



## Makale Türü (Araştırma Makalesi)

## İnsansız hava aracından çekilen videolar kullanılarak derin öğrenme yaklaşımı ile nesne tespiti

*Object detection by deep learning approach using images taken from unmanned aerial vehicle*Aydın USTA<sup>1</sup>, Muhammet Ali ARSERİM<sup>2</sup><sup>1</sup>Dicle Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, [aysancetin@gmail.com](mailto:aysancetin@gmail.com), Orcid No:0000-0002-9190-5061<sup>2</sup>Dicle Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, [marserim@dicle.edu.tr](mailto:marserim@dicle.edu.tr), Orcid No:0000-0002-9913-5946

## MAKALE BİLGİLERİ

## ÖZ

## Makale Geçmişi:

Geliş 20 Ekim 2022  
Revizyon 23 Aralık 2022  
Kabul 18 Mart 2023  
Online 23 Mart 2023

## Anahtar Kelimeler:

İHA, YoloV3, Darknet-53,

Günümüzde, İnsansız Hava Araçları (İHA) sınır güvenliği, sahil güvenliği, savunma, saldırı başta olmak üzere arama kurtarma, zirai ilaçlama, yangın söndürme gibi geniş bir kullanım alanına sahiptir. Bununla beraber İHA'nın bazı görevleri otonom bir şekilde yerine getirebilmesi ise bilgisayarlı görü sisteminin buna entegrasyonu ile olur. Bu alandaki uygulamalardan biri olan havadan nesne tespiti uygulamaları, uzaklık, yakınlık kavramlarına bağlı olarak farklı boyutlardaki nesnelere tespit edememe, yavaş tespit, yanlış tahminleme gibi çeşitli hatalar içerebilir. Derin Öğrenme (DÖ) uygulamaları ile bu hatalar en aza indirilebilir. Bu çalışmada VRAT[1] video setinden alınan görüntülerle YOLOv3 DÖ ağı eğitilmiş ve daha sonra DJI Mavic 2 Zoom İHA kamerasından elde edilen görüntülerle tekrar eğitim yapılarak videodaki araçların ve yayaların tespiti sağlanmıştır. Eğitim ve test süreci Google Colab Tesla T4 GPU makinesinde gerçekleştirilmiştir. Modelin performansı ilk ve ikinci eğitim için Loss, mAP 2.345, %79 ve 1.171, %70.09 olarak bulunmuştur.

## ARTICLE INFO

## ABSTRACT

## Article history:

Received 20 October 2022  
Received in revised form 23  
December 2023  
Accepted 18 March 2023  
Available online 23 March  
2023

## Keywords:

UAV, YOLOV3, Darknet-53

Today, Unmanned Aerial Vehicles (UAV) have a wide range of uses such as border security, coast guard, defense, attack, search and rescue, agricultural spraying, fire extinguishing. However, the ability of the UAV to perform some tasks autonomously is possible with the integration of the computer vision system. Aerial object detection applications, which is one of the applications in this field, may contain various errors such as not being able to detect objects of different sizes, slow detection, wrong estimation, depending on the concepts of distance and proximity. These errors can be minimized with Deep Learning (DL) applications. In this study, the YOLOv3 DL network was trained with the images taken from the VRAT[1] video set, and retrained with the images obtained from the DJI Mavic 2 Zoom UAV camera to detect the vehicles and pedestrians in the video. The training and testing process was carried out on the Google Colab Tesla T4 GPU machine. The performance of the model for the first and second training was found as 2.345, 79% and 1.171, 70.09% for loss, and mAP values.

Doi: 10.24012/dumf.1191160

\* Sorumlu Yazar

## Giriş

İnsansız Hava Aracı (İHA), bir operatör tarafından uzaktan kumanda edilebilen veya bir uçuş rotası boyunca otomatik hareket edebilen, göreve ilişkin faydalı yük taşıyabilen ve bir takım görevi otonom olarak yerine getirebilen bir hava aracıdır [2]. İHA'lar başlangıçta gözlem, saldırı ve savunma amaçlı üretilip kullanılsa da günümüzde gözetleme, keşif, taşımacılık, zirai ilaçlama, yangın söndürme, enerji nakil hatlarının bakım ve kontrolü, muhabirlik, fotoğraf ve video çekme gibi geniş bir kullanım yelpazesine sahiptir. Bu kullanım alanlarının birçoğunda İHA'nın otonom olması, yani bir insan gibi görüp, değerlendirip, karar verip, aksiyon alması beklenir. Özellikle de yangın söndürme, sınır güvenliği, sahil güvenliği gibi kritik durumlarda, mesafeden veya başka sebeplerden dolayı komuta merkezi ile iletişim anında oluşabilecek ve sonuçları tolere edilemeyecek gecikmenin yaşanmaması adına, otonom karar ve aksiyon mekanizması, üzerinde durulması gereken çok önemli bir detaydır.

“Drone ile Çekilmiş Videolarda Derin Öğrenme Tabanlı İnsan ve Araç Tespiti” adlı çalışmada sinir ağı olarak YOLOv3 DarkNet-53 kullanılmıştır. Ağın 1000 adet resim ile eğitilmiş ve %78 civarında bir doğruluk oranına ulaşılmıştır [3]. “İnsansız Hava Araçlarının Derin Öğrenme Temelli Nesne Tespiti ve Tanınması” adlı bir diğer çalışmada Faster-RCNN ve YOLOv4'ün karşılaştırılması amaçlanmıştır. Tehdit edici unsurların tespiti ve tanınması için İHA'dan elde edilen 2595 adet askeri operasyon görüntüsü kullanılmıştır. Nesne tespiti ve tanınmasında Faster-RCNN mimarisinde yaklaşık %93 doğruluk oranı elde edilirken, YOLOv4 mimarisinde yaklaşık %88 oranında doğruluk elde edilmiştir [4]. DÖ tekniklerinden YOLOv2 modeli kullanılarak İHA'lardan elde edilen görüntülerdeki araçların tespiti üzerine yapılan çalışmada %77 doğruluk oranı elde edilmiştir [5]. İHA ile gerçek zamanlı ve bulut tabanlı nesne tespiti için Pascal VOC2012 veri seti ile eğitilen algoritmalarından Faster R-CNN ile %83.9, SSD300 ile %81.6, SSD500 %82.6, YOLO ile %78.3 ve Fast YOLO ile %79.4 doğruluk oranına ulaşılmıştır [6].

Çalışmanın kapsamı Derin Öğrenme (DÖ) algoritmalarından YOLOv3 algoritması kullanılarak İHA'dan alınan görüntüdeki araçların ve yayaların tespitidir. Giriş bölümünde, bu alanda yapılan benzer çalışmalara, Materyal ve Metot bölümünde kullanılan modelin başarımlarının hesaplanmasına ve YOLOv3 mimarisine ve çalışma prensibine değinilmiştir. Deneysel Sonuçlar ve Tartışma bölümünde yapılan çalışmalar ve elde edilen sonuçlar tartışılmıştır. Son Bölümde, elde edilen sonuçların değerlendirilmesi yapılmıştır.

## Materyal ve Metot

### DJI Mavic 2 Zoom Drone

Bu çalışmada, döner kanatlı İHA olan DJI Mavic 2 Zoom Drone kullanılmıştır. 31 dk uçuş süresine sahip olan Mavic 2 Zoom 1/2.3-inch 12-megapixel sensör ve 4x zoom ile güçlendirilmiş Zoom Kamera, 2x optik zoom (24–48 mm) lense sahiptir [7]. Şekil 1'de, çalışmada kullanılan Mavic 2 Zoom Drone görülmektedir.

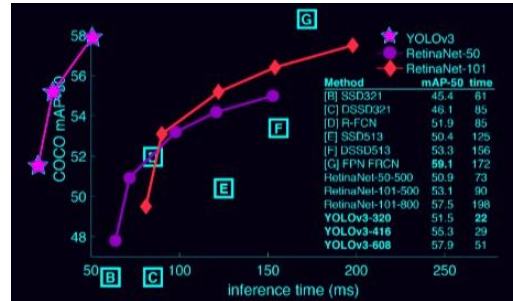


Şekil 1. DJI Mavic 2 Zoom Drone [7]

### YOLOv3 Mimarisi ve Çalışma Prensibi

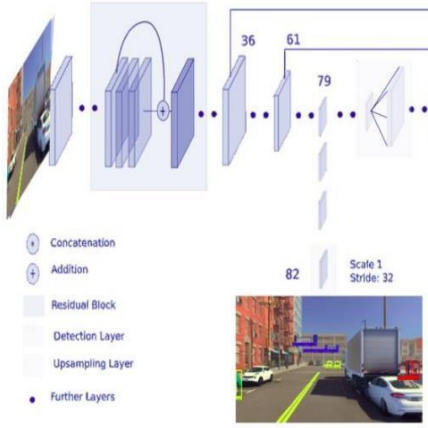
Nesne tespit, tanıma ve takibi işlemleri oldukça zor süreçlerdir. Görüntü üzerinde kayan bir pencere oluşturularak nesne aranır. Bu arama işlemi sırasında, nesneyi tanımlayan öznelik haritalarının oluşturulma işlemleri ve sınıflandırma işlemleri defalarca yapılır. Tüm bunlar ağır hesapsal yüküdür ve zaman gerektirir. Bununla beraber özellikle tespit, tanıma ve takip işlemlerinin havadan yapılması durumunda uzaklık, yakınlık, yükseklik kavramlarına bağlı olarak farklı boyutlardaki nesnelere tespit edememe, yavaş tespit, yanlış tahminleme gibi birkaç hata gündeme gelir.

YOLOv3, C/Cuda kullanılarak geliştirilmiş olan, temel katman sayısı 53 olan Darknet-53 framework'ü üzerinde çalışır. Bu sayede oldukça yüksek performans gösterir [8]. Darknet-53, girdi olarak bir görüntü alan ve öznelik haritasını çıkaran, YOLOv3 için bir omurgadır [9]. Şekil 2'de YOLOv3 ve diğer algoritmaların COCO veri setinde 0.5 IoU (mAP-50) ile karşılaştırması görülmektedir. Grafikten de anlaşılacağı üzere YOLO rakiplerine karşı süre ve doğruluk açısından çok iyi durumdadır [10].

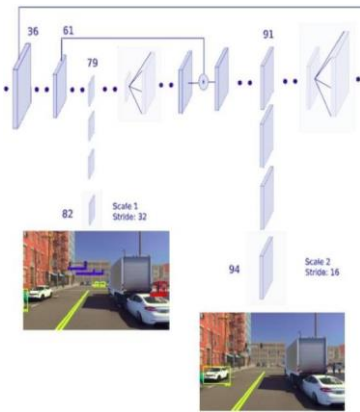


Şekil 2. YOLOv3 vs Diğer Algoritmalar [11].

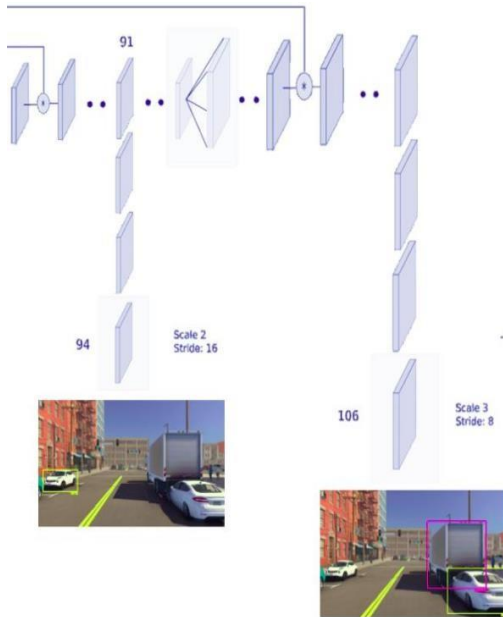
YOLOv3 algoritması üç ölçekte tahminler yapar. Şekil 3'deki 13×13 katmanı (Bölge 82) büyük maske kullanan en büyük tahmin ölçeğidir ancak küçük nesnelere tahmin edebilir. Şekil 4'deki 26×26 katmanı (Bölge 94) orta maske kullanan orta büyüklükteki tahmin ölçeğidir, orta büyüklükteki nesnelere tahmin edebilir. Şekil 5'deki 52×52 katmanı (Bölge 106) en küçük tahmin ölçeğidir, daha küçük bir maske ile daha büyük nesnelere tahmin edebilir [9]. Yani Bölge 82, Bölge 94, Bölge 106, üç farklı ölçekte (82, 94, 106) tahmin edilen farklı büyüklükteki parametreleri temsil eder [12].



Şekil 3. 13×13 katmanı Bölge 82 [9].



Şekil 4. 13×13 katmanı Bölge 82 ve 26×26 katmanı Bölge 94 [9]



Şekil 5. 26×26 katmanı Bölge 94 ve 52×52 katmanı Bölge 106 [9]

Bununla beraber YOLO, farklı boyuttaki nesnelere, sabitleme kutularını (Anchor Box) kullanarak öğrenir. Sabitleme Kutuları, gelişmiş performans için ayarlanması gereken en önemli parametrelerden biridir. Algoritma çalışırken çok fazla, gereksiz sınırlayıcı kutular, bir nesne için birden fazla sınırlayıcı kutu oluşacaktır [13]. Bu durumda maksimum olmayı önleme (non-maximum suppression) denen bir teknik uygulanır ve maksimum güven puanına sahip sınırlayıcı kutu dışındaki sınırlayıcı kutular silinir. Bahsi geçen, hesapsal yük ve zaman problemlerinin çözümüne yönelik bir mimariye ve çalışma prensibine sahip olması açısından bu çalışmada YOLOv3 kullanılmıştır.

### Veri Seti

Veri Seti olarak Mavic 2 Zoom dan alınan videolar ve VIRAT [1] tarafından üretilen video veri setleri kullanılmıştır. Videolar ffmpeg yazılımı ile frame'lere ayrılmış, Mavic 2 Zoom videolarından 377 resim, VIRAT [1] videolarından 500 resim, toplamda 877 resim kullanılmıştır. Resimlerin etiketleme işlemi LabelImg [14] aracı ile yapılmış olup, yazılan bir kod bloğu yardımı ile %90'i eğitim seti, %10'u doğrulama (test) seti olacak şekilde rassal olarak ayrılmıştır.

### Modelin Başarım Ölçütleri ve Hesaplanması

Sınıflandırma çalışmalarında bir modelin başarısını ölçmek ve sayısal olarak ifade etmek için doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-score gibi metrikler kullanılır. Ancak, bu çalışmada nesnelere sınıflandırılmasına ek olarak konumlarının bulunması da amaçlandığından başarımlar ölçüsü olarak bir sınıf olduğu durumda ortalama kesinlik değeri (Average Precision - AP) birden fazla sınıf olduğu durumlarda ise genel ortalama kesinlik değeri (mean average precision - mAP) kullanılır.

**Doğruluk (Accuracy):** Bir modelin başarısını ölçmek için çok kullanılan ancak tek başına yeteli olmayan bir metriktir.

$$Accuracy = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{FP} + N_{TN} + N_{FN}} \quad (1)$$

**Kesinlik (Precision):** Modelin yaptığı "pozitif" tahminlerin, gerçekte kaçının "pozitif" olduğunu gösteren bir metriktir.

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (2)$$

**Duyarlılık (Recall):** Modelin yapması gereken "pozitif" tahminlerden, gerçekte kaçını pozitif tahminlediğini gösteren bir metriktir.

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (3)$$

**F1-score:** Eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamak için doğruluk (accuracy) yerine tercih edilen bir metriktir. F1-skoru, kesinlik değerinin ve duyarlılık değerinin harmonik ortalaması alınarak hesaplanır.

$$F1 - score = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

**Ortalama Kesinlik Değeri (Average Precision):** Ortalama kesinlik değeri (AP), kesinlik-duyarlılık (precision-recall) eğrisi altında kalan alandır. Modelin tahminlerinin ne kadar doğru olduğunu ölçer, yani doğru olan tahminlerin yüzdesel gösterimidir.

$$AP = \int_0^1 p(r) dr \quad (5)$$

**Genel Ortalama kesinlik değeri (Mean average precision):** AP, her nesne sınıfı için hesaplanır. Sınıf sayısının birden fazla

olması durumunda ise genel ortalama kesinlik değeri yani mAP değeri hesaplanır. Bu da tüm nesne sınıflarına ait AP değerinin ortalamasıdır.

Model mAP değerine göre değerlendirilir, mAP modelin hassasiyetinin iyi bir ölçüsüdür.

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (6)$$

### Deneysel Sonuçlar ve Tartışma

Çalışmada kullanılacak olan Darknet-53 framework'ü klonlanır ve derlenir. Bu işlemin yapıldığı yerde "darknet" adında bir klasör oluşur. Bu klasör eğitim işlemi için Google Colab'da kullanılacak olan klasördür.

i) "darknet" klasörü içerisine "custom\_data" adında bir klasör oluşturulur.

ii) "custom\_data" klasörü içerisine "cfg" ve "images" adında iki klasör, "custom.names", "train.txt", "test.txt" ve "detector.data" adlarında dört dosya oluşturulur.

iii) "cfg" adlı dosya, sinir ağının konfigürasyon dosyasıdır.

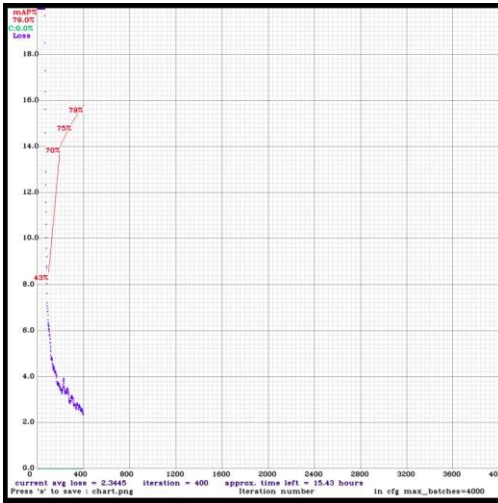
iv) "images" adlı dosya içerisinde eğitim ve test işlemleri için kullanılacak olan resimler ve resimlerin etiketleri bulunur.

v) "custom.names" adlı dosya içerisine ağın öğrenmesi ve tespit etmesi istenen objelerin sınıf isimleri "car" ve "person" şeklinde yazılır.

vi) "train.txt" ve "test.txt" dosyaları içerisinde eğitim ve doğrulama için kullanılacak olan "images" klasöründeki resimlerin yolları yazılır.

vii) "detector.data" adlı dosya içerisine ise eğitim esnasında ihtiyaç duyulacak "custom.names", "train.txt" ve "test.txt" dosyalarının erişim yolları, sınıf isimlerinin sayısı ve eğitim esnasında eğitilen ağırlıkların tutulacağı dosyanın yolu yazılır.

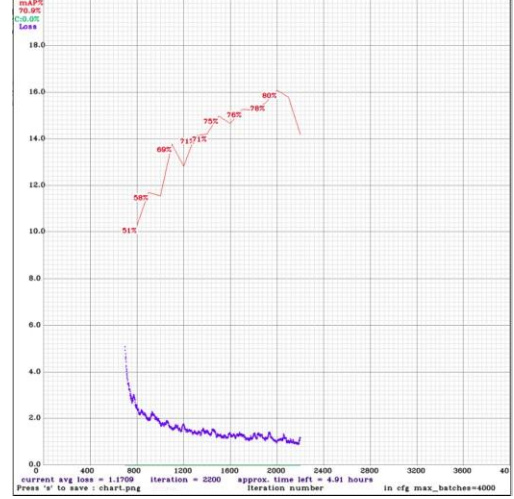
Bahsi geçen "cfg" adlı dosya içerisinde bazı konfigürasyonlar yapıldıktan sonra, eğitim işlemi yapılmıştır. Elde edilen mAP değerleri Şekil 6'da gösterilmektedir.



Şekil 6. İlk Eğitim mAP Grafiği

Şekil 6'deki mAP grafiğinde de görüldüğü gibi ilk eğitim 15.43 saat sürmüş, ortalama loss değeri 2.345 ve mAP değeri %79 olacak şekilde bir başarı elde edilmiştir.

Ancak ilk eğitim sonrası loss değeri istenen şekilde küçülmediğinden, ağ yeniden eğitime tabi tutulmuş ve Şekil 7'deki değerler elde edilmiştir. Şekilden görüldüğü üzere eğitim yaklaşık 5 saat sürmüş ve loss değeri 1'e yaklaşmıştır. Bununla beraber mAP değerinin önceki değere kıyasla %71'e düştüğü görülmektedir.



Şekil 7. Yeniden eğitim sonrası mAP grafiği

Eğitim ve yeniden eğitim süreçlerinde kullanılan veri seti sayıları ve eğitim sonrası elde edilen loss ve mAP değerleri Tablo 1'de görülmektedir.

Tablo 1: Eğitim - yeniden eğitim süreçlerinde kullanılan veri setleri ile eğitim sonrası elde edilen loss ve mAP değerleri

	Veri Seti	Eğitim Seti	Doğrulama Seti	Loss Değeri	mAP Değeri(%)
İlk Eğitim	500	450	50	2.345	79
Yeniden Eğitim	377	339	388	1.171	70.9

Tablo 2'de, elde edilen mAP değerlerinin sonuçlarının bu çalışma ile neredeyse birebir aynı olan YOLOv3 DarkNet-53 kullanılarak araç ve yaya tespiti yapan "Drone ile çekilmiş Videolarda Derin Öğrenme Tabanlı İnsan ve Araç Tespiti" adlı çalışma [3] ile karşılaştırma tablosu görülmektedir.

Tablo 2. Çalışma sonuçlarının [3] ile karşılaştırılması.

	Eğitim Seti	Öğrenme Katsayısı	Küme Boyutu	Loss Değeri	mAP Değeri(%)
İlk Eğitim	500	0.001	32	2.345	79
Yeniden Eğitim	377	0.001	32	1.171	70.9
[3]	1000	0.001	64	--	78.84
[3]	1000	0.001	128	--	77.23

Tablodan da görüldüğü gibi başarı oranı %70-80 arasında olmaktadır. Bununla birlikte eğitim verisi arttıkça başarı oranı arttırmakta ancak yapılan yeni eğitimde bir önceki eğitime göre başarı oranı azalmaktadır.

Eğitim ve Yeniden eğitim sonrası elde edilen ağırlıklar test işlemine tabi tutulduklarında Şekil 8'de İlk eğitim sonrası oluşan ağırlıkların resim üzerinde test sonuçları, Şekil 9'da ise Yeniden eğitim sonrası oluşan ağırlıkların resim üzerinde test sonuçları görülmektedir.





Şekil 8. İlk eğitim sonrası oluşan ağırlıkların resim üzerinde test edilmesi



Şekil 9. Yeniden eğitim sonrası oluşan ağırlıkların resim üzerinde test edilmesi

Şekil 10 ve Şekil 11'de ise İlk eğitim ve Yeniden eğitim sonrası oluşan ağırlıkların video üzerinde test sonuçları görülmektedir.



Şekil 10. İlk Eğitim sonrası oluşan ağırlıkların video üzerinde test edilmesi



Şekil 11. Yeniden eğitim sonrası oluşan ağırlıkların video üzerinde test edilmesi

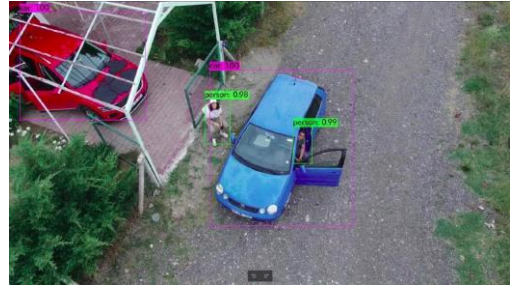
İlk eğitim sonrası oluşan ağırlıklar ile yeniden eğitim sonrası oluşan ağırlıklar aynı resim ile test edilmek istendiğinde Şekil 12 ve Şekil 13'de testin resim üzerindeki sonucu, Tablo 3 ve Tablo 4'de ise testin süre ve yüzde bazında ağ çıktısı görülmektedir.



Şekil 12. İlk eğitim sonrası oluşan ağırlıkların resim üzerindeki tespitleri

Tablo 3. Şekil 12'deki tespitlerin süre ve yüzde bazlı ağ çıktısı

Tahmin Süresi(ms)	602.322
car	%38
car	%28
car	%32
car	%37



Şekil 13. Yeniden eğitim sonrası oluşan ağırlıkların resim üzerindeki tespitleri

Tablo 4. Şekil 13'deki tespitlerin süre ve yüzde bazlı ağ çıktısı

Tahmin Süresi(ms)	599.079
car	%100
person	%98
car	%100
person	%99

Tablo 5'de, Şekil 12, Tablo 3, Şekil 13 ve Tablo 4'deki ilk eğitilen ağırlıklar ile yeniden eğitilen ağırlıkların seçilen resim üzerindeki tahmin sonuçlarının değerlendirilmesi görülmektedir.

Tablo 5. Eğitim - yeniden eğitim tahmin sonuçlarının değerlendirilmesi

	Tespit süresi(ms)	Tahmin(%) Sınıf 1 (Car)	Tahmin(%) Sınıf 2(Person)
İlk Eğitim	602.3	34	0
Yeniden Eğitim	599.1	100	99

## Sonuçlar

Bu çalışmada, DÖ yöntemi ile İHA'dan alınan video görüntülerindeki araçların ve yayaaların tespiti amaçlanmıştır.

İHA olarak DJI Mavic 2 Zoom Drone kullanılmıştır. Mavic 2 Zoom 1/2.3-inch 12-megapixel sensör ve 4x zoom ile güçlendirilmiş Zoom Kamera, 2x optik zoom (24-48 mm) lense sahiptir [7].

DÖ algoritması olarak ESA mimarilerini kullanan ve nesne tanıma mimarileri arasında en fazla öne çıkan Darknet-53 framework'ü üzerinde çalışan YOLOv3 algoritması kullanılmıştır.

Veri seti olarak Mavic 2 Zoom dan alınan video görüntülerden 377 resim, VIRAT [1] tarafından üretilen video

datasetlerinden 500 resim, toplamda 877 resim kullanılmıştır.

Video'yu, frame'lere ayırmak için ffmpeg yazılımı kullanılmıştır. Resimlerin etiketleme işlemi LabelImg [14] aracı ile yapılmış olup, %90'i eğitim, %10'u doğrulama (test) seti olacak şekilde bir kod bloğu yardımı ile rassal olarak ayrılmıştır.

Algoritma Google Colab Tesla T4 GPU makinesi kullanılarak eğitilmiştir. Eğitim yaklaşık 16 saat sürmüş olup, ortalama loss değeri 2.345 ve mAP değeri %79 olacak şekilde bir başarımla elde edilmiştir.

DÖ Ağı, ilk eğitim setinde hiçbir şekilde benzeri olmayan Mavic 2 Zoom ile elde edilen resimler (Şekil 9, Şekil 11, Şekil 12) ile test edilmiştir. Bu test işleminde Şekil 9'deki spor aletlerinin bir kısmını yaya bir kısmını da araç olarak tespit etmesi, Şekil 11'deki ağaçları yaya olarak tespit etmesi ve Şekil 12'deki oturan ve araç içerisindeki yayaları tespit edememesinden dolayı ilk eğitim sonucunda oluşan ağırlık değerleri yeniden eğitim işlemine tabi tutulmuştur. Eğitim yaklaşık 5 saat sürmüş, loss değeri istendiği gibi azalır 1'e yaklaşmıştır. Bununla beraber mAP değerinin önceki değere kıyasla %70.9'a düşmesi istenmeyen ve beklenmeyen bir durumdur.

Tablo 1'de yukarıda bahsi geçen eğitim ve yeniden eğitim sonrası elde edilen sonuçlar detaylı bir şekilde görülmektedir. Bununla beraber Tablo 2'deki [3] referans nolu çalışma ile bu çalışmanın değerleri incelendiğinde, öğrenme katsayılarının iki çalışmada da aynı olduğu ancak küme boyutu (batch) ile kullanılan veri setlerinin farklı olduğu, bu çalışmadaki veri setinin sayıca daha az olduğu görülmektedir. Bununla beraber [3] referans nolu çalışmanın eğitildiği ortam ve loss değeri ile ilgili detaylar bilinmemektedir. Tablo 2'deki mAP değerleri incelendiğinde ise bu çalışmanın ilk eğitimi sonrası elde edilen mAP değerinin, [3] referans nolu çalışmanın mAP değeri ile aynı olduğu görülmektedir. Ancak ilk eğitim sonrası elde edilen ağırlıkların detaylı testi sonucu, yapılan bazı yanlış tespitler, araştırmacıyı yeniden eğitim işlemine itmiştir ancak elde edilen sonuç kısmen beklendiği gibi olmayıp mAP değeri %70.9'a gerilemiştir. Eğitim setinin az olması ve çalışma boyunca Google Colab'da yaşanan problem yani Google Colab'da başlatılan eğitimin tamamlanmadan sonlanması ise tespit edilen sebepler arasında gösterilebilir.

Şekil 9, Şekil 11, Şekil 12'deki yeniden eğitim sonrası oluşan ağırlıkların resim üzerindeki tahmin sonuçları ile Şekil 8 ve Şekil 10'deki ilk eğitim sonrası oluşan ağırlıkların resim üzerindeki tahmin sonuçları karşılaştırılmıştır. Sonuçlar ilk bakışta her ne kadar benzer gibi görünse de ilk eğitime kıyasla ikinci eğitimdeki tahmin oranları daha da artmıştır. Bu, ilk eğitim sonrası oluşan ağırlıklar ile yeniden eğitim sonrası oluşan ağırlıkların aynı resim üzerindeki test sonuçlarında, Şekil 12, Tablo 3, Şekil 13 ve Tablo 4'de çok daha net görülmektedir.

Şekil 12 ve Şekil 13'de testin resim üzerindeki sonucu, Tablo 3 ve Tablo 4'de ise testin süre ve yüzde bazında ağ çıktısı görülmektedir. Tablo 5'de, ilk eğitim sonrası oluşan ağırlıkların ve yeniden eğitim sonrası oluşan ağırlıkların tespit süreleri ile araç ve yaya bazında tespit oranları görülmektedir. Test için kullanılan resimdeki 2 aracın ve 2 yayanın tahmin ortalama başarımlarını yüzde olarak incelendiğinde araç ve yaya başarımlarının ilk eğitimde %34 ve %0 iken yeniden eğitimde %100 ve %99 ile istenen sonuca erişmiştir. Burada da ilk eğitim setine çok daha farklı resimlerin eklenip, eğitim setinin daha da zenginleştirilmesi ve yeniden eğitim yapılması Şekil

13 ve Tablo 4'deki tahmin başarımının Şekil 12 ve Tablo 3'deki başarımdan yüksek olmasının nedenlerinden biri olarak gösterilebilir.

Sonuç olarak, veri seti sayısının artırılması, konfigürasyon dosyasındaki parametrelerin daha iyi ayarlanması ve eğitim işleminin daha iyi bir makinede yapılması çok daha iyi mAP değeri ve loss değerine erişilmesine olanak tanır.

İleriki süreçte varılan sonuçlar üzerinden bahsi geçen çalışmanın iyileştirilmesi üzerinde ilerlenecek ve ek olarak tespitlerin gerçek zamanlı (real time) yapılması üzerine çalışılacaktır.

## **Etik kurul onayı ve çıkar çatışması beyanı**

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur.

Hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır

## **Yazar Katkıları**

Muhammet Ali ARSERİM ve Aysan ÇETİN

-Çalışma konsepti ve tasarım

-Veri toplama

-Verilerin analizi ve yorumlanması üzerinde çalışmışlardır.

## **Teşekkür**

Bu çalışma, DÜBAP MÜHENDİSLİK.19.007 nolu proje tarafından desteklenmiştir.

## **KAYNAKLAR**

- [1] The Virat Video Dataset, 11 Jan 2012. Erişim Tarihi: 18 Mayıs 2020. <https://viratdata.org/>
- [2] O. Bayraktar, F. Özdemir, Ö. Çetin ve G. Yılmaz, "İnsansız Hava Araçları İçin Otonom İniş Sistemi Simülasyonu Tasarımı", Bilişim Teknolojileri Dergisi, c: 5, sayı: 2, ss: 1-8, 2012.
- [3] B. Bender, M. E. Atasoy ve F. Semiz, "Deep Learning-Based Human and Vehicle Detection in Drone Videos", 6th International Conference on Computer Science and Engineering (UBMK), pp. 446-450, September 2021.
- [4] Ozkan, Z., Bayhan, E., Namdar, M., & Basgumus, A. "Object Detection and Recognition of Unmanned Aerial Vehicles Using Raspberry Pi Platform", In 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp: 467-472), 2021.
- [5] T. Tang, Z. Deng, S. Zhou, L. Lei ve H. Zou, "Fast vehicle detection in UAV images", 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP), pp. 1-5, May 2017.
- [6] J. Lee, J. Wang, D. Crandall, S. Šabanović ve G. Fox, "Real-time, cloud-based object detection for unmanned aerial vehicles", 2017 First IEEE International Conference on Robotic Computing (IRC), (pp. 36-43), April 2017.
- [7] MAVIC 2 Specs - Mavic 2 Zoom, DJI, Erişim Tarihi: 12 Ocak 2022. [https://www.dji.com/mavic-2?site=Brandsite&from=insite\\_search](https://www.dji.com/mavic-2?site=Brandsite&from=insite_search)
- [8] H. V. Karakuş, Darknet YOLOv3 hızlı bakış, 2020. Erişim tarihi: 23 Kasım 2021. <https://medium.com/@karakus.haciveli/darknet-yolov3-hizli-bakis-ddc9cd5582ea>

[9] S. Dulepet, P. Maji, M. Harsh and K. Washabaugh, Deploying a Scalable Object Detection Inference Pipeline Part, 2020. Eriřim Tarihi:21 Aralık 2020. <https://developer.nvidia.com/blog/deploying-a-scalable-object-detection-inference-pipeline/>

[10] Y. Mesci, YOLO Algoritmasını Anlamak, 2019. Eriřim tarihi: 7 Aralık 2021. <https://medium.com/deep-learning-turkiye/yolo-algoritmasını-anlamak-290f2152808f>

[11] YOLO: Real-Time Object Detection, Eriřim tarihi: 7 Aralık 2021. <https://pjreddie.com/darknet/yolo/>

[12] The meaning of output parameters during YOLO v3 training. Eriřim Tarihi: 31 Ağustos 2022. <https://blog.actorsfit.com/a?ID=00850-083895a9-5fda-4738-939b-1149afe66ca1>

[13] A. Christiansen, Anchor Boxes - The key to quality object detection, 2018. Eriřim tarihi: 7 Aralık 2021. <https://towardsdatascience.com/anchor-boxes-the-key-to-quality-object-detection-ddf9d612d4f9>

[14] <https://github.com/tzutalin/labelImg>