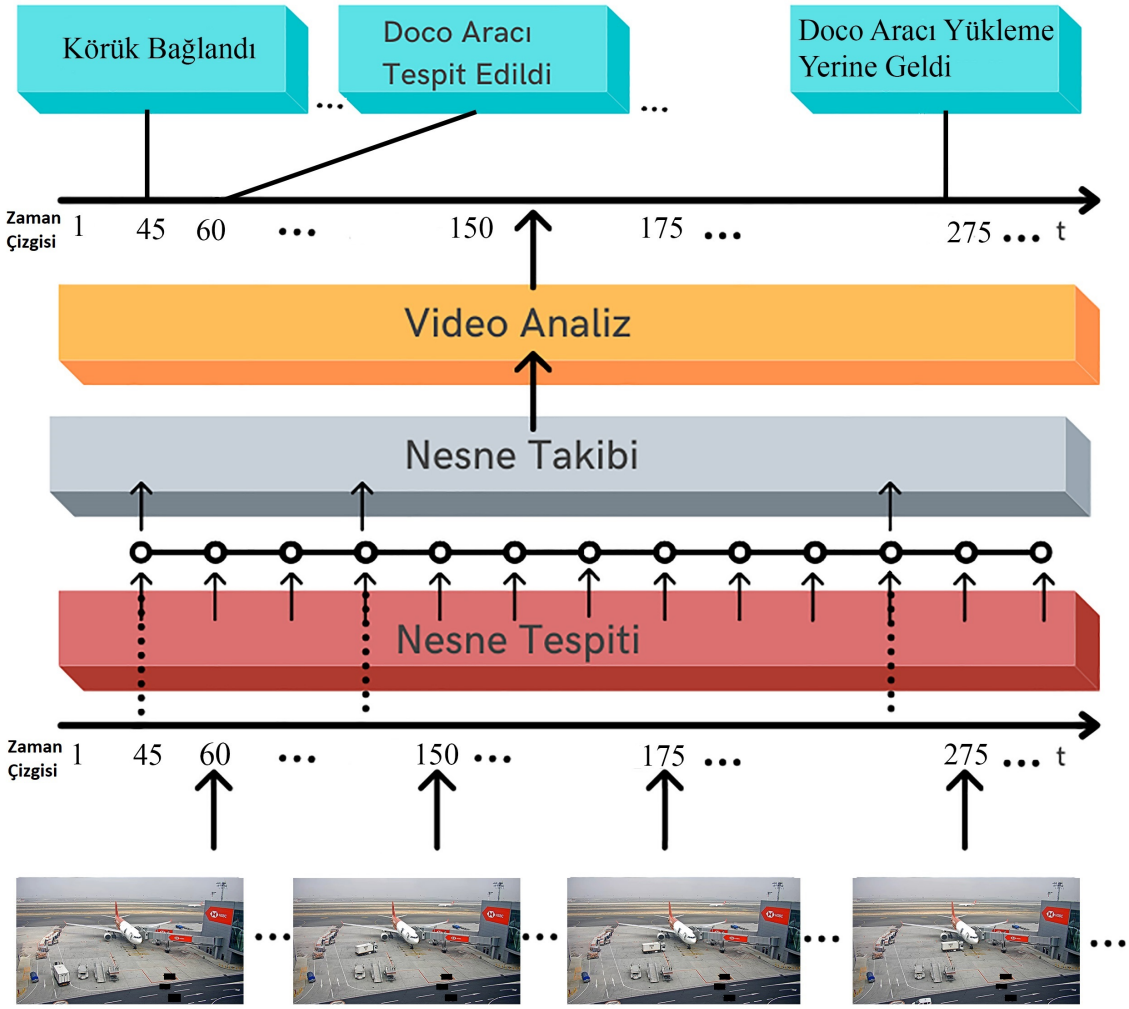
Şekil 3.4 Oluşturulan sınıf örneği

Tablo 3.1 Sınıflar ve örnek sayıları

Sınıf İsmi	Örnek Sayısı
Uçak	2386
Ön sağ kapı	1779
Ön sol kapı	400
Arka sol kapı	300
Arka sağ kapı	1419
Kuyruk	2239
İkram Aracı	792
Pushback aracı	386
Ön Tekerlek	1869
Yakıt aracı	770
Merdiven	663
Körük	2202

tespiti sembolik olarak koyulmuştur.

Derin öğrenme tespit etme ve takip etme yöntemlerine dahil edilmiştir. Tespit sonrası oluşan çıktı takip için kullanılacak modele verilir. Takip sonucuna göre nesnenin konumu, hareketli ise hareket yönü ve şiddeti gibi bilgiler kullanılarak nesnelerin durumu hakkında bilgi edinilir. Nesnelerin durumuna göre de bazı çıkarımlarda bulunularak olaylar hakkında yorumlanmış log çıktısı üretilir. Nesne takibi zorluklarından olan takip edilen bir nesnenin sonraki çerçevelerde tespit edilememesinden kaynaklı takip etme de kesintiler oluşabilmektedir. Bu sebeple tespit etmenin başarısı takip etmenin başarısını etkilemekte ve sonuç olarak da analiz çıktısını etkilemektedir.

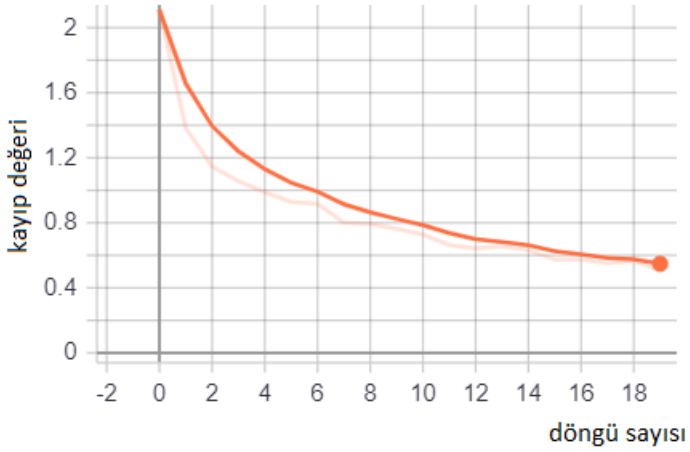


Şekil 3.5 Sistem Genel Mimarisi

Nesne tespiti için Mask R-CNN kullanılmıştır. Aprondaki videolar üzerinden görüntüler alınmış ve bu görüntüler kullanılarak uçağın ön sağ kapı, ön sol kapı, arka sağ kapı, arka sol kapı bileşenleri eklenerek model eğitilmiştir. Bu bileşenlerin konumları analiz kısmında kullanılarak olaylar ile ilgili çıkarımlarda bulunulmuştur. Tespit etme işleminin çıktısı SiamMask modeline verilerek takip etme işlemi yapılmıştır. Burada tek bir uçak operasyonu üzerinde çalışılmıştır. Bu sebeple çoklu nesne takibi yerine tek nesne takibinin çoklanması ile bir sistem geliştirilmiştir. Çoklu nesne takibinde nesnelerin ilişkilendirilmesinde bir çok yöntem önerilmiştir. Ancak sonuçların belli bir düzeyde başarılı olması ve basitliği sebebiyle merkez noktaları takip eden ve bu noktalar üzerinden nesneleri ilişkilendiren bir yöntem geliştirilmiş ve uygulanmıştır. Sonrasında kimlik kazanmış nesneler üzerinde analizler yapılmış, çıkarımlarda bulunulmuştur.

3.2.1 Eğitim Aşaması

Nesne tespiti için veri kümesi hazırlandıktan ve model gerçeklendikten sonra ön sağ, ön sol, arka sağ, arka sol, ön tekerlek, kuyruk, ikram aracı, yakıt aracı, geri itme aracı, körük/merdiven tespit edilebilmesi için eğitim başlatılmıştır. Eğitim için ayrılan veri kümesinin %20'si validasyon için kullanılmıştır. Toplam 20 döngü (epoch) kullanılarak eğitim sağlanmıştır. Kayıp değerleri şekil 3.6 gösterilmiştir. Her adımda kayıp değerinin düşmesi sistemin öğrenebildiğini göstermektedir.



Şekil 3.6 Nesne tespiti kayıp değerleri

3.2.2 Omurga Ağı

Nesne tespiti, zengin özellik çıkarmak için güçlü omurga ağına ihtiyaç duymaktadır. Omurga ağı, nesne tespitinde özellik çıkarmak için kullanılmaktadır. Görüntüleri girdi olarak alır ve karşılık gelen girdi görüntüsünün özellik haritalarını çıkarır. Buna bağlı olarak, omurga ağının performansı genel ağın performansını etkilemektedir.

Nesne tespitinde popüler olarak ResNets [36], AlexNet [57], VGG-16 [2], GoogLeNet [58], DarkNet-19 [38] gibi CNN mimarileri kullanılmaktadır. ResNet mimarisinin 18 katmanlı, 34 katmanlı, 50 katmanlı, 101 katmanlı, 152 katmanlı varyasyonları bulunmaktadır. Mimarileri şekil 3.7' gösterilmektedir.

Daha iyi doğruluk değerlerini elde edebilmek için sığ omurga yapıları kullanmak yerine daha derin omurga yapıları benimsenmiştir. Faster R-CNN'de daha zengin özellik belirleyebilmek için ResNet kullanılmıştır [4]. Nesne tespiti için farklı veri kümeleri üzerinde farklı omurga ağ mimarileri kullanan çalışmalar [59] incelendiğinde ResNet101 mimarisi ile başarılı sonuçlar alındığı görülmüştür. Bu tez çalışmasında nesne tespiti için omurga ağı olarak ResNet101 kullanılmıştır.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Şekil 3.7 ResNet mimarileri [36]

3.2.3 Nesnerin İlişkilendirilmesi

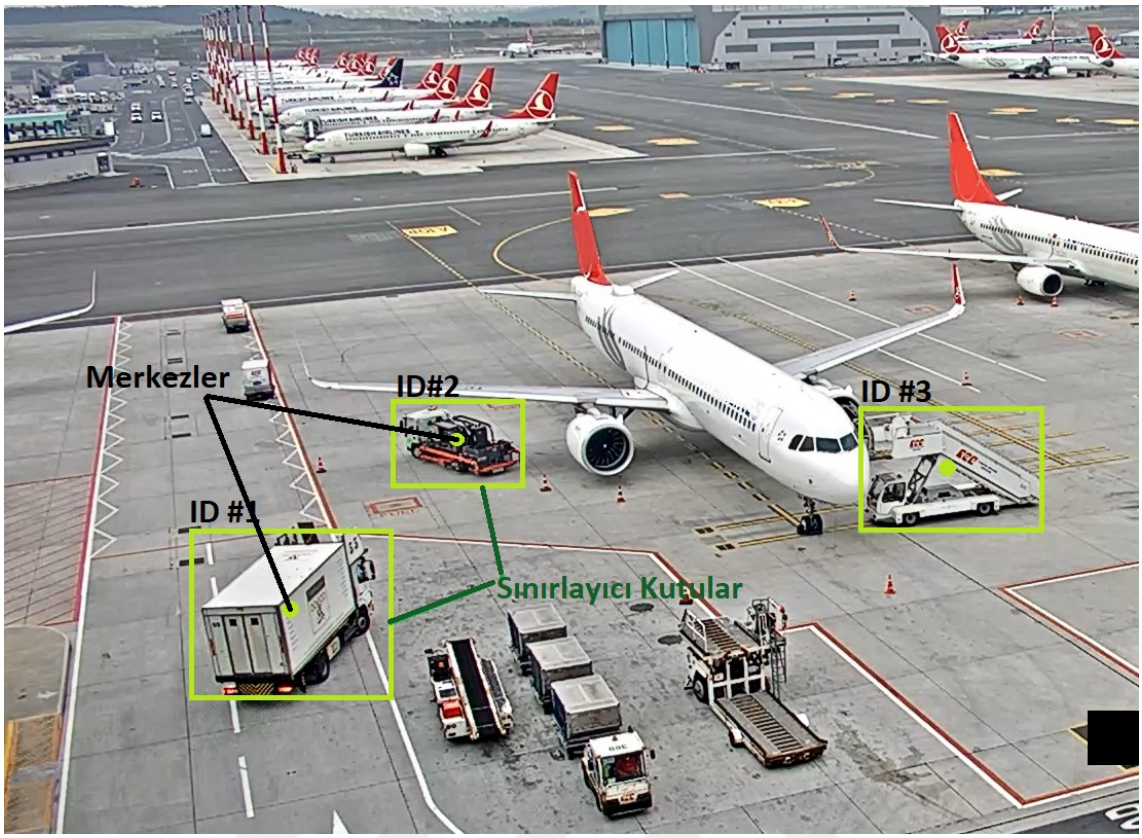
Tespit ve takip sonrası nesnelere kimlik atanması için nesnelerin merkez noktalarından yararlanılmıştır. Temel olarak merkez takibi ile birlikte yeni nesnelere kimlik ataması yapılmaktadır. Varolan nesneler, anlık çerçeveden gelen nesneler ile ilişkilendirilip merkez noktaları güncellenmektedir.

Başlangıçta nesneler tespit edilip, merkez noktaları belirlenmekte ve her tespit edilen nesne farklı bir "id" ile kimliklendirilmektedir (Şekil 3.8).

Nesneler kimliklendirildikten sonra güncelleme işlemi için öklid mesafe hesaplaması kullanılmaktadır (Şekil 3.9). Her bir nesne merkezlerinin birbiriyle uzaklık hesabının olduğu bir matris oluşturulmaktadır. Matris ile en yakın merkez noktaları belirlenmektedir. Böylece eski ve yeni noktalar arasında ilişki kurularak o kimliğin yeni merkez noktası güncellenmektedir (Şekil 3.10). Eğer algoritma ile takip edilen bir önceki bir nesne, yeni merkezlerin hiçbiri ile belli bir süre eşleştirilemezse bu nesnenin takibi sonlandırılmaktadır.

3.2.4 Operasyonların Tespiti

Operasyonda işleyiş olarak birbirine benzeyen olaylar bulunmaktadır. Ancak çalışmada farklı olayları incelemek adına ikram yüklemesi, yakıt yüklemesi, geri itme olayı gibi birbirinden farklı olaylar üzerinde çalışılmıştır. Operasyonlar uçağın belirli bölümlerinden yapılmaktadır. Uçak ile herhangi bir nesnenin etkileşimde olabilmesi için belirli bir yakınlıkta olması gerekmektedir. Bu sebeple olayların başlamasını belirleyebilmek için aracın uçak ile olan yakınlığı IoU metriği kullanılarak değerlendirilmektedir.



Şekil 3.8 Başlangıç aşaması

Sistemde tespit edilen nesnelerin bir kısmı takip edilebilir nesne, bir kısmı da sabit nesneler olarak belirlenmiştir. Takip etme algoritması ile elde edilen sınırlayıcı kutu koordinatlarındaki değişim incelenerek uçağın durup durmadığı belirlenmektedir. Uçak sabit bir konuma gelene kadar bir operasyon yapılamayacağı için sistem operasyonlarla ilgili herhangi bir çıktı üretmemektedir. Uçak durduktan sonra kapıların, kuyruğun ve tekerleğin yeri tespit edilerek sabit nesneler listesinde tutulmaktadır. Ancak ikram aracı, geri itme aracı ve yakıt aracı takip edilebilir nesne ketegorisi altında bulunmaktadır. Burada uçak nesnesi durana kadar takip edilir, durduktan sonra pushback olayına kadar takip edilmesine gerek yoktur.

İkram operasyonu için, ikram yükleme işlemi (Şekil 3.11) çoğunlukla sağ ön kapıdan yapılmaktadır. Bu sebeple ikram aracı sağ ön kapıya yanaşmaktadır. İkram yükleme işlemi kontrol edilirken aracın kapıya göre olan konumuna bakılmaktadır. Öncelikle ikram aracı çerçevede tespit edilince sistem tespit edildiğine dair bir log üretmektedir. Sonrasında ikram aracı sağ ön kapıya geldiğinde ikram aracının yükleme pozisyonuna geldiğine dair bilgi üretmektedir. Doco aracı yerine geldikten sonra optik akış ile hareket yönüne bakılarak, hareket yönü yukarı ise ikram yüklemenin başladığı, aşağı ise yüklemenin bittiği bilgisi üretilir. Son olarak yükleme pozisyonundan ayrılarak çerçeveden çıktığında ikram aracının gittiğine dair bilgi üretilmektedir.



Şekil 3.9 Nesnelerin merkez noktalarıyla öklid mesafesi hesaplaması

Yakıt operasyonu için, yakıt yükleme işlemi kanat altından yapılmaktadır (Şekil 3.12). Dolayısıyla yakıt aracının yakıt yükleme pozisyonuna gelip gelmediğini anlayabilmek için uçak ve yakıt aracının IoU parametresi hesaplanmaktadır. Bu bilgiyi elde ettikten sonra yakıt aracının durmasının tespit edilmesinden sonra optik akış ile yukarı yönlü hareket ile yakıt yüklemenin başladığı bilgisi elde edilmektedir. Benzer şekilde aşağı yönlü hareket ile yakıt işleminin bittiği tespiti sonrası çerçeveden ayrılmasıyla yakıt aracının gittiği bilgisi elde edilmektedir.

Pushback işlemi (Şekil 3.13) için, pushback aracının çerçevede tespit edilme bilgisinden sonra pushback aracı ile ön tekerlek IoU parametresi kontrol edilerek pushback aracının tekerleğe bağlanıp bağlanmadığı bilgisi ve aynı zamanda körüğün uçaktan ayrıldığı bilgisi de edinilmektedir. Pushback aracının hareket etmesiyle pushback işleminin başladığı bilgisi elde edilmektedir. Böylece operasyonlar sonlanmaktadır.

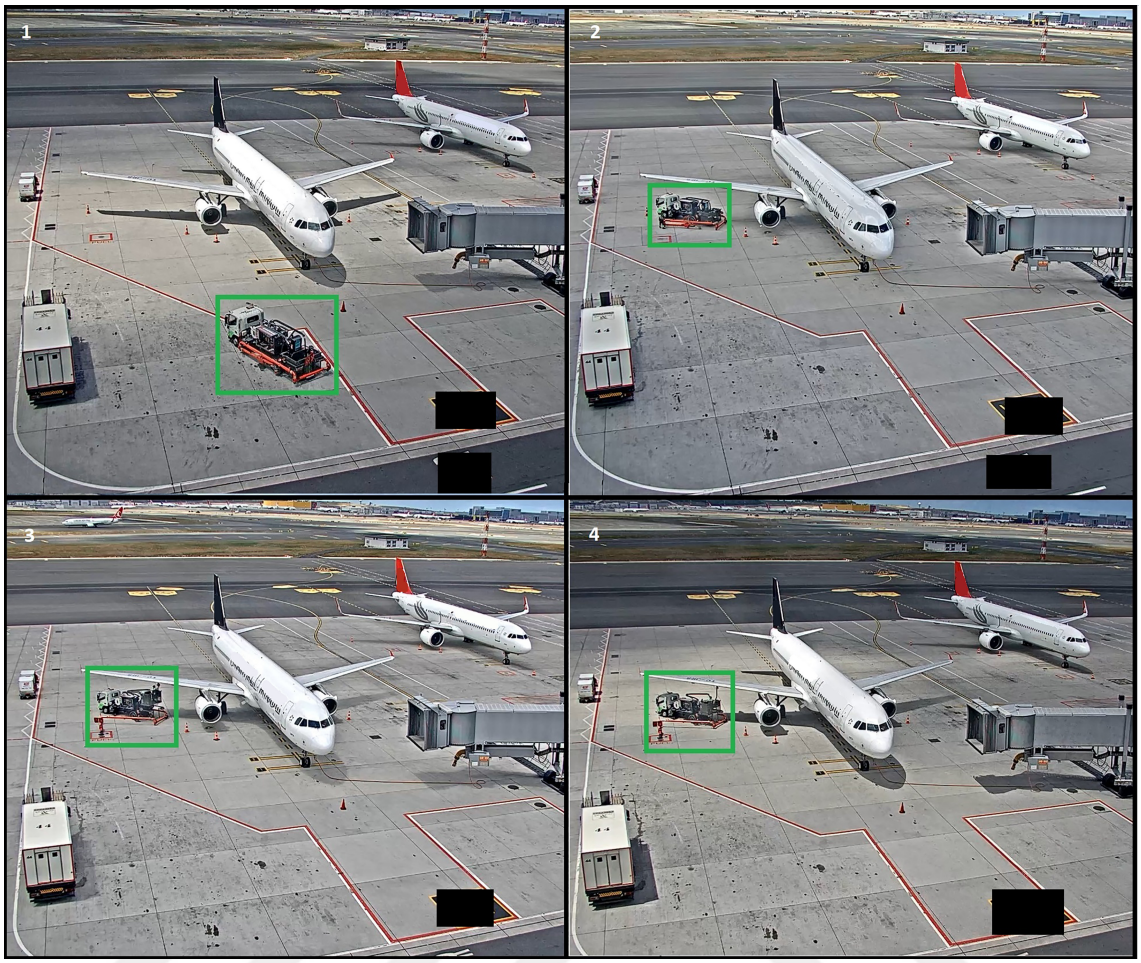
Genel olarak analiz yapılırken nesnenin durup durmadığı, hareketli ise optik akış ile hareket yönü, uçak bölümlerinin konum bilgisi, tespit edilen araçların konum bilgisi kullanılmaktadır.



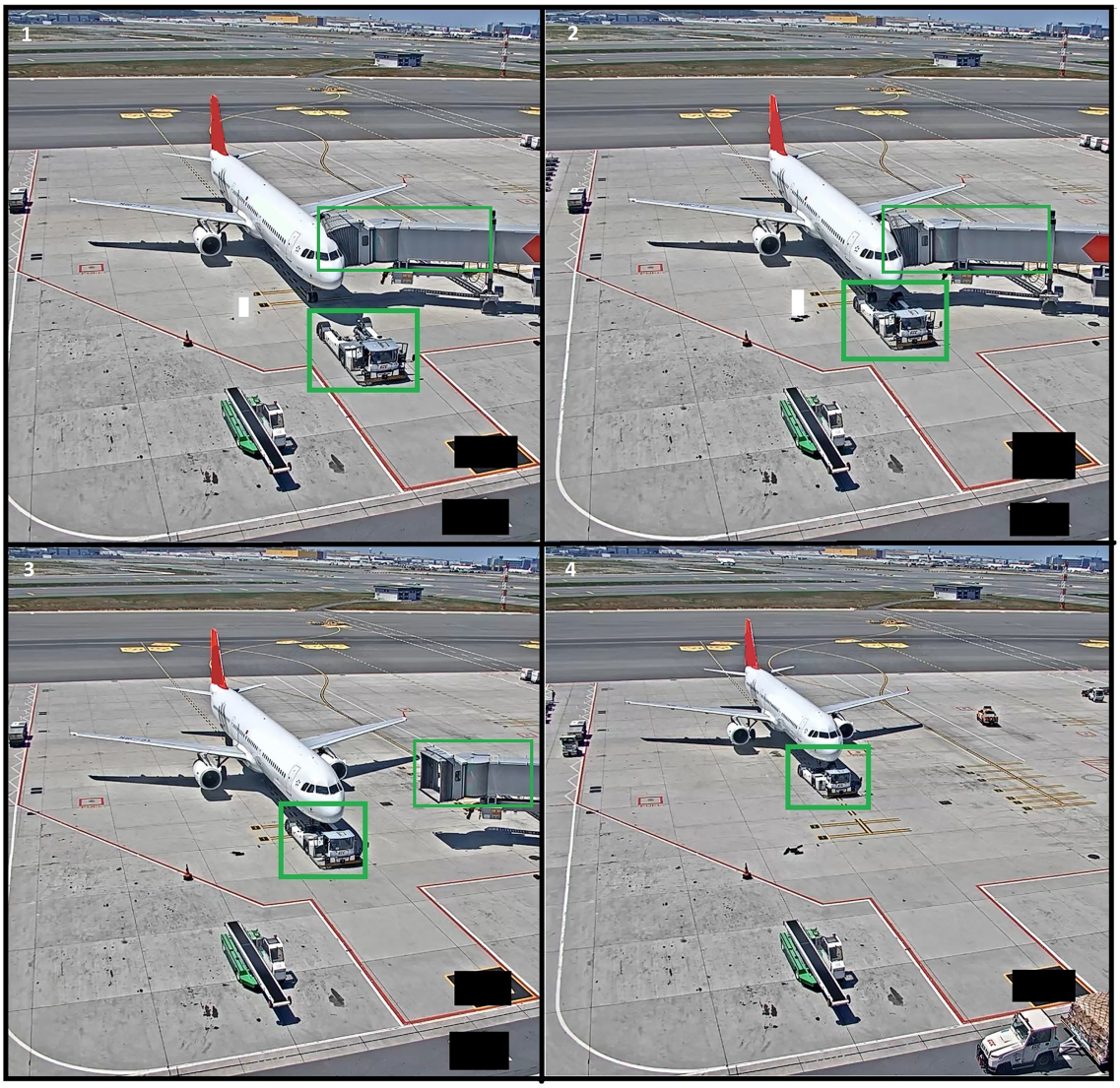
Şekil 3.10 Nesnelerin merkez noktalarının güncellenmesi



Şekil 3.11 İkram Yüklemesi Olayı



Şekil 3.12 Yakıt aracının gelmesi ve yakıt yükleme işlemi



Şekil 3.13 Pushback işlemi

4
DENEYSEL SONUÇLAR

Bu çalışmada, apronlarda uçakların park haline geçmesinden itibaren yapılan operasyonlarının tespitini ve takibini yapan, derin öğrenme tabanlı gerçek zamanlı bir sistem tasarlanmış ve gerçeklenmiştir. Sistem başarısı farklı senaryolar ile test edilerek değerlendirilmiştir.

4.1 Performans Ölçümü

Operasyonların değerlendirilebilmesi için önce operasyon aracı tespit edilmekte daha sonra da takip edilmektedir. Bu bölümde tespit ve takip algoritmasının başarısını ölçebilmek için kullanılan metriklerden bahsedilmektedir.

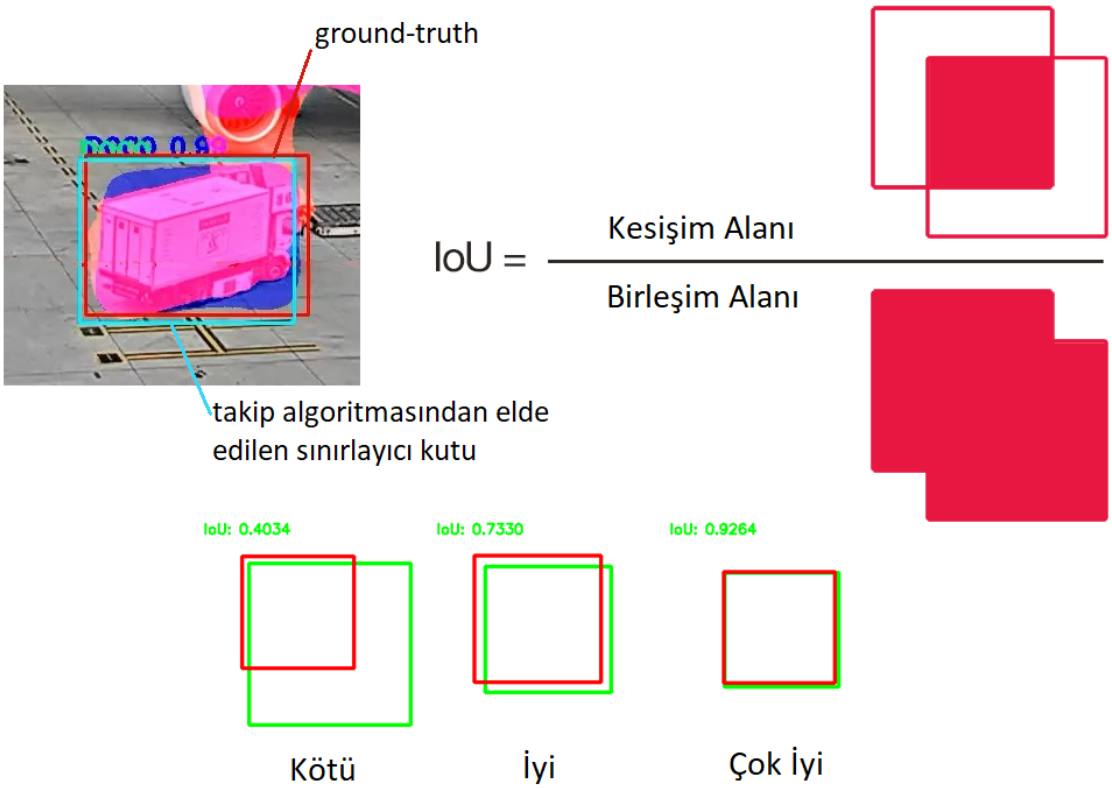
4.1.1 IoU

Nesnenin olması gereken değeri olarak adlandırılan gerçek referans(ground-truth) değeri ile tahmin edilen değerin benzerliği Jaccard indeksi ile ölçülmektedir. Bu sebeple IoU (Intersection over Union) hesabı, sınırlayıcı kutuların kesiştikleri alanın (intersection) bu iki kutunun bileşiminin (union) alanına bölümü olarak hesaplanmaktadır (Şekil 4.1).

4.1.2 Kesinlik (Precision) ve Duyarlılık (Recall)

Olasılık modelini kullanan tüm sistemler, sisteme ait örnekleri dört ayrı başlık altında toplamaktadır [60]. Bunlar doğru pozitif (true positive, TP), yanlış pozitif (false positive, FP), doğru negatif (true negative, TN) ve yanlış negatif (false negative, FN) değerleridir. TP, tahmin edilen pozitif örnek sayısını, FP ise yanlış tahmin edilen pozitif örnek sayısını ifade etmektedir.

Kesinlik, tahmin edilen örneklerdeki doğru pozitif sonuçların, pozitif olarak tahmin edilen tüm sonuçlara oranı ile hesaplanmaktadır. Kesinlik değerinin formülünü eşitlik



Şekil 4.1 IoU

4.1'de görülmektedir.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Tahmin edilen örneklerdeki doğru pozitif sonuçların, olması gereken tüm pozitif sonuçlara oranı duyarlılık değerini vermektedir. Duyarlılık değerinin formülünü eşitlik 4.2'de görülmektedir.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

4.1.3 Ortalama Kesinlik (Average Precision, AP) Hesaplaması

AP, her sınıf için hesaplanır. mAP'yi elde etmek için (Eşitlik 4.3) bu değerlerin ortalaması alınmaktadır.

$$mAP = \frac{1}{N} \sum_{i} AP_i \tag{4.3}$$

Çoklu nesne tespiti ve nesne takibinin başarıları ayrı olarak gösterilmektedir.

4.2 Nesne Tespit Başarısı

Nesne tespiti incelendiğinde nesnelerin farklı bir nesne olarak tahmin edilmesi gözlemlenmemiştir. Gerçekte nesne olduğu halde nesnenin tahmin edilememesi durumları görülmüştür. Bu sebeple karmaşıklık matrisi oluşturulmamıştır. Her sınıf için ayrı kesinlik değerleri hesaplanarak tablo 4.1'de gösterilmektedir. BG olarak gösterilen değer arka planı temsil etmektedir. Uçak genelde sağ taraftan görüldüğü için sol tarafın örnek sayısı azdır. Bu sebeple sağ taraftaki başarı oranı daha yüksek olarak görülmektedir. Ayrıca ön tekerleğin, pushback aracı tarafından kapatılması gibi nesnelerin birbirinin önüne geçmesi durumunda tespit edilmesi zorlaşmaktadır.

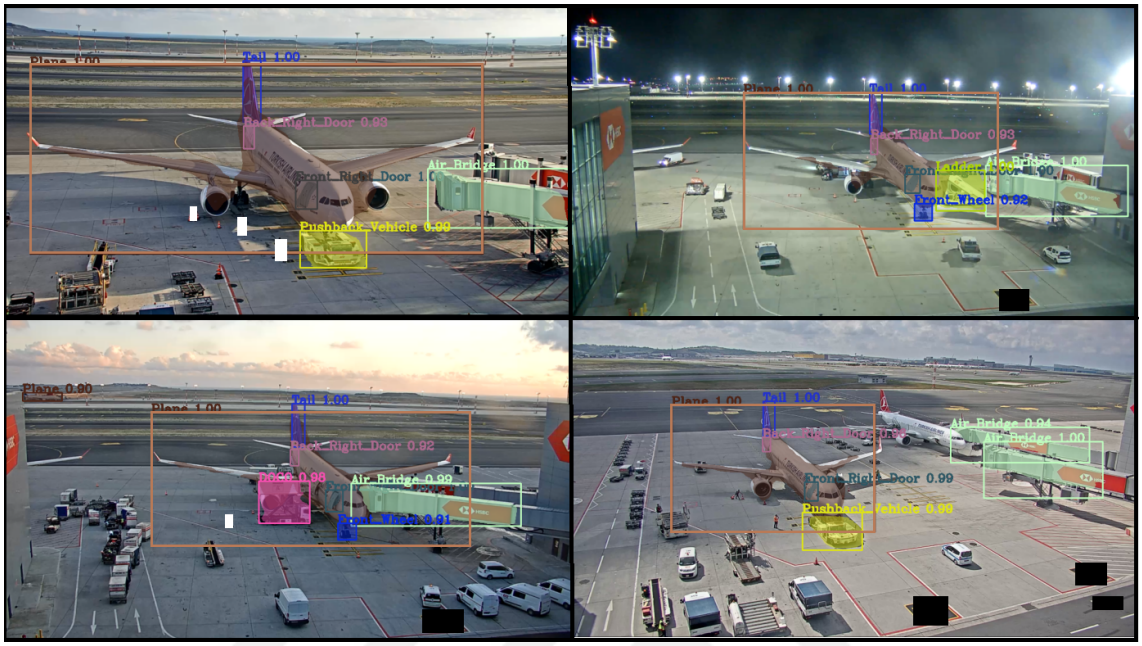
Tablo 4.1 Duyarlılık değerleri

Sınıflar	**Duyarlılık Değerleri**
Uçak	0.98
Ön sağ kapı	0.98
Ön sol kapı	0.76
Arka sol kapı	0.71
Arka sağ kapı	0.89
Kuyruk	1
İkram aracı	0.88
Pushback aracı	0.89
Ön tekerlek	0.87
Yakıt aracı	0.84
Merdiven	0.88
Körük	0.87

Modelin eğitim aşamasında bellek kısıtlamasından dolayı veriler parçalar halinde (mini-küme) modele verilmiştir. Teorik olarak 1'den eğitim kümesindeki tüm verilerin sayısına kadar değer alabilmektedir. Ancak kısıtlar dikkate alınarak model tasarlanırken veri kümesinin boyutuna ve bellek boyutuna göre küme boyutu (batch

size) verilmektedir. Ayrıca sistemin ezberleme yapmaması için eğitim başlamadan önce veri kümesinin karıştırılması yöntemi uygulanmıştır.

Nesne tespiti uygulanan test veri kümesinden örnekler şekil 4.2'de gösterilmektedir.



Şekil 4.2 Nesne Tespit Örnekleri

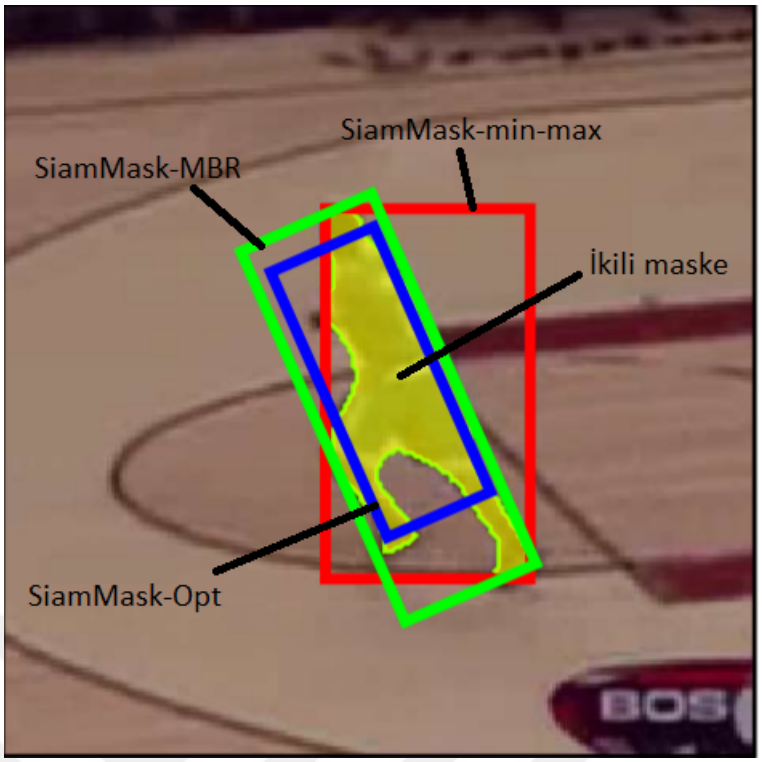
4.3 Nesne Takip Başarısı

SiamMask algoritmasının orjinal çalışmasında sınırlayıcı kutuların segment kullanılarak çıkarılmış farklı çeşitleri üzerinde IoU değerleri hesaplanıştır (Tablo 4.2). Burada çerçevedeki tek nesne üzerinden takip yapıldığı için değerler bu doğrultuda hesaplanmıştır (Şekil 4.3).

Tablo 4.2 VOT-2016 Veri kümesi üzerinde farklı sınırlayıcı kutu gösterim performansları[52]

	mIoU(%)	**mAP@0.5 IOU**	**mAP@0.7 IOU**
SiamFC [53]	50.48	56.42	9.28
SiamRPN [61]	60.02	76.20	32.47
SiamMask-Min-max	65.05	82.99	43.09
SiamMask-MBR	67.15	85.42	50.86
SiamMask-Opt	71.68	90.77	60.47

Birden fazla nesne takibi için IoU değerleri nesne bazında hesaplanmış olarak örnek şekil 4.4'de verilmiştir. Takip algoritmasının ürettiği uçak için döndürülmüş sınırlayıcı kutu döndürülmemiş haline çevrilerek IoU değeri hesaplanmıştır.



Şekil 4.3 Farklı sınırlayıcı kutu gösterimleri[52]

Nesnelerin birbirinin önüne geçtiği ve önünde bir süre kalması durumunda takip algoritması başarılı bir şekilde takip yapamamaktadır. Şekil 4.5'de görüldüğü gibi ikram aracı, yakıt aracının önüne geçtiği durumda nesneler birbiriyle karışmakta ve çerçevedeki başka bir nesnenin ikram aracına yaklaşmasıyla ikram aracının takibi yanlış bir nesne üzerinden devam ederek başarısız sonuçlanmaktadır.

Takip etme başarısı her bir nesne için ayrı IoU hesaplanarak bulunmuştur (Tablo 4.3).

Tablo 4.3 Nesnelerin takip başarısı

Nesne İsmi	mIoU(%)	**AP@0.5 IOU**	**AP@0.7 IOU**
Uçak	60	85	30
İkram Aracı	80	95	85
Yakıt Aracı	72	95	55
Pushback Aracı	86	90	85
Körük	70	75	65

4.4 Operasyon Tespit Başarısı

Sistemin başarısında nesne tespit ve takibi önemli rol oynamaktadır. Sistemdeki operasyon tespit başarısızlıklarının esas sebebinin nesne tespit ve takibine dayandığı

görülmektedir. Tespit ve takip başarılı şekilde yapıldığında olayların tespitinin başarılı olduğu görülmektedir. Körük yanaşması, ikram yüklemesi, pushback, yakıt yüklemesi işlemlerinin değerlendirilmesi gece ve gündüz ortamlardan alınan görüntüler üzerinde test edilmiştir. Gece alınan görüntülerde de ortam ışıklandırmasının olayların tespiti için yeterli olduğu görülmektedir. Olayların gerçek referans(ground-truth) ve tahmin edilen anlardaki saniye farkı tablo 4.4'de gösterilmiştir. Tahmin edilen anlar, gerçek referans anlarına göre gecikmeli olduğundan sonuçlar gecikme süresi olarak gösterilmiştir. Pushback işlemi pushback bağlanma ve başlama olayını içermektedir. Yakıt ve ikram yükleme işlemleri ilgili araçların park pozisyonuna gelmesini, işlemin başlamasını, bitmesi ve aracın ayrılması olaylarını içermektedir. Tespit ve takibin başarılı olduğu durumlarda olaylar başarılı bir şekilde değerlendirilerek başlangıç, bitiş zamanları elde edilmektedir. Körük yanaşması, ikram yüklemesi, yakıt yüklemesi, pushback işlemleri için şekil 4.6'de örnekler gösterilmektedir. Ayrıca sistem tasarımı anormal durumları da tespit edebilecek şekilde yapılmıştır. Fakat veri kümesinde anormal durum bulunmadığı için gerçek veriler üzerinde testler yapılamamıştır.

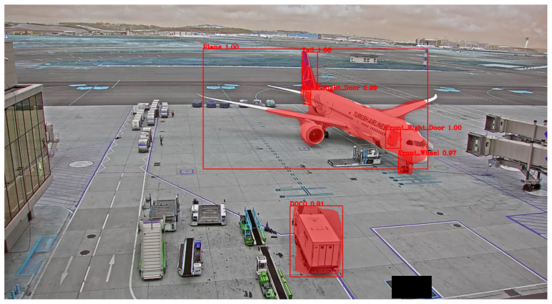
Tablo 4.4 Olayların tespitindeki gecikme süresi

Olay	**Gecikme süresi(sn)**
Uçağın gelmesi	1.8
Pushback işlemi	2.9
Yakıt yükleme işlemi	3.2
İkram yükleme işlemi	2
Körük işlemi	3

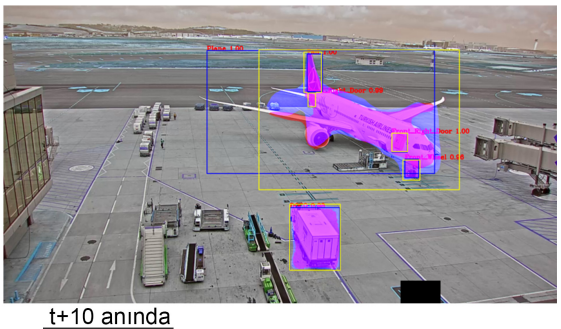
4.5 Kullanılan Kütüphaneler

Derin öğrenme için kullanılan birçok kütüphane bulunmaktadır (Tablo A.1). Bu çalışmada tensorflow, keras ve pytorch kütüphanelerinden yararlanılmıştır. Eğitim aşaması sonucunda ağırlıkların belirlenmesi CPU üzerinden yapılmış ancak maliyet ve zaman açısından GPU gereksiniminden dolayı test aşaması Google Colab platformu üzerinden yapılmıştır. Mask R-CNN ve SiamMask için githubda bulunan açık kaynak kodundan yararlanılmıştır [1].

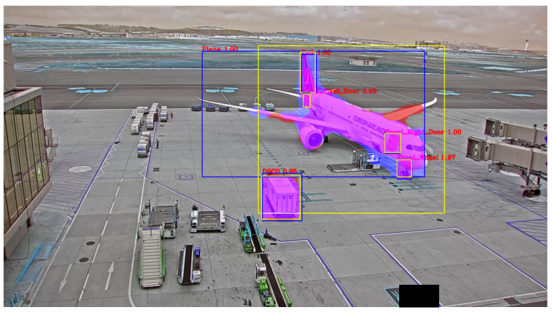
[1]https://github.com/matterport/Mask_RCNN, https://github.com/foolwood/SiamMask



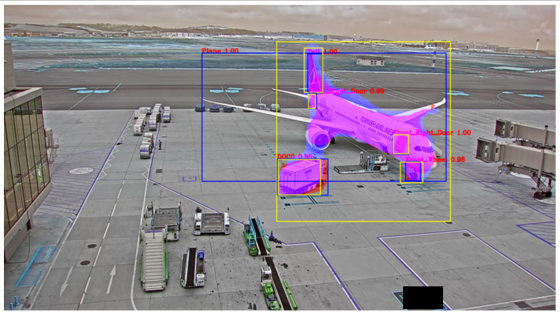
t anında nesnelerin tespiti



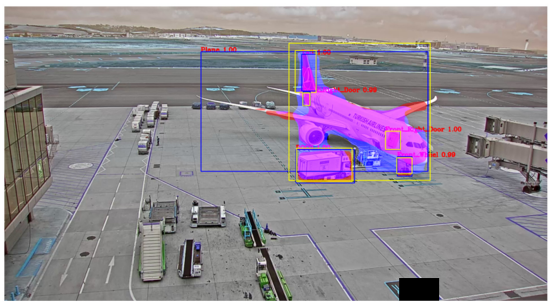
t+10 anında
- Uçak, IoU:56.21
- Sağ ön kapı, IoU:90.53
- Kuyruk, IoU:61.81
- Sağ arka kapı, IoU:83.81
- Ön tekerlek, IoU:63.73
- İkram aracı, IoU:90.24



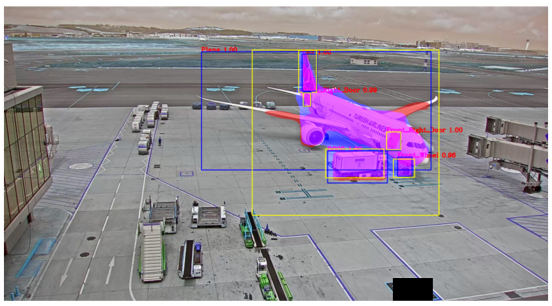
t+40 anında
- Uçak, IoU:54.68
- Sağ ön kapı, IoU:90.84
- Kuyruk, IoU:78.58
- Sağ arka kapı, IoU:88.61
- Ön tekerlek, IoU:84.86
- İkram aracı, IoU:85.22



t+70 anında
- Uçak, IoU:53.27
- Sağ ön kapı, IoU:95.18
- Kuyruk, IoU:68.41
- Sağ arka kapı, IoU:77.41
- Ön tekerlek, IoU:57.76
- İkram aracı, IoU:81.39

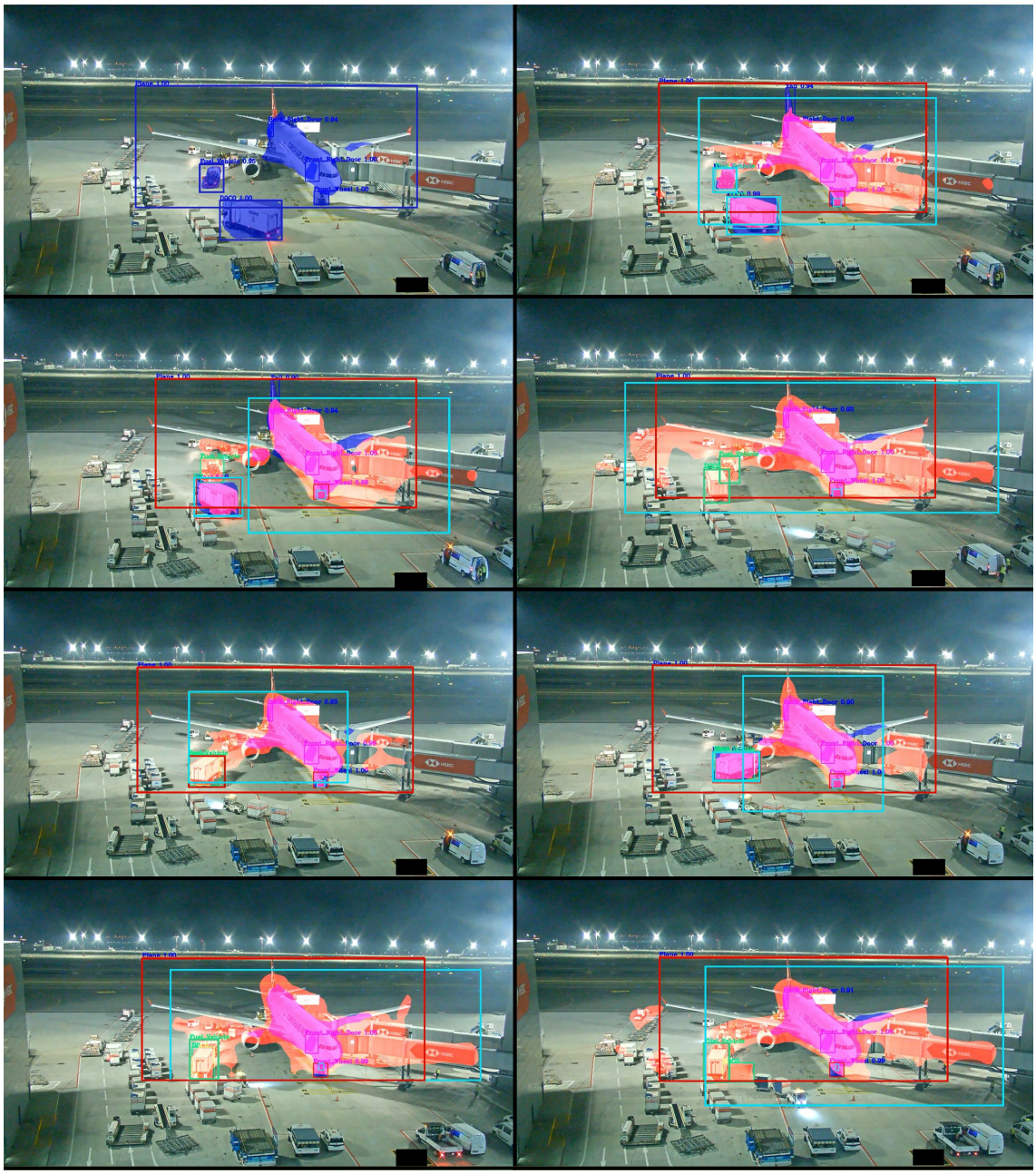


t+90 anında
- Uçak, IoU:55.59
- Sağ ön kapı, IoU:96.65
- Kuyruk, IoU:59.82
- Sağ arka kapı, IoU:83.39
- Ön tekerlek, IoU:78.70
- İkram aracı, IoU:87.94

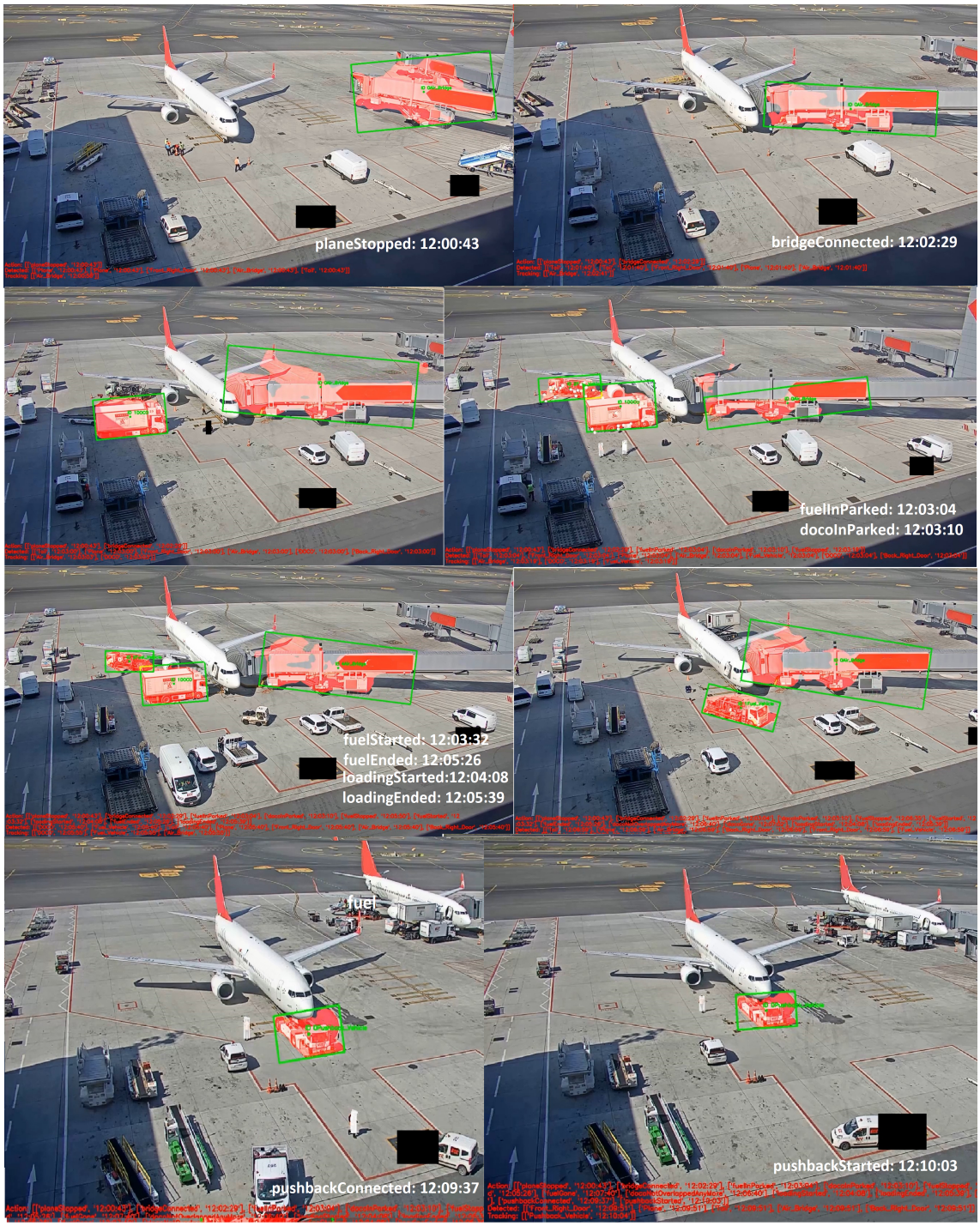


t+170 anında
- Uçak, IoU:57.34
- Sağ ön kapı, IoU:93.63
- Kuyruk, IoU:69.69
- Sağ arka kapı, IoU:88.74
- Ön tekerlek, IoU:59.33
- İkram aracı, IoU:72.85

Şekil 4.4 Nesne Takip örnek IoU değerleri



Şekil 4.5 Nesnelerin birbirinin önüne geçerek takibin başarısız olduğu örnek - Takip etme algoritmasından elde edilen sınırlayıcı kutu mavi renk, gerçek referans(ground-truth) kırmızı ile gösterilmiştir.



Şekil 4.6 Operasyon Tespiti

5
SONUÇ VE ÖNERİLER

Bu tez çalışmasında Türk Hava Yolları Teknoloji AR-GE Merkezi'nden alınan apron görüntüsü verileri kullanılarak operasyonel çevrim süreçlerinin doğru ölçümü ve optimizasyonunu amaçlayan derin öğrenme tabanlı bir sistem tasarlanmış ve gerçeklenmiştir. Gerçekleştirilen sistem ile apron görüntüleri üzerinde tespit, takip ve analiz yapılmıştır. Sistem modüler olarak tasarlanmış, yeni bir nesne eklenmek istendiğinde sistem karmaşıklığını artırmayacak şekilde gerçeklenmiştir. Sistem çıktısında olay analizi yapılarak log çıktısı üretilmektedir. Log çıktısı tespit etme, takip etme ve analiz yani sistemi tamamıyla kapsamaktadır. Ayrıca bir olayın beklenenden uzun sürmesi durumunda uyarı mesajları da verilmektedir. Akış içerisinde yer alan nesne tespiti, nesne takibi üzerinden analizler yapıldığı için tespit ve takibin başarısı sistemin başarısını etkilemektedir. Bu sebeple nesne tespitinin, çoklu nesne takibinin ve olayların testi ayrı olarak yapılmıştır. Kısıtlı veri kümesiyle bile umut verici sonuçlar elde edilmiştir.

Bu çalışmada farklı apron alanlarından alınan tek kamera görüntüsü üzerinde çalışılmıştır. Bu çalışmanın devamında birden fazla kamera kullanılarak bütün operasyonların tespiti üzerinde çalışılması planlanmaktadır. Ayrıca tespit/takip aşamasındaki problemlerin incelenerek iyileştirilmesine yönelik çalışmalar yapılacaktır. Bunların yanında veri kümesinin anormal durumları içerecek şekilde genişletilmesi planlanmaktadır.

KAYNAKÇA

[1] Y. Bai, Q. Liu, L. Xie, W. Qiu, Y. Zheng, A. L. Yuille, "Semantic part detection via matching: Learning to generalize to novel viewpoints from limited training data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7535–7545.

[2] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] W. Shi, R. van de Zedde, H. Jiang, G. Kootstra, "Plant-part segmentation using deep learning and multi-view vision," *Biosystems Engineering*, vol. 187, pp. 81–95, 2019.

[4] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[5] C. Wang, Y. Dai, W. Zhou, Y. Geng, "A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition," *Journal of advanced transportation*, vol. 2020, 2020.

[6] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] N. Algethami, S. Redfern, "A robust tracking-by-detection algorithm using adaptive accumulated frame differencing and corner features," *Journal of Imaging*, vol. 6, no. 4, p. 25, 2020.

[8] G. Khan, Z. Tariq, M. U. G. Khan, "Multi-person tracking based on faster r-cnn and deep appearance features," in *Visual Object Tracking with Deep Neural Networks*, IntechOpen, 2019.

[9] S. Ren, K. He, R. Girshick, J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, IEEE, 2016, pp. 3464–3468.

[11] N. Wojke, A. Bewley, D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, IEEE, 2017, pp. 3645–3649.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

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

[14] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[15] P. Thai, S. Alam, N. Lilith, P. N. Tran, B. N. Thanh, "Deep4air: A novel deep learning framework for airport airside surveillance," *arXiv preprint arXiv:2010.00806*, 2020.

[16] D. Thirde, M. Borg, J. M. Ferryman, F. Fusier, V. Valentin, F. Brémond, M. Thonnat, O. Team, "A real-time scene understanding system for airport apron monitoring.," in *ICVS*, vol. 6, 2006, p. 26.

[17] J. Shi *et al.*, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, IEEE, 1994, pp. 593–600.

[18] S. H. G. Müdürlüğü, *Apron operasyonları emniyet kılavuzu*, `http://web.shgm.gov.tr/documents/sivilhavacilik/files/pdf/kurumsal/yayinlar/Apron_Operasyonlari_Emniyet.pdf`, Accessed: 2020-01-19.

[19] B. Kröse, B. Krose, P. van der Smagt, P. Smagt, "An introduction to neural networks," 1993.

[20] S. Yavuz, M. Deveci, "Đstatiksel normalizasyon tekniklerinin yapay sinir ağin performansina etkisi," *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, no. 40, pp. 167–187, 2012.

[21] İ. Özkan, E. Ülker, "Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri," *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, vol. 6, no. 3, pp. 85–104, 2017.

[22] L. Deng, D. Yu, "Deep learning: Methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[23] S. Gazel, C. T. BATİ, "Derin sinir ağları ile en iyi modelin belirlenmesi: Mantar verileri üzerine keras uygulaması," *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, vol. 29, no. 3, pp. 406–417, 2019.

[24] D. Britz, *Understanding convolutional neural networks for nlp*, `http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/`, Accessed: 2020-12-20, 2015.

[25] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, S. M. Wild, "Deephyper: Asynchronous hyperparameter search for deep neural networks," in *2018 IEEE 25th international conference on high performance computing (HiPC)*, IEEE, 2018, pp. 42–51.

[26] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Springer, 2010, pp. 177–186.

[27] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[28] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Gülcü, K. Zeki, "Konvolüsyonel sinir ağlarında hiper-parametre optimizasyonu yöntemlerinin incelenmesi," *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, vol. 7, no. 2, pp. 503–522, 2019.

[30] S. Sharma, "Activation functions in neural networks," *Towards Data Science*, vol. 6, 2017.

[31] P. K. Mallick, V. E. Balas, A. K. Bhoi, A. F. Zobaa, "Cognitive ınformatics and soft computing," *Proceeding of CISC 2017, Springer Nature Singapore Pte Ltd*, 2020.

[32] Y. Wang, Y. Li, Y. Song, X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020.

[33] R. Lienhart, J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings. international conference on image processing*, IEEE, vol. 1, 2002, pp. I–I.

[34] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[36] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[38] J. Redmon, A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

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

[40] *Pr-207: Yolov3: An ıncremental ımprovement*, Accessed: 2021-06-01. [Online]. Available: `https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement`.

[41] *[pr12] you only look once (yolo): Unified real-time object detection*, Accessed: 2021-06-01. [Online]. Available: `https://www.slideshare.net/TaegyunJeon1/pr12-you-only-look-once-yolo-unified-realtime-object-detection`.

[42] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[43] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[44] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[45] B. Yilmaz, M. E. Karsligıl, "Detection of airplane and airplane parts from security camera ımages with deep learning," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2020, pp. 1–4.

[46] A. N. Ömeroğlu, N. Kumbasar, E. A. Oral, I. Y. Ozbek, "Mask r-cnn algoritması ile hangar tespiti hangar detection with mask r-cnn algorithm," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2019, pp. 1–4.

[47] G. L. Foresti, L. Snidaro, "Vehicle detection and tracking for traffic monitoring," in *International Conference on Image Analysis and Processing*, Springer, 2005, pp. 1198–1205.

[48] O. Sidla, Y. Lypetskyy, N. Brandle, S. Seer, "Pedestrian detection and tracking for counting applications in crowded situations," in *2006 IEEE International Conference on Video and Signal Based Surveillance*, IEEE, 2006, pp. 70–70.

[49] C. Kim, J.-N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 2, pp. 122–129, 2002.

[50] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[51] M. Y. Abbass, K.-C. Kwon, N. Kim, S. A. Abdelwahab, F. E. A. El-Samie, A. A. Khalaf, "A survey on online learning for visual tracking," *The Visual Computer*, vol. 37, pp. 993–1014, 2021.

[52] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[53] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, Springer, 2016, pp. 850–865.

[54] *Vot2018 challenge*, Accessed: 2021-06-01. [Online]. Available: `https://www.votchallenge.net/vot2018/`.

[55] *Pascal voc dataset mirror*, Accessed: 2021-06-01. [Online]. Available: `https://pjreddie.com/projects/pascal-voc-dataset-mirror/`.

[56] Accessed: 2021-05-01. [Online]. Available: `https://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html`.

[57] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[59] A. B. Amjoud, M. Amrouch, "Convolutional neural networks backbones for object detection," in *International Conference on Image and Signal Processing*, Springer, 2020, pp. 282–289.

[60] C. Goutte, E. Gaussier, “A probabilistic interpretation of precision, recall and f-score, with implication for evaluation,” in *European conference on information retrieval*, Springer, 2005, pp. 345–359.

[61] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, “Distractor-aware siamese networks for visual object tracking,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.

A
DERİN ÖĞRENME KÜTÜPHANELERİ

Tablo A.1'de, derin öğrenme için geliştirilen kütüphanlerin bazıları listelenmiştir[1].

Tablo A.1 Bazı derin öğrenme kütüphaneleri ve ek bilgileri

Kütüphane	**Platform**	**Yazıldığı dil**	**İlk sürüm**
PyTorch	Linux, macOS, Windows	Python, C, C++, CUDA	2016
TensorFlow	Linux, macOS, Windows, Android	C++, Python, CUDA	2015
Keras	Linux, macOS, Windows	Python	2015
Flux	Linux, MacOS, Windows	Julia	2017
Theano	Çapraz platform	Python	2007
Torch	Linux, macOS, Windows, Android,iOS	C,Lua	2002
Caffe	Linux, macOS, Windows	C++	2013
Deeplearning4j	Linux, macOS, Windows, Android	C++, Java	2014

[1]https://en.wikipedia.org/wiki/Comparison_of_deep-learning_software

B
HAVACILIK TERİMLERİ

Tablo B.1'de bazı havacılık terimleri[1] listelenmiştir.

Tablo B.1 Bazı havacılık terimleri

Türkçe Terim	**İngilizce Terim**	**AÇIKLAMA**
Taksi	Taxi	Uçağın yerde yaptığı hareket anlamındadır. Park yerinden kalkış koşusuna başlayana kadar olan kalkış taksi ve geliş taksi kavramlarını içine alan safhadır.
Yer güç ünitesi (jenaratör)	Ground power unit (GPU)	Dışarıdan sağlanan harici elektrik kaynağı.
Havalandırma aracı	Air condition unit (ACU)	Uçaklara klima hizmetinde kullanılan havayı vermek üzere imal edilmiş özel donanımlı araç.
Uluslararası Hava Taşımacılığı Birliği	International Air Transport Association (IATA)	Hava taşımacılığının güvenilir, ekonomik ve düzenli yapılabilmesi, hava ticaretinin gelişmesi ve ilgili problemlerin çözülmesi, ICAO ve diğer uluslararası organizasyonlarla işbirliği sağlamak birliğin hedefleri arasındadır.
Uluslararası Sivil Havacılık Örgütü	International Civil Aviation Organization (ICAO)	Uluslararası sivil havacılığın güvenli ve düzenli bir şekilde büyümesini sağlamak, barışçıl amaçlara yönelik uçak tasarımı ve işletmesini teşvik etmek, sivil havacılık için havayolları, havaalanları ve hava seyir tesislerinin gelişimini desteklemek, uluslararası kamuoyunun güvenli, düzenli, verimli ve ekonomik hava taşımacılığı ihtiyaçlarını karşılamak amacıyla 1944'de Birleşmiş Milletler'in bir kuruluşu olarak kurulan ve halen çalışmalarına devam eden, Türkiye'nin de üye olduğu bir örgüttür.
Kargo alanı	Cargo area	Kargo işlemleri için tahsis olunan tüm yer, alan ve tesisler. Buna apron sahaları, kargo binaları ve depoları, araç park yerleri ve bunlarla bağlantıyı sağlayan yollar dahildir.

[1] https://www.dhmi.gov.tr/Sayfalar/Sozluk.aspx

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. B. Yilmaz, M. E. Karsligıl, "Detection of airplane and airplane parts from security camera images with deep learning," in 2020 28th Signal Processing and Communications Applications Conference (SIU), IEEE, 2020, pp. 1–4.