



POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)

URL: <http://dergipark.org.tr/politeknik>



Akıllı mobil cihazlarda YOLOv7 modeli ile nesne tespiti

Object detection with YOLOv7 model on smart mobile devices

Yazar(lar) (Author(s)): Batuhan KARADAĞ¹, Ali ARI²

ORCID1: 0000-0002-4661-6607

ORCID2: 0000-0002-5071-6790

To cite to this article: Karadağ B. and Arı A., “Object detection with YOLOv7 model on smart mobile devices”, *Journal of Polytechnic*, 26(3): 1207-1214, (2023).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Karadağ B. Ve Arı A., “Akıllı mobil cihazlarda YOLOv7 modeli ile nesne tespiti”, *Politeknik Dergisi*, 26(3): 1207-1214, (2023).

Erişim linki (To link to this article): <http://dergipark.org.tr/politeknik/archive>

DOI: 10.2339/politeknik.1296541

Akıllı Mobil Cihazlarda YOLOv7 Modeli ile Nesne Tespiti

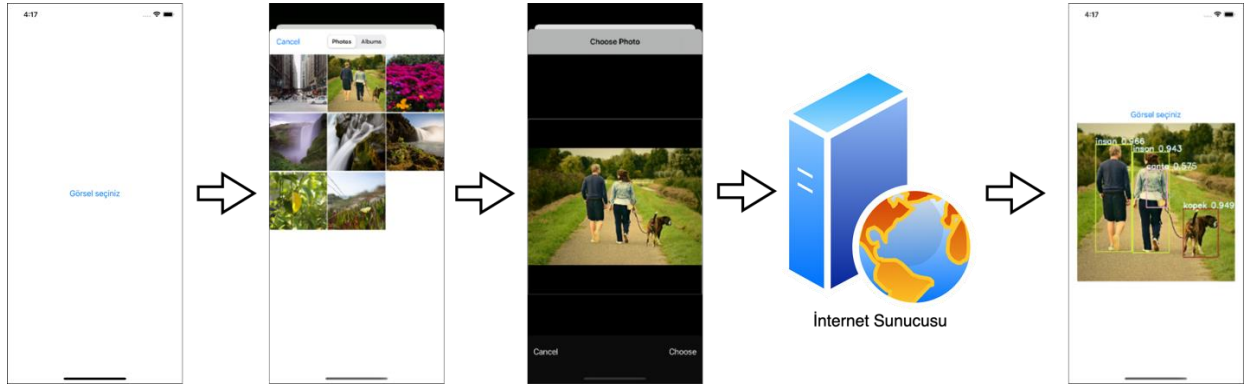
Object Detection with YOLOv7 Model on Smart Mobile Devices

Önemli noktalar (Highlights)

- ❖ Derin öğrenme algoritmaları yüksek donanımlı cihazlarda çalışması beklenirken, bu çalışmada YOLOv7 nesne tespit algoritmasının akıllı mobil cihazlar üzerinde çalıştırılması sağlanmıştır. / While deep learning algorithms are expected to work on highly equipped devices, in this study, YOLOv7 object detection algorithm is provided to run on smart mobile devices.
- ❖ YOLOv7 nesne tespit yönteminin literatüre katkısı değerlendirilmiştir. / The contribution of the YOLOv7 object detection method to the literature has been evaluated.

Grafik Özet (Graphical Abstract)

YOLOv7 nesne tespit yöntemi kullanılarak mobil cihazlarda gerçek zamanlı çekilen bir görüntünün veya halihazırda galeride bulunan herhangi bir görüntünün sunucuya aktarıldıktan sonra görüntü içerisinde bulunan nesne veya nesnelerin tespit edilmesi sağlanmıştır. / By using the YOLOv7 object detection method, after an image taken in real time on mobile devices or any image currently in the gallery is transferred to the server, the object or objects in the image are detected.



Şekil. Uygulamanın çalışma prensibi/ Figure. Working principle of the application

Amaç (Aim)

YOLOv7 nesne tespit aracının mobil cihazlarda kullanılabilir hale getirilmesi hedeflenmiştir. / It is aimed to make the YOLOv7 object detection tool available on mobile devices.

Tasarım ve Yöntem (Design & Methodology)

Önceden eğitilmiş modelin, mobil cihazlar üzerinde çalıştırılması için bir takım ön işlemlerin gerçekleştirilmesinin ardından arka-uç kodlanarak sunucu ile nesne tespit işlemi gerçekleştirilmiştir. / After performing some pre-processes to run the pre-trained model on mobile devices, the back-end was coded and object detection was performed with the server.

Özgünlük (Originality)

Literatürdeki çalışmalara bakıldığında mobil cihazlarda React Native platformunda YOLOv7 nesne tespit yönteminin daha önce kullanılmadığı görülmüştür. Bu da yapılan bu çalışmayı özgün kılmıştır. / When we look at the studies in the literature, it has been seen that the YOLOv7 object detection method has not been used before on the React Native platform on mobile devices. This made this study original.

Sonuç (Conclusion)

Yapılan çalışma sonucunda YOLOv7 nesne tespit aracı ile akıllı mobil cihazlarda hızlı ve doğru sonuç alınmıştır. / As a result of the study, fast and accurate results were obtained on smart mobile devices with the YOLOv7 object detection tool.

Etik Standartların Beyanı (Declaration of Ethical Standards)

Bu makalenin yazar(lar)ı çalışmalarında kullandıkları materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel bir izin gerektirmediğini beyan ederler. / The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Akıllı Mobil Cihazlarda YOLOv7 Modeli ile Nesne Tespiti

Araştırma Makalesi / Research Article

Batuhan KARADAĞ^{1,2}, Ali ARİ³

¹İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Malatya / Türkiye

²İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Hatay / Türkiye

³İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya / Türkiye

(Geliş/Received : 12.05.2023 ; Kabul/Accepted : 29.05.2023 ; Erken Görünüm/Early View : 06.06.2023)

ÖZ

Derin öğrenmeye dayalı güncel nesne tespit algoritmalarından biri olan YOLOv7 modelinin Microsoft COCO verisetinde aldığı %51.2'lik ortalama kesinlik başarısı, diğer nesne tespit yöntemlerinin ilerisinde olduğunu kanıtlamıştır. YOLO ilk sunulduğu dönemden itibaren, hız ve doğruluk açısından etkili olması sebebiyle ticari alandaki nesne tespit problemlerinde tercih edilen bir model olmuştur. Genellikle derin öğrenmeye dayalı sistemlerin çalıştırılabilmesi için yüksek kapasitede donanımlara ihtiyaç duyulmaktadır. Bu çalışmada, günümüzde ticaretin önemli araçlarından biri haline gelen akıllı mobil cihazlarda nesne tespiti yapılabilmesi için YOLOv7 modelinin sunucuda aktif edilmesi ile akıllı mobil cihazlarda grafik işlemci birimi kullanılmadan nesne tespiti yapılabilmesi amaçlanmıştır. Yapılan çalışma ile YOLOv7 nesne tespit algoritması, iOS işletim sistemine sahip mobil cihazlarda başarı ile çalıştırılmıştır. Bu sayede mobil cihazlarda çekilen bir görüntü veya halihazırda galeride bulunan herhangi bir görüntü sunucuya aktarıldıktan sonra, doğruluk ve hız açısından etkili bir şekilde görüntü içerisinde bulunan nesnelere tespitinin gerçekleştirilmesi sağlanmıştır.

Anahtar Kelimeler: YOLOv7, Nesne Tespiti, Mobil Nesne Tespiti, Mobil YOLOv7.

Object Detection with YOLOv7 Model on Smart Mobile Devices

ABSTRACT

The YOLOv7 model, which is one of the current object detection algorithms based on deep learning, achieved an average accuracy of 51.2% in the Microsoft COCO dataset, proving that it is ahead of other object detection methods. YOLO has been a preferred model for object detection problems in the commercial field since it was first introduced, due to its speed and accuracy. Generally, high-capacity hardware is needed to run deep learning-based systems. In this study, it is aimed to detect objects in smart mobile devices without using a graphic processor unit by activating the YOLOv7 model on the server in order to be able to detect objects in smart mobile devices, which have become one of the important tools of trade today. With the study, the YOLOv7 object detection algorithm has been successfully run on mobile devices with iOS operating system. In this way, an image taken on mobile devices or already in the gallery after any image is transferred to the server, it is ensured that the objects in the image are detected effectively in terms of accuracy and speed.

Keywords: YOLOv7, Object Detection, Mobile Object Detection, Mobile YOLOv7.

1. GİRİŞ (INTRODUCTION)

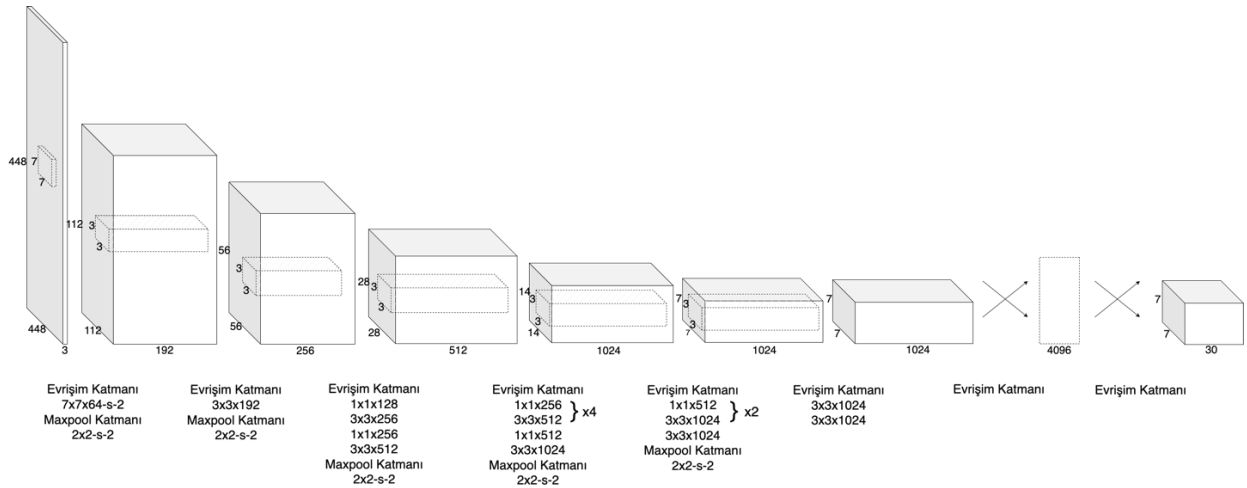
Son yıllarda bilgisayarlı görünün popülerleşmesiyle birlikte nesne tespit algoritmalarının gelişimi hız kazanmıştır. Bilgisayarlı görü ve makine öğrenmesi algoritmalarının gerçek hayatta karşılaşılan problemlerde kullanımının artması ve derin sinir ağlarının, uygulanabilirlik ve verimlilik bakımından avantajlı olabilmesi sebebiyle bu alana ilgiler artmıştır [1, 2]. Nesne tespit algoritmaları özünde geleneksel yöntemler ve derin öğrenme yöntemleri olarak isimlendirilerek ikiye ayrılır. Nesne tespit problemlerinde kullanılan derin öğrenmeye dayalı nesne tespit yöntemleri hız ve doğruluk bakımından geleneksel yöntemlerden daha iyi sonuç vermektedir [3]. Literatürde derin öğrenme

yöntemlerine dayalı nesne tespit algoritmaları incelendiğinde farklı yaklaşımlar olduğu görülmektedir [4]. Temelde nesne tespit algoritmaları kendi aralarında ikiye ayrılır [5]. Bunlardan ilki, sınıflandırma algoritmaları olarak bilinen R-CNN (Bölgesel Tabanlı Evrişimsel Sinir Ağı (Region Based Convolutional Neural Network)) [6], Fast R-CNN (Hızlı - Bölgesel Tabanlı Evrişimsel Sinir Ağı (Fast - Region Based Convolutional Neural Network)) [7] ve Faster R-CNN (Daha Hızlı Bölgesel Tabanlı Evrişimsel Sinir Ağı (Faster - Region Based Convolutional Neural Network)) [8] gibi bölge önerisine bağlı olarak çalışan nesne tespit algoritmaları, ikincisi ise, regresyon algoritmaları olarak bilinen SSD (Tek Atış Dedektörü (Single Shot Detector)) [9] ve YOLO (Yalnızca Bir Kere Bak (You Only Look Once)) [10 - 12] vb. algoritmalarıdır. Yöntem olarak farklı olsalar da her iki türden algoritmalar nesne tespitinde oldukça başarılı

*Sorumlu Yazar (Corresponding Author)
e-posta : batuhan.karadag@iste.edu.tr

sonuçlar vermektedir [5]. Sınıflandırma algoritmaları tamamı ile bölgelere dayalı olduğundan ötürü, girdi görüntüleri için öncelikle bölge önerileri üretilmektedir. Önerilen bölgelerde ağ, hangi konumun nesne içerdiğini önceden kontrol etmesi sebebiyle bu yöntem daha doğru sonuç vermektedir [13]. Lakin bununla birlikte daha fazla işlem maliyetine sebep olmaktadır. Bununla birlikte regresyon algoritmalarında, bu süreç daha hızlı ilerlemektedir. Regresyona dayalı nesne tespit algoritmaları, öncelikle girdi görüntüsü içerisinde nesne olup olmadığını, daha sonrasında bu nesnenin ne olduğunu tahmin etmektedir. Regresyona dayalı nesne tespit algoritmalarından biri olan YOLO günümüzdeki nesne tespit problemlerinde yaygın olarak kullanılan bir derin öğrenme algoritmasıdır [14]. YOLO algoritması ile nesne tespit problemlerinde doğru ve hızlı bir şekilde sonuç almak mümkün kılınmıştır. 2015 yılında ilk kez Redmon ve arkadaşları tarafından karşımıza çıkartılan YOLO algoritması [10] günümüze kadar birçok nesne tespit uygulamalarında kullanılmıştır. Bilim insanları tarafından her geçen sene YOLO algoritması geliştirilerek daha az maliyetle daha doğru sonuç verecek özelliğe eriştirilmiş ve geliştirilmeye devam edilmektedir. Wang ve arkadaşlarının önermiş olduğu YOLOv7 [15] nesne tespit algoritması %51.2'lik başarıyı sayesinde literatürde bulunan diğer nesne tespit algoritmalarından daha başarılı olduğunu ispatlamıştır.

Bu çalışmada, YOLOv7 nesne tespit algoritmasının akıllı mobil cihazlara uygulanmasıyla, mobil cihazlardan seçilen bir görüntünün içerisinde bulunan nesnelerin kısaca bir sürede tespit edildiği belirtilmiştir.



Şekil 1. YOLO mimarisi. Modelde 24 adet konvolüsyon katmanı 2 adet tam bağlantı katmanı bulunmaktadır (Architecture of YOLO. The model has 24 convolution layers and 2 fully connected layers.) [10]

3. YOLOv7

YOLOv7, resmi olarak yayımlanmamış bir YOLO sürümü olmasına rağmen, temel prensipte aynı şekilde görev yapan bir nesne tespit algoritmasıdır. YOLO'da girdi görüntüsü, bir dizi konvolüsyon (Evrişim) katmanı ve özellik çıkarma katmanı ile işlendikten sonra sonuç olarak görüntü içerisinde bulunan nesnelere belirlemekte

2. YALNIZCA BİR KERE BAK (YOU ONLY LOOK ONCE)

YOLO algoritması, nesne tespit problemlerinde sonuca hızla ulaşabilmesi sebebiyle günümüze kadar dikkat çekmiş ve birçok çalışmada tercih edilmiş bir nesne tespit yöntemidir. Geçmişten günümüze geleneksel ve modern derin öğrenme algoritmaları incelendiğinde, çok katmanlı ağ yapısı sayesinde CNN (Evrişimsel Sinir Ağı (Convolutional Neural Network)) tabanlı mimarilerin farklı alanlardaki bilimsel çalışmalarda kullanıldığı gözlemlenmiştir [16, 17]. CNN modeli, verilen görüntü üzerindeki nitelik çıkarımında ortaya koyduğu başarı ile nesne tespit problemlerinde tercih edilen bir ağ modelidir [18].

YOLO'da girdi görüntüsü, sadece bir kez evrişimsel sinir ağından geçirilir ve görüntü üzerinde sınırlayıcı kutular oluşturulur. Sınırlayıcı kutuların içinde bulunan nesnelerin öncelikle nesne olup olmadığı, ardından bu nesnenin hangi sınıf etiketine sahip olduğu kontrol edilir ve nesne tespiti tamamlanmış olur. YOLO bu işlemi yaparken, her sınırlayıcı kutunun içinde güven skoru hesaplar ve bu hesap sonucunda nesne tespiti gerçekleştirilir. Özünde Redmon ve arkadaşları [10], nesne tespit problemini bir regresyon problemine indirgemiş ve bu sayede yalnızca bir kez görüntünün evrişimsel sinir ağında işlenmesi sonucunda daha az maliyetle sonuca varılabileceği gösterilmiştir.

YOLO mimarisi Şekil 1.'de gösterilmiştir.

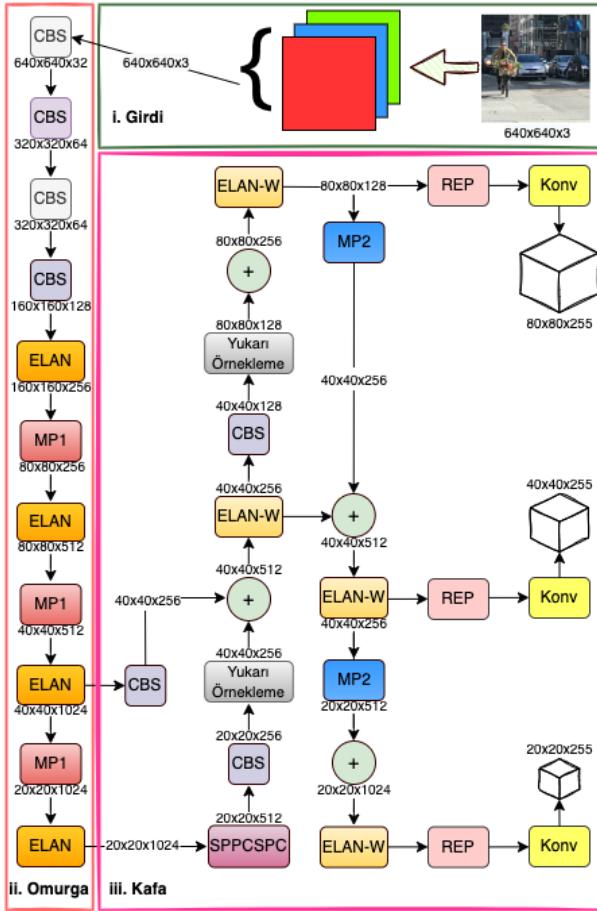
ve onları sınıflamaktadır.

YOLOv7 temelde üç ana bileşenden oluşur. Bunlar;

1. Girdi (Input),
2. Omurga (Backbone),
3. Kafa (Head)

olarak adlandırılır.

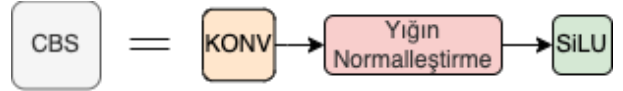
Girdi bileşeni, modele aktarılan görüntü veya videolardan oluşur. Omurga bileşeni, girdi görüntüsünden özellik çıkarımında kullanılan önceden eğitilmiş bir ağıdır. Omurga katmanı birkaç konvolüsyon katmanından, E-ELAN (Genişletilmiş Verimli Katman Toplama Ağları) katmanından ve dönüşümlü olarak en-boy oranını yarıya indiren, kanalları ikiye katlayan ve özellikleri çıkaran MP (Maksimum Havuzlama (Maximum Pooling)) katmanlarından oluşur. Kafa bileşeni, son aşamadaki işlemleri gerçekleştirmek için kullanılır. Katman özellik haritalarına sınırlayıcı kutuları uygular ve nihai çıktıyı oluşturur. Bu nihai çıktılar: nesne etiketleri, sınırlayıcı kutu çizimleri, nesne tespit skorları olabilir. YOLOv7 mimarisi Şekil 2.'de sunulmuştur.



Şekil 2. YOLOv7 Mimarisi (Architecture of YOLOv7) [19]

YOLOv7 algoritmasında, genellikle girdi görüntüsü 640x640 biçiminde yeniden boyutlandırılır, ardından bu görüntü omurga ağına aktarılır.

CBS (Konvolüsyon - Yığın Normalleştirme (Batch Normalization) - SiLU (Sigmoid-Weighted Linear Unit)), ağın farklı katmanlarından özellik haritalarını birleştiren ve daha ayrıntılı özellik haritaları oluşturmayı sağlayan yapıdır. CBS sayesinde daha başarılı nesne tespit sonucu alınması sağlanılmıştır. CBS ağının yapısı Şekil 3.'de gösterilmiştir.

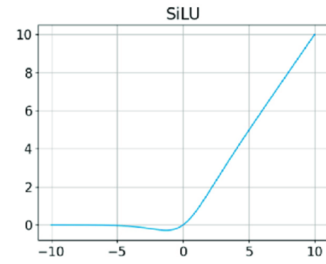


Şekil 3. YOLOv7 mimarisinde yer alan CBS ağının mimarisi (Architecture of CBS network in YOLOv7 architecture) [20]

Omurga bölümü genelde ELAN (Verimli Katman Toplama Ağları), MP yapılarını ve SiLU aktivasyon fonksiyonunu kullanır. SiLU aktivasyon fonksiyonu Eşitlik 1 ile gösterilmiştir.

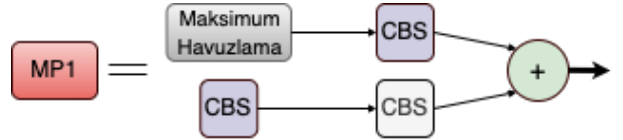
$$f(x) = \frac{x}{1 + e^{-x}} \quad (1)$$

SiLU'nun grafiği Şekil 4.'de gösterilmiştir.



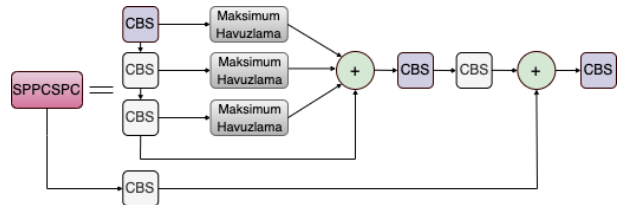
Şekil 4. SiLU Aktivasyon Fonksiyonu Grafiği (SiLU Activation Function Chart)

Maksimum havuzlama ağının mimarisi Şekil 5.'de gösterilmiştir.



Şekil 5. YOLOv7 mimarisinde yer alan MP1 ağının mimarisi (Architecture of MP1 network in YOLOv7 architecture) [20]

SPPCSPC ağı, SPP (Uzamsal Piramit Havuzlama (Spatial Pyramid Pooling)) blokları CSPNet'tir (Aşamalar Arası Kısmi Ağ (Cross Stage Partial Network)). CSPNet, yinelenen gradyan sorununu azaltmak için tasarlanan bir ağ modelidir [20]. Kafa'ya ait blokta yer alan SPPCSPC ağının mimarisi Şekil 6.'da gösterilmiştir.

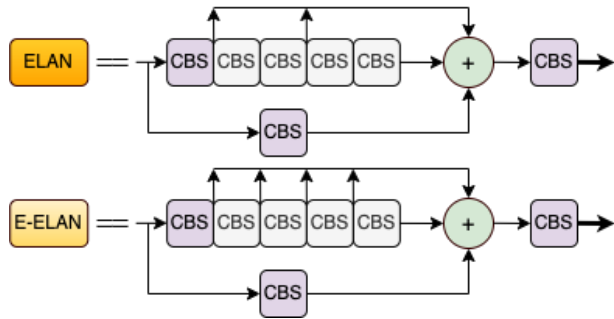


Şekil 6. YOLOv7 mimarisinde yer alan SPPCSPC ağının mimarisi (Architecture of SPPCSPC network in YOLOv7 architecture) [20]

3.1. Genişletilmiş Verimli Katman Toplama Ağları (Extended Efficient Layer Aggregation Networks)

Modern nesne tespit yöntemleri tasarlanırken, asıl önem arz eden terimler parametre sayısının, hesaplama miktarının ve yoğunluğunun optimize edilmesi olarak düşülür [15]. YOLOv7, mimarisinde E-ELAN olarak adlandırılan derin ağların etkili bir şekilde kullanımını sağlayan ve kendinden önceki modellerle kıyaslandığında yenilikle güçlendirilmiş bir ağ modelidir.

YOLOv4 ve YOLOR modellerinin üzerine geliştirilerek oluşturulan YOLOv7 modelinde Daha önceki modellerde karşılaşılan ELAN (Verimli Katman Toplama Ağı) daha az parametre kullanımına yol açtığından dolayı daha hızlı ve doğru sonuç alınabilmesi adına bu ağ genişletilmek istenmiştir [22, 23]. ELAN ve E-ELAN mimarileri Şekil 7.'de sunulmuştur.



Şekil 7. ELAN ve E-ELAN mimarileri (ELAN and E-ELAN architectures) [15, 20]

E-ELAN'da her CBS ağının ardından birleştirme (concat) işlemi gerçekleştirilmektedir. Bu sayede daha çok parametre kullanımı sağlanmaktadır.

3.2. Model Ölçeklendirme (Model Scaling)

Model ölçeklendirme, modele girdi olarak verilen görüntünün boyutu (görüntünün çözünürlüğü), modelin katman sayısı (derinlik), kanal sayısı (genişlik), öznetelik sayısı gibi farklı terimleri optimize etmek için kullanılan yöntemdir [15, 22]. Wang ve arkadaşları tarafından, DenseNet [24] veya VoVNet [25] gibi modellerin derinliğini ölçeklendirdiklerinde, bazı katmanların girdi genişliğini değiştireceğini gözlemlemişlerdir [15].

4. MATERYAL VE METOD (MATERIAL and METHOD)

4.1. YOLOv7 Eğitim Parametreleri (YOLOv7 Training Parameters)

Wang ve arkadaşlarının [15] önerdikleri YOLOv7 modelinde, görüntü 640x640 piksel olarak yeniden boyutlandırılmıştır. Önerilen YOLOv7 modeli için belirlenen eğitim parametreleri aşağıdaki gibidir:

- Yığın boyutu (Batch Size): 16,
- Devir (Epoch): 100,
- Eğitim oranı: 0,0001,

- Aktivasyon fonksiyonu: SiLU,
- Kayıp fonksiyonu: Pürüzsüz L1 Kaybı (Smooth L1 Loss)

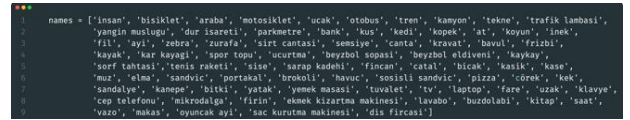
Wang ve arkadaşları [15], belirlenen bu parametreler ile MS COCO veri seti üzerinde gerçekleştirilen eğitimin en iyi performansı sağladığını belirlemişlerdir.

4.2. React Native (RN)

React Native (RN), Meta tarafından 2015'te geliştirilen Android ve iOS işletim sistemlerinde çalıştırılabilen çapraz yazılımlar üretmeye yarayan ve Javascript betiğini kullanan açık kaynaklı bir yazılım yapısıdır. Günümüzde başta Facebook olmak üzere, Microsoft, Uber, Skype, Shopify vb. gibi ün kazanmış şirketler tarafından tercih edilmiştir. Mobil uygulama geliştiricileri tarafından popüler olarak kullanılan bu yapı, açık kaynaklı olması sebebiyle ve yazılımcı toplulukları sayesinde gelişmiş kütüphaneye sahiptir.

4.3. Veri Seti (Dataset)

Wang ve arkadaşları [15] tarafından ortaya çıkarılan YOLOv7 modelinde, modelin eğitimi için Microsoft (MS) tarafından 2017'de oluşturulan COCO (Bağlamda Yaygın Nesnelere (Common Objects In Context)) veriseti kullanılmış, doğrulama ve hiperparametre seçimleri için "val2017" veriseti kullanılmıştır. Bu verisetinde 80 adet nesne etiketi bulunmaktadır. Verisetinde bulunan etiketler Şekil 8.'de gösterilmiştir.



Şekil 8. MS COCO veritesinde bulunan etiketler (Labels in MS COCO dataset)

4.4. Modelin TensorFlow Lite Formatına Dönüşümü (Conversion of Model to TensorFlow Lite Format)

YOLOv7 PyTorch kütüphanesi ile geliştirilmiştir. Eğitilen PyTorch [26] modelin mobil cihazlarda kullanılabilmesi için, model üzerinde birkaç düzenleme yapılması gerekmektedir. Bunun sebebi, YOLOv7'yi test etmek için kullanılan ortam olan RN'in TensorFlow [27] (TF) kütüphalesinden biri olan TensorFlow Lite'ı (TFLite) desteklemesidir. PyTorch modelin TFLite modele dönüşüm haritası Şekil 9.'da gösterilmiştir.



Şekil 9. Model dönüşüm haritası (Model transformation map) [28]

Eğitilen PyTorch modelinin, Tensorflow Lite modeline dönüştürülmesi için gerekli adımlar:

i. PyTorch modelinin ONNX modeline dönüştürülmesi (Converting PyTorch model to ONNX model)

ONNX (Açık Sinir Ağı Değişimi (Open Neural Network Exchange)), makine öğrenmesi modellerinin eğitildiği bir ortamdan farklı bir platformda modelin kullanılması için ihtiyaç duyulan açık kaynaklı bir araç yöntemidir. ONNX modelleri, Tensorflow, PyTorch, Caffe2, MXNet, MATLAB ve daha birçok derin öğrenme çerçevesi tarafından desteklenmektedir.

PyTorch model ONNX modele çevrildiğinde, bu modelin taşınabilirliği artar ve farklı ortamlarda daha hızlı ve daha az maliyetle çalışabilen bir model haline gelmiş olur.

ii. ONNX modelinin, TensorFlow modele dönüştürülmesi (Converting ONNX model to TensorFlow model)

TFLite model dönüşümü sağlanması için modelin önce TF modele dönüştürülmesi gerekmektedir.

iii. TensorFlow modelinin, TensorFlow Lite modele dönüştürülmesi (Conversion of TensorFlow model to TensorFlow Lite model)

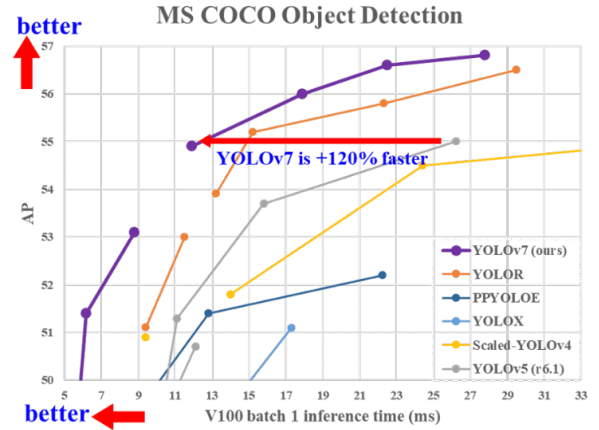
TensorFlow Lite model, mobil ve IoT (Nesnelerin İnterneti (Internet of Things)) cihazlarında derin öğrenme modellerinin çalıştırılabilmesi için tercih edilen düşük kaynak tüketen bir derin öğrenme kütüphanesidir. TensorFlow Lite ile model, mobil cihazlar gibi düşük güç seviyesine sahip cihazlar için ekstra optimize edilerek modelin daha hızlı sonuç vermesi ve daha az güç tüketmesini sağlamaktadır. TensorFlow Lite ile, eğitilen model iOS ve Android gibi cihazlarda kullanılabilir hale getirilmektedir.

4.5. TFLite Modelin Mobil Cihaza Entegrasyonu (Integration of TFLite Model to Mobile Device)

RN'de modelin çalıştırılması için arka uç (Back-End) yapıya ihtiyaç duyulmuştur. Bu sebeple, Python'da Flask kütüphanesi ile sunucu oluşturulmuştur. Hazırlanan bu sunucuda, temelde iki işlem gerçekleştirilecektir. Bunlardan ilki, girdi görüntülerini sunucuya aktarmak için kullanılan POST işlemi, ikincisi ise sunucuya aktarılan görüntülerden nesne tespit işlemi yapıldıktan sonra sonuçları tekrar cihaza döndürmek ve görüntülemek için kullanılan GET işlemidir.

5. SONUÇLAR (RESULTS)

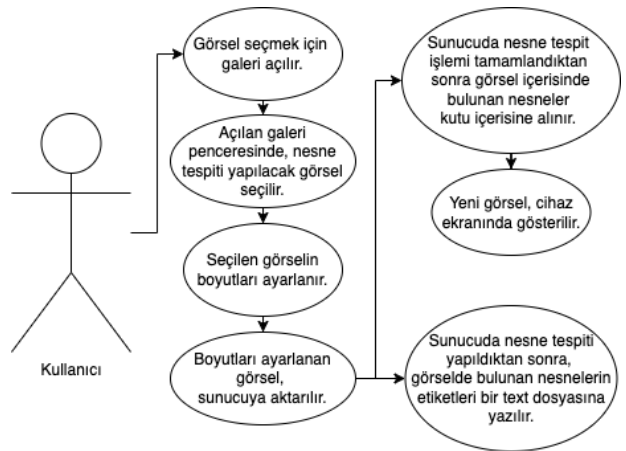
YOLOv7, 5-160 FPS (Bir saniyedeki görüntü sayısı) aralığında diğer tüm nesne tespit algoritmalarının üzerinde bir doğruluk vermektedir [15]. Diğer bazı nesne tespit algoritmaları ile YOLOv7'nin MS COCO verisetindeki başarımı Şekil 10.'da sunulmuştur.



Şekil 10. Microsoft COCO veriseti üzerinde eğitilen bazı nesne tespit algoritmalarının başarımları (Performance results of some object detection algorithms trained on Microsoft COCO dataset) [15]

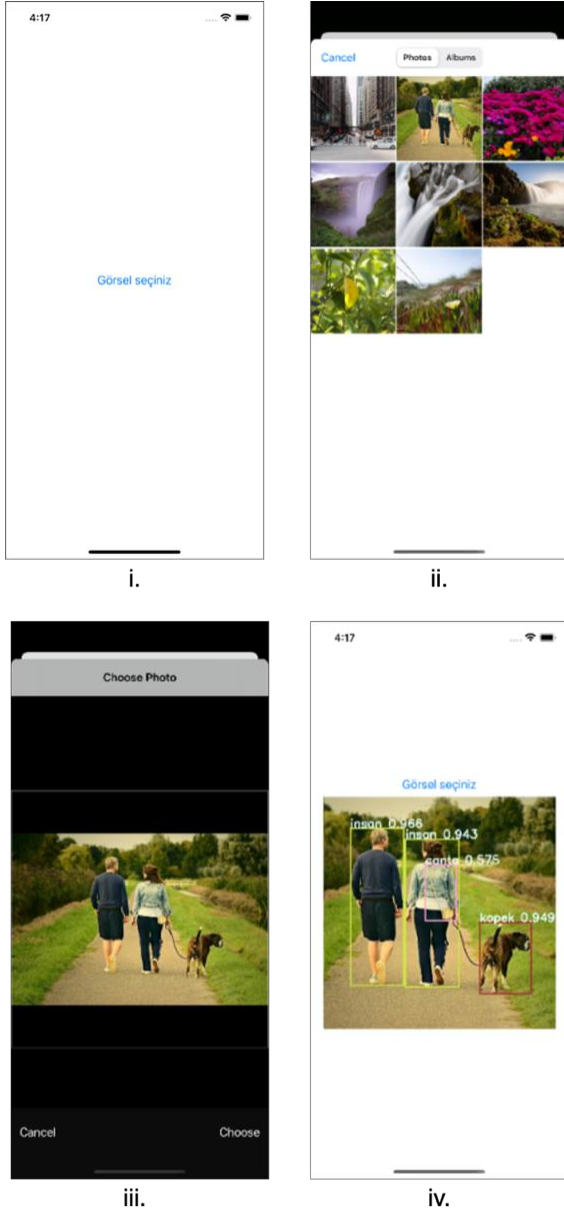
Önceden eğitilen YOLOv7 modeli ve Microsoft COCO veriseti, Google Colab platformunda Tesla T4 cihazı üzerinde eğitimi gerçekleştirilmiş ve model dışarı aktarılmıştır.

Bu çalışmanın, Kullanıcı Senaryosu Diyagramı (Use Case Diagram) Şekil 11.'de sunulmuştur.



Şekil 11. Geliştirilen uygulamanın kullanıcı senaryosu diyagramı (User case diagram of the developed application)

Şekil 12.'de uygulamanın çalışma şeklini anlatan ekran görüntüleri sunulmuştur.



Şekil 12. Çalışmada geliştirilen uygulamanın ekran görüntüleri (Screenshots of the application developed in this study)

Testler iOS işletim sisteminde çalıştırılan iPhone 14 cihazında gerçekleştirilmiştir.

- Uygulamanın anasayfası gösterilmektedir. Bu sayfada kullanıcıdan görsel seçilmesi beklenmektedir.
- Görsel seçilmesi için kullanıcının galeri sayfası açılmaktadır. Kullanıcın nesne tespiti yapmak istediği görseli seçmesi beklenir.
- Görselin tamamında nesne tespiti yapılacaksa, kırılmadan devam edilir. Belirli bir bölümünde nesne tespiti yapılacaksa, o alan seçilip kırılır.
- Nesne tespit işlemi tamamlanır. Görselde bulunan nesnelere kutu içine alınır ve üstlerine doğrulukları yazılır.

Literatür incelendiğinde YOLOv7, diğer nesne tespit algoritmalarından daha başarılı olduğu görülmüştür. Bu karşılaştırmayı yaparken, MS COCO test ve doğrulama verisetleri kullanılmıştır.



- Test veriseti: Modelin gelecekteki performansını değerlendirmede kullanılır.
- Doğrulama veriseti: Eğitim aşamasındaki modelin performansını değerlendirmede kullanılır.

Diğer bazı nesne tespit yöntemleri, aynı MS COCO test veriseti üzerinde gerçekleştirildiğinde edinilmiş olan FPS, test ve doğrulama verisetindeki ortalama hassasiyetleri (AP^{test} ve AP^{val}) skorlarının verileri Çizelge 1.'de sunulmuştur.

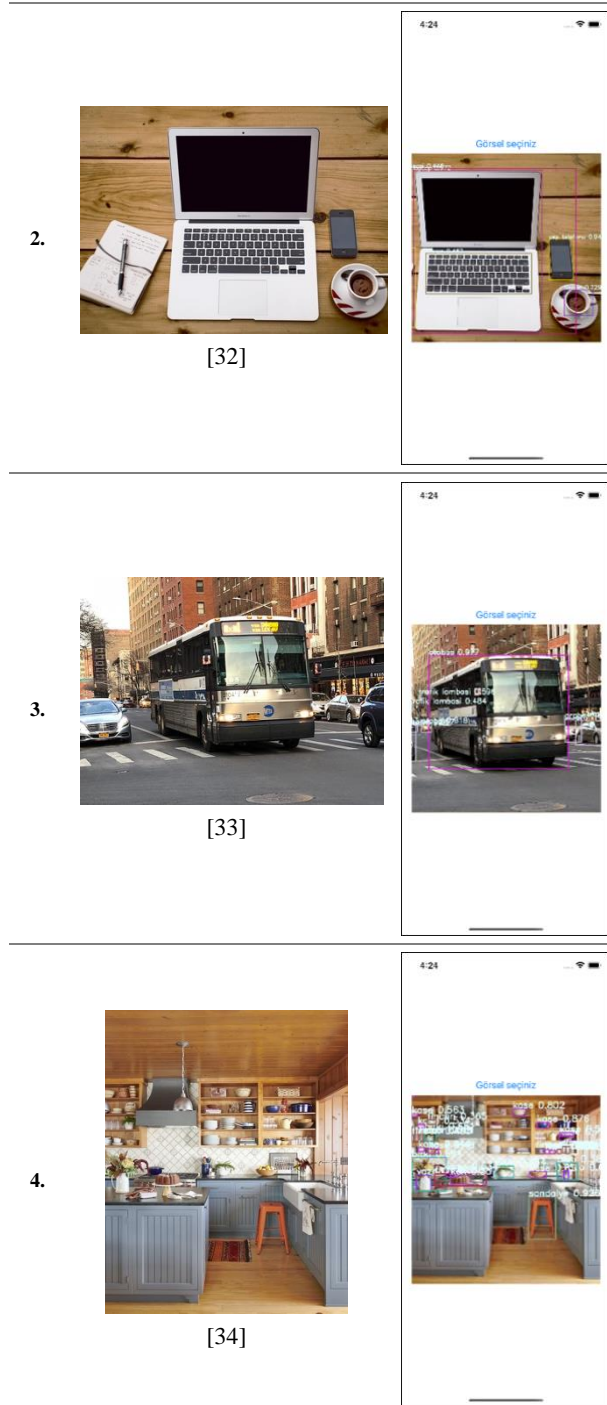
Model	Parametre Sayısı (Milyon)	FPS	AP^{test} (%)	AP^{val} (%)
YOLOv7-Tiny	6.2	286	38.7	38.7
YOLOv7	36.9	161	51.4	51.2
YOLOv7-X	71.3	114	53.1	52.9
YOLOv7-W6	70.04	84	54.9	54.6
YOLOv7-E6	97.2	56	56	55.9
YOLOv7-D6	154.7	44	56.6	56.3
YOLOv7-E6E	151.7	36	56.8	56.8

Çizelge 1. Diğer nesne tespit algoritmalarının test verisinde aldığı ortalama hassasiyet skorları (Average precision scores of other object detection algorithms in test data) [15, 29, 30]

Bu çalışmada ortaya çıkarılan uygulama ile Şekil 12.'de i., ii., iii. ve iv. adımlar sırasıyla yürütülerek elde edilen birkaç örnek Çizelge 2.'de sunulmuştur.

No.	Mobil Uygulamadan Sunucuya Gönderilen Görüntü	Sunucudan Mobil Uygulamaya Gönderilen Görüntü
1.		

[31]



Çizelge 2. Çalışmada geliştirilen mobil uygulama ile görüntülerden nesne tespiti örnekleri (Examples of object detection from images with the mobile application developed in this study)

6. DEĞERLENDİRME (EVALUATION)

Geçmişten günümüze kadar YOLO nesne tespit algoritmaları hakkındaki literatür incelendiğinde, YOLO hakkındaki gelişmelerin kısa zamanda ve hızlı bir şekilde geliştiği görülmektedir. Bu gelişmeler umut vaat etmekle birlikte birçok araştırma konusuna yardımcı olmuştur. YOLO'nun doğruluk hassasiyetinin yüksek olması ve nesne tespit sonuçlarını hızlı bir şekilde

bulabilmesi sebebiyle ticari alanda da tercih edilir bir model olmayı başarmıştır.

Bu çalışmada, YOLO ailesinin yeni üyelerinden biri olan (resmi olmayan) YOLOv7'nin akıllı mobil cihazlarda uygulanabilirliği gösterilmiştir. Bu modelin akıllı mobil cihazlara uygulanabilir olması, YOLOv7'nin ticari kullanımına pozitif katkı sağlayacağı düşünülmüştür. YOLOv7, doğruluk ve hız açısından diğer rakiplerine göre daha ileri bir nesne tespit aracıdır. Fakat YOLOv7;

- Küçük boyutlu görseller içerisinde nesne tespiti,
- Görsel içerisinde bulunan küçük boyutlu görseller,
- Aydınlık-Karanlık gibi çevre koşulları.

gibi durumlarda nesne tespiti konusunda zayıf kalabilmektedir. Buna rağmen, literatüre bakıldığında 160 FPS'e kadar sonuç verebilmesi sebebiyle diğer nesne tespit yöntemlerine göre video içerisinde nesne tespitinde öne çıkmayı başarmıştır.

ETİK STANDARTLARIN BEYANI (DECLARATION OF ETHICAL STANDARDS)

Bu makalenin yazar(lar)ı çalışmalarında kullandıkları materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel bir izin gerektirmediğini beyan ederler.

YAZARLARIN KATKILARI (AUTHOR'S CONTRIBUTIONS)

Batuhan KARADAĞ: Model eğitiminin ve mobil uygulamanın geliştirilmesinde katkıda bulunmuştur.

Ali ARI: Sonuçları değerlendirme ve makale yazım işleminde katkıda bulunmuştur.

ÇIKAR ÇATIŞMASI (CONFLICT OF INTEREST)
Makalenin yazar/yazarların, çalışma kapsamında herhangi bir kişisel ve finansal çıkar çatışması bulunmamaktadır.

KAYNAKLAR (REFERENCES)

- [1] Cai Y., Li H., Yuan G., Niu W., Li Y., Tang X., Ren B. Ve Wang Y., "YOLOmobile: Real-Time Object Detection on Mobile Devices via Compression-Compilation Co-Design", *arXiv:2009.05697*, (2020).
- [2] Arı A., "Multipath feature fusion for hyperspectral image classification based on hybrid 3D/2D CNN and squeeze-excitation network", *Earth Science Informatics*, 16:175–191, (2023).
- [3] Tao J., Wang H., Zhang X., Li X. and Yang H., "An object detection system based on YOLO in traffic scene", *2017 6th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China, 2017, pp. 315-319, (2017).
- [4] Dersuneli M. , Gündüz T. ve Kutlu Y. , "Bul-Tak Oyuncağı Şekillerinin Klasik Görüntü İşleme ve Derin Öğrenme Yöntemleri ile Tespiti", *Bitlis Eren*

- Üniversitesi Fen Bilimleri Dergisi*, c. 10, sayı. 4, ss. 1290-1303, Ara. 2021, (2021).
- [5] Liu C., Tao T., Liang J., Li K. and Chen Y., "Object Detection Based on YOLO Network," *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 2018, pp. 799-803, (2018).
- [6] Girshick R., Donahue J., Darrell T. and Malik J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 580-587, (2014).
- [7] Girshick R., "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448, (2015).
- [8] Ren S, He K, Girshick R ve Sun J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Trans Pattern Anal Mach Intell*, 2017 Jun;39(6):1137-1149, (2017).
- [9] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y. ve Berg A. C., "SSD: Single Shot MultiBox Detector", *arXiv:1512.02325*, (2015).
- [10] Redmon J., Divvala S., Girshick R. ve Farhadi A., "You Only Look Once: Unified, Real-Time Object Detection", *arXiv:1506.02640*, (2015).
- [11] Redmon J. ve Farhadi A., "YOLO9000: Better, Faster, Stronger", *arXiv:1612.08242*, (2015).
- [12] Redmon J. ve Farhadi A., "YOLOv3: An Incremental Improvement", *arXiv:1804.02767*, (2018).
- [13] Er, Ö. ve Bilge, H. Ş., "Bir Küçük Nesne Tespit Zorluğu Olarak Hava Görüntülerinden Araç Tespiti", *Veri Bilimi 4 (2021) : 73-83*, (2021).
- [14] Sultana F., Sufian A. ve Dutta P., "A Review of Object Detection Models Based on Convolutional Neural Network", *arXiv:1905.01614*, (2020).
- [15] Wang C., Bochkovskiy A. ve Liao H.M., "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", *arXiv:2207.02696*, (2022).
- [16] Donuk K., Arı A., Özdemir M.F. ve Hanbay D., "Deep Feature Selection for Facial Emotion Recognition Based on BPSO and SVM", *Politeknik Dergisi*, 26(1) : 131-142, (2023).
- [17] Karadağ B., Arı A. ve Karadağ M., "Derin Öğrenme Modellerinin Sinirsel Stil Aktarımı Performanslarının Karşılaştırılması", *Politeknik Dergisi*, 24(4) : 1611-1622, (2021).
- [18] Korkmaz Ş., Alkan M., "Derin öğrenme algoritmalarını kullanarak deepfake video tespiti", *Politeknik Dergisi*, *(*) : *, (*).
- [19] Wang Y., Wang H. ve Xin Z., "Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7", *in IEEE Access*, vol. 10, pp. 133936-133944, (2022).
- [20] Jiang, K., Xie, T., Yan, R., Wen, X., Li, D., Jiang, H., Jiang, N., Feng, L., Duan, X. ve Wang, J., "An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation", *Agriculture 2022*, 12(10), 1659, (2022).
- [21] Hossain J., "YOLOv7 explanation and implementation from scratch", <https://www.kaggle.com/code/jobayerhossain/yolov7-explanation-and-implementation-from-scratch> ,(2022), Erişim Tarihi: 16.04.2023.
- [22] Eği, Y., "YOLO V7 and Computer Vision-Based Mask-Wearing Warning System for Congested Public Areas", *Journal of the Institute of Science and Technology 13*, 22-32, (2023).
- [23] Hussain M., Al-Aqrabi H., Munawar M., Hill R., Alsoubi T, "Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections", *Sensors*, 22(18), 6927, (2022).
- [24] Huang G., Liu Z., Maaten L. ve Weinberger K, "Densely Connected Convolutional Networks", *arXiv:1608.06993*, (2016).
- [25] Lee Y., Hwang J., Lee S., Bae Y. ve Park J., "An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection", *arXiv:1904.09730*, (2019).
- [26] "PyTorch", <https://pytorch.org> , (2016), Erişim Tarihi: 02.02.2023.
- [27] "TensorFlow", <https://www.tensorflow.org/?hl=tr> , (2015), Erişim Tarihi: 08.03.2023.
- [28] Chau S.C., "Journey putting YOLO v7 model into TensorFlow Lite (Object Detection API) model running on Android", <https://medium.com/geekculture/journey-putting-yolo-v7-model-into-tensorflow-lite-object-detection-api-model-running-on-android-e3f746a02fc4>, (2022), Erişim Tarihi: 16.03.2023.
- [29] Kukil ve Rath S., "YOLOv7 Object Detection Paper Explanation & Inference", <https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/> , (2022), Erişim Tarihi: 20.03.2023.
- [30] Roboflow, "What is YOLOv7?", <https://roboflow.com/model/yolov7>, (2022), Erişim Tarihi: 20.03.2023.
- [31] Görsel 1, [https://www.travelandleisure.com/thmb/Cp3v7EPYiVhuAwOO8yxYv4qUjY0=/750x0/filters:no_upscale\(\):max_bytes\(15000\):strip_icc\(\)/safari-truck-giraffes-micatosafaris-SAFARIGUIDETIPS0721-2549bb165aa34dc193cb8b6f3958654b.jpg](https://www.travelandleisure.com/thmb/Cp3v7EPYiVhuAwOO8yxYv4qUjY0=/750x0/filters:no_upscale():max_bytes(15000):strip_icc()/safari-truck-giraffes-micatosafaris-SAFARIGUIDETIPS0721-2549bb165aa34dc193cb8b6f3958654b.jpg) ,Erişim Tarihi: 08.05.2023.
- [32] Görsel 2, https://hackernoon.com/hn-images/1*anJ8xj06Q-xr6XosDF1Etw.jpeg, Erişim Tarihi: 08.05.2023.
- [33] Görsel 3, https://www.rcp-vision.com/wp-content/uploads/2020/06/800px-Lex_Av_E_92_St_06.jpg, Erişim Tarihi: 08.05.2023.
- [34] Görsel 4, https://hips.hearstapps.com/hmg-prod/images/kitchen-paint-colors-blue-1672376788.jpg?resize=480:*, Erişim Tarihi: 08.05.2023.