Fırat Üniv. Müh. Bil. Dergisi
34(1), 459-471, 2022

Araştırma Makalesi
https://doi.org/10.35234/fumbd.1060378

Derin Öğrenme Temelli Robotik Maske Kontrol Sistemi

Mustafa GÖKSU[1*], Ahmet ALKAN[2]
[*1,2] Elektrik-Elektronik Mühendisliği, Mühendislik ve Mimarlık Fakültesi, Kahramanmaraş Sütçü İmam Üniversitesi, Kahramanmaraş, Türkiye
[*1] mustafagoksu02@gmail.com, [2] aalkan@ksu.edu.tr

(Geliş/Received: 19/01/2022; Kabul/Accepted:14/02/2022)

Öz: Korona virüsün (COVID-19) hızlı bulaşması nedeniyle dünya büyük bir sağlık kriziyle karşı karşıya kalmıştır. Korona virüsün yayılmasını engellemek için Dünya Sağlık Örgütüne (WHO) göre en etkili tedbir, halka açık yerlerde ve kalabalık alanlarda maske takmaktır. Ancak kalabalık ortamlarda uzun süre kalan kişilerde sıkılma, boş verme ve umursamazlık gibi nedenlerle insanlar bu kuralı ihlal edebilmektedir. Bu nedenle kalabalık alanlarda insanların izlenmesi ve gerektiğinde ilgililerin uyarılarak toplum sağlığını korumak önem arz etmektedir. Bu çalışmada maske takmayan, maskesini yanlış takan ve maskesini doğru takan kişileri belirleme sürecini otomatikleştirmek için iki derin öğrenme modeli kullanan bir robotik model geliştirilmiştir. İnternetten elde edilen veri setleri ve çevreden alınan fotoğraflar kullanılarak özgün bir veri seti oluşturulmuştur. Geliştirilen yapay zekâ modellerinin daha iyi tahmin sonuçları verebilmesi için veri seti görüntüleri üzerinde veri çoğaltma (aynalama, döndürme) teknikleri kullanılmıştır. Çalışmada gerçek zamanlı olarak maskeli, maskesiz ve maskesini yanlış takan kişilerin tespiti gerçekleştirilmiş sesli olarak kişilere dönütler verilmiştir. Geliştirilen yapay zekâ modellerinde üç sınıf (maskeli, maskesiz, maskesini yanlış takan) için ortalama tahmin/sınıflandırma başarı oranı (mAP) %96,58 ile %98,45 olarak tespit edilmiştir. Hız ve eğitim süresi açısından YOLOv4-tiny algoritmasıyla geliştirilen modelin daha başarılı olduğu tespit edilmiştir. Çalışma kapsamında geliştirilen yapay zekâ modellerinin farklı donanımlar üzerinde gerçek zamanlı çalıştırılması için MKS (Maske Kontrol Sistemi) olarak adlandırılan etkileşimli yazılım önerilmektedir. MKS yazılımı geliştirilen hareketli bir robot üzerinde çalıştırılmıştır. Prototip robotla gerçekleştirilen uygulamalarda oldukça yüksek maske denetleme başarımları elde edilmiştir. Geliştirilen robotun, kullanımıyla kurumlara personel ve zaman tasarrufu sağlayabileceği, COVID-19 tedbirlerinin kontrolü ve toplum bilincinin arttırılmasında yararlı olacağı düşünülmektedir.

Anahtar kelimeler: Derin Öğrenme, COVID-19, YOLOv4, YOLOv4-tiny, Robot.

Deep Learning Based Robotic Mask Control System

Abstract: Due to the rapid transmission of the coronavirus (COVID-19), it has faced a major health crisis. According to the World Health Organization (WHO), the most effective measure to prevent the spread of the coronavirus is to wear a mask in public and crowded areas. However, people who stay in crowded environments for a long time may violate this rule for reasons such as boredom, disregard, and indifference. For this reason, it is important to monitor people in crowded areas and to protect public health by warning those concerned when necessary. In this study, a robotic model using two deep learning models was developed to automate the process of identifying people who do not wear masks, wear masks incorrectly, and wear masks correctly. An original data set was created using data sets obtained from the internet and photographs taken from the environment. For the developed artificial intelligence models to give better estimation results, data duplication (mirroring, rotation) techniques were used on the data set images. In the study, masked, unmasked, and masked people were identified in real-time, and verbal feedback was given to the people. In the developed artificial intelligence models, the mean Average Precision (mAP) for the three classes (with mask, without a mask, and with the wrong mask) was determined as 96.58% to 98.45%. It has been determined that the model developed with the YOLOv4-tiny algorithm is more successful in terms of speed and training time. Interactive software called MCS (Mask Control System) is recommended for the real-time running of artificial intelligence models developed within the scope of the study on different hardware. It has been determined that the model developed with the YOLOv4-tiny algorithm is more successful in terms of speed and training time. Interactive software called MCS (Mask Control System) is recommended for real-time running of artificial intelligence models developed within the scope of the study on different hardware. MKS software was run on a developed mobile robot. Very high mask inspection performances have been achieved in the applications performed with the prototype robot. It is thought that the developed robot can save time by reducing personnel costs to institutions with its use, and will be useful in controlling COVID-19 measures and increasing public awareness.

Key words: Deep Learning, COVID-19, YOLOv4, YOLOv4-tiny, Robot.

[*] Sorumlu yazar: mustafagoksu02@gmail.com. Yazarların ORCID Numarası: [1] 0000-0002-7235-2019, [2] 0000-0003-0857-0764

1. Giriş

COVID-19 virüsü ve bu virüsün etkileri tüm insanlığın sağlığını tehdit ederek, sadece sağlığımızı etkilememekte, yaşam şeklimizi de değiştirmeye başlamaktadır. COVID-19 hayatımıza maske takmak, sosyal mesafeye, temizlik ve hijyene dikkat etmek, kapalı ve kalabalık yerlerden kaçınmak gibi yeni zorunluluklar getirmiştir. Sağlık bakanlığımız kontrollü sosyal hayat stratejisiyle yaşam biçimimizi şekillendirmiştir. Yeni yaşam biçimimizde uymamız gereken maske kullanımının önemi Dünya Sağlık Örgütü tarafından yayınlanan makaleler ile vurgulanmaktadır.

Hastalığa yol açan SARS-CoV-2 RNA'nın havada 3 saat kalabildiği ve paslanmaz çeliklerde 48 saat, bakır yüzeylerde 4 saat, karton yüzeylerde 24 saat kalabildiği tespit edilmiştir [1].

Hastalıkla mücadelede asemptomatik şekilde geçiren vakaların yüksek olması önemli bir problemdir. Tüm enfeksiyonların %60 oranında hafif semptomlu veya asemptomatik ait olduğu bu vakaların virüsü hasta olmayanlara geçirebileceği tahmin edilmektedir [2].

Daha önceden yapılan bir araştırmaya göre COVID-19 virüsü ile mücadele etmek ve korunmanın en etkili yöntemlerinin kişisel hijyen kuralları, sosyal mesafe ve maske kullanımı olduğu ifade edilmiştir [3].

Virüsün her yeni varyantı farklı özellikler göstermekte, özellikle en güncel varyantı olan Omicron'un yayılma/bulaşma hızının daha da fazla olduğu belirtilmektedir. Dolayısıyla virüsle mücadelede en etkili yol temizlik, maske ve mesafeye dikkat etmektir. Ancak toplumsal hayatta iş/okul vb. ortamlarda mesafenin korunması mümkün olmadığından tek ve en etkili korunma aracı, kurallarına uygun maske kullanımıdır. Maske kullanımında kuralların alışkanlık haline gelmesi için etkili denetim ve kontrol çok önemlidir. Maske takmayarak kuralları ihlal edecek çok az bir grup, büyük çoğunluğun kurallara uymuş olmasını anlamsız hale getirebilmektedir. Maske kullanmayanların bu konu hakkında bilinçlendirilmesi ve maske kullanımının denetlenmesi önemlidir. Bu amaçla, çalışmada bir maske kontrol ve uyarı sistemi geliştirilmiştir. Maske kontrol sisteminin hayata geçirilmesi ile maske denetimi yapan özel ve kamu personelinin görev yükü azaltılarak iş gücünden tasarruf edilmiş olacaktır. Son yıllarda literatürde maske kontrolü yapabilen derin öğrenme yöntemleri ile geliştirilen değişik çalışmaların yapılmaya başlandığı görülmektedir.

Literatürdeki bir çalışmada 4 sınıfa ait (maskeli, maskesiz, yanlış maske kullanımı ve maske alanı) 52.635 görsel kullanılarak bir veri seti oluşturulmuş, bu veri seti ile 8 çeşit YOLO ağ algoritmaları eğitilmiştir. Çalışmanın eğitim sonuçlarına bakıldığında en yüksek mAP (ortalama hassasiyet değeri) değerini veren YOLOv4 ağı ile eğitilen model %71,69 değerine ulaşmıştır. Aynı çalışmada YOLOv4-tiny ağı ile geliştirilen modelin %57 mAP değerine ulaşmıştır [4].

1376 ve 873 adet maskeli ve maskesiz kişilere ait 2 farklı veri seti üzeri üzerinde yapılan diğer bir çalışmada yüz tespiti işleminden sonra yüzlerde maske olup olmadığı CNN ile sınıflandırılmıştır. Başarı oranları sırasıyla %95.77 ve %94.58 elde edildiği görülmüştür [5].

YOLOV4 ağının görüntü ölçekleme ve hesaplamalarını azaltacak sistemin önerildiği çalışmada yüz maskesinin tespiti için eğitilen modelin mAP değerinin %98,3 değeri elde edilmiştir [6].

Maske tespiti yapan başka bir çalışmada Pynq-YOLO-Net olarak isimlendirilen YOLO ağı ile geliştirilen model %97'lik algılama doğruluğu elde etmiştir [7].

YOLOv4 algoritması kullanılarak geliştirilen başka bir çalışmada maskeli ve maskesiz kişileri tanıyabilen modelin 4000 iterasyon sonunda %98 ortalama hassasiyet değerine (mAP) ulaştığı görülmüştür [8].

1376 adet görsel üzerinde destek vektörleri (SVM) ve MobileNetV2 birleştirilmesi ile yapılan başka bir çalışmada maske takan kişilerin sınıflandırılması yapılmış ve başarı oranı %97,1 elde edildiği görülmüştür [9].

Yapılan başka bir çalışmada 1415 adet veri seti üzerinde özellik çıkarma için ResNet50, nesne algılama için YOLOv2 algoritmaları kullanılarak %81 hassasiyet yüzdesine ulaşıldığı belirtilmektedir [10].

YOLOv3 algoritması ile maske tespiti yapan bir çalışmada 4000 iterasyon sonunda %96 başarı elde ettiği görülmüştür [11].

Maskeli ve maskesiz kişilerin tespiti için ilk aşamada yüzleri tespit eden bir modelden geçirildikten sonra ikinci aşamada tespit edilen yüzde maske olup olmadığı sınıflandıran CNN tabanlı bir ağ kullanılmış ve %99,5 doğruluk oranı elde edilmiştir [12].

Yüzde maske tespiti için MobileNetV2, VGG16 ve ResNet50 derin öğrenme algoritmalarının kullanıldığı çalışmada en iyi başarımın %97,82 ile VGG16 algoritması ile elde ifade edilmektedir [13].

YOLOv5 derin öğrenme algoritması ile maske tespiti yapan başka bir çalışmada 300 iterasyon sonunda %96,5 doğruluğa erişildiği görülmüştür [14].

SSD (Single Shot Detector) ve MobileNetV2 sınıflandırıcının birleştirilmesi ile yapılan başka bir çalışma %92,64 başarı oranı ulaştığı ve gömülü sistemlerde kullanılabilecek düzeyde olduğu belirtilmiştir [15].

4098 adet görsel ile oluşturulmuş bir ver seti üzerinde maskesini takan ve takmayan kişileri sınıflandırmak için MobileNetV2 derin öğrenme ağı üzerinden transfer öğrenme tekniği kullanılarak yapılan başka bir çalışmada %98'lik doğruluğa ulaşıldığı görülmüştür [16].

25.000 görüntüden oluşturulan maskesiz ve maskeli kişileri gerçek zamanlı raspberry pi donanımı üzerinde ayrıştıran bir çalışmada MobileNetV2 ve SSD mimarileri kullanılarak %96 doğruluk elde edildiği rapor edilmiştir [17].

Maskesini takan kişilere kapıyı otomatik olarak açmasını hedefleyen başka bir çalışmada MobileNetV2 kullanılmış %95,85 doğruluk elde edildiği görülmüştür [18].

Görüldüğü gibi son yıllarda literatürde maske kontrolü yapabilen derin öğrenme yöntemleri ile geliştirilen değişik çalışmaların yapılmaya başlandığı görülmektedir. Ancak geliştirdiğimiz maske kontrol sistemi sadece bir yazılım olmayıp, önerilen yazılımın işlevsel bir robot üzerinde gömülü sisteme taşınarak prototip bir ürüne dönüştürülmüş olması dikkate alınırsa, çalışmayla literatürdeki birçok çalışmanın ilerisine geçilmiştir.

Bu çalışmada, 2864 adet görsel 3 farklı sınıf için etkilendirilmiştir. Çalışmada kullandığımız veri setinde maskesiz, maskeli ve maskesini yanlış kullanan kişiler olmak üzere 3 sınıf bulunmaktadır. Veri setinin %70'lik kısmı eğitim için, %20'lik kısmı doğrulama ve % 10'luk kısmı test için ayrılmıştır. Veri seti YOLOv4-tiny ve YOLOv4 algoritmaları kullanılarak Model 1 ve Model 2 olarak isimlendirilen modeller oluşturulmuştur. Her iki modelle eğitim sonucunda elde edilen başarım metrikleri karşılaştırılmıştır. Elde edilen başarılı maske kontrol sistemi robot üzerinde de uygulanarak, başarılı bir prototip oluşturulmuştur. Çalışmada 2. bölümde kullanılan veri seti hakkında bilgiler verilmiştir. Kullanılacak yöntemler 3. bölümde özetlenmiştir. 4. bölümde deneysel çalışma ve uygulamalar verilerek elde edilen başarım metrikleri sunulmuştur. 5. bölümde tartışma sonuçlar verilmiştir.

2. Materyal

Çalışmada web'te ücretsiz olarak erişilebilen veri setleri ve çevremizden elde ettiğimiz fotoğraflarla oluşturulan özgün bir veri seti oluşturulmuştur [19-20]. Kullanılan veri seti Şekil.1'de örnekleri görülen özelliklerinde, JPG formatında görüntülerden oluşmaktadır. Çalışmada kullanılan bu görüntüler üç farklı sınıfta (maskeli, maskesiz, maskesini yanlış takan) etiketlenmiştir.

Önerilen MKS sistemi için Şekil 1.a-c'de sırasıyla maskesiz, maskeli ve maskesini yanlış takan kişileri sınıflandırılmasına ait 1835 görselin, 1328 daha önceden veri setlerini hazırlayanlar tarafından 507 tanesi de bu çalışmada veri setinin zenginleştirilmesi için yeni görüntüler eklenerek etiketlendirilmiştir.

(a) Maskesiz (b) Maskeli (c) Maskesini yanlış takan

Şekil 1. Veri setine kullanılan 3 sınıfa ait görüntülerden bir bölüm.

Veri çoğaltma işlemi sonrası kullanılan her sınıfa ait, eğitim ve doğrulama setini oluşturan veri sayıları Tablo 1'de sunulmuştur. Veri setinin %70'lik kısmı eğitim için, %20'lik kısmı doğrulama, %10'luk kısmı ise test için ayrılmıştır. Tablo 1'de veri setinin %70'lik kısmına denk gelen 1301 adet görüntüde aynalama, 90 derece saat yönü ve 90 derece saat yönünün tersine döndürme veri çoğaltma işlemleri uygulanarak 3903 görüntü sayısına çıkarılmıştır. Veri setinde yer alan bütün görüntüler 416*416 boyutuna göre tekrar boyutlandırılmıştır.

Tablo 1. Veri setinde bulunan sınıflara göre görüntü sayıları.

Görüntü sayısı	Maskesiz Kişiler	Maskeli Kişiler	Maskesini Yanlış Takanlar	Toplam
Eğitim seti %70	288	574	439	1301
Doğrulama seti %20	79	163	124	366
Test seti %10	36	82	50	168
Toplam	403	819	613	1835

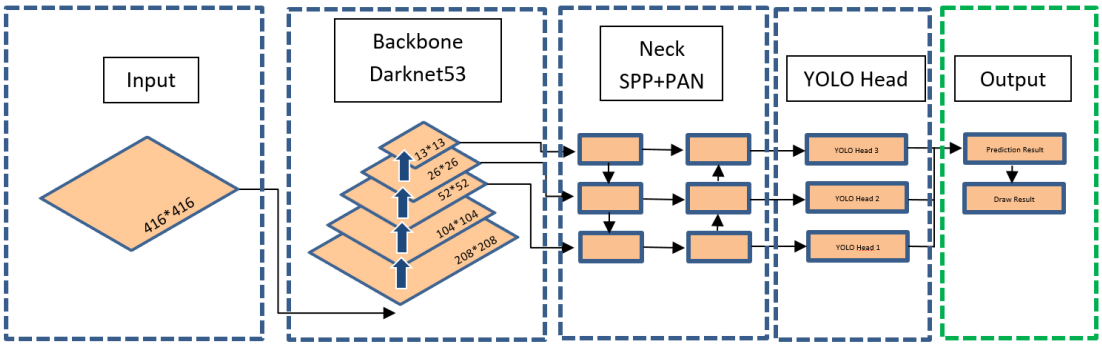
3. Metotlar

Bu bölümde çalışmada kullanılan modeller ve metotlar açıklanmaktadır. Yapay zekânın alt kümelerinden olan derin öğrenme tekniği klasik makine öğrenmesi tekniklerinden farklı olarak dış müdahale olmadan modellerin yanlış karar verip vermediği algılayabilen bir tekniktir [21]. Çalışmada derin öğrenme yöntemlerinden YOLOv4 ve YOLOv4-tiny nesne tespiti algoritmaları kullanılmıştır. YOLOv4-tiny YOLOv4'e göre katman sayısı ve modüller değiştirilerek gömülü sistemler, mobil cihazlar ve düşük donanıma sahip cihazlar için daha iyi FPS değerleri elde etmek için tasarlanmıştır [22].

Çalışmada kullanılan veri seti üzerinde iki farklı derin öğrenme algoritması (YOLOv4-tiny ve YOLOv4) kullanılarak yapay zekâ modelleri Google Colab platformu üzerinde Tesla P100-PCIE-16GB GPU kullanılarak eğitilmiştir. Her iki model için geliştirilen ortam ve iterasyon sayısı 6000 olarak belirlenmiştir. Model 1 ve Model 2 olarak adlandırılan modeller sırasıyla YOLOv4-tiny ve YOLOv4 algoritmalarının COCO veri seti üzerinde eğitilen ağırlıkları kullanılarak transfer öğrenmesi yöntemiyle eğitilmiştir.

3.1. YOLOv4

YOLOv4 algoritması Joseph Redmon tarafından 2016 yılında geliştirilmiştir. Gerçek zamanlı nesne tespiti yapabilen bu algoritma aynı anda nesnenin sınıfını ve nesnenin tespit edilen alanı çerçeve içine alabilmektedir [23]. Şekil 2'te çalışmada kullanılanYOLOv4 mimarisi görülmektedir. YOLOv4 mimarisinde Input bölümünde giriş olarak 416*416 çözünürlükte görüntüler alınmaktadır. Backbone bölümünde giriş olarak verilen görüntülerde özellik çıkarım işlemleri gerçekleşmektedir. Neck bölümünde daha fazla bilgi çıkarımları gerçekleştirilerek modelin daha iyi tahmin yapabilmesi sağlanmaktadır. Neck bölümü PAN ağı ile SPP katmanının birleşiminden oluşmaktadır. PAN ağı özellik toplamasını sağlarken, SPP havuzlama katmanı PAN ağı ile çıkarılan özelliklerin boyutlarından bağımsız olarak sabit boyutlu olmasını sağlamaktadır. YOLO Head bölümünde ise sınırlayıcı kutuların koordinatlarını, güven skoru bulunmakta ve sınırlayıcı kutuların sınıf tahmin işlemleri gerçekleşmektedir [24].



Şekil 2. YOLOv4 Mimarisi çalışmaya göre düzenlenmiştir (Lin et al. 2021).

3.2. YOLOv4-tiny

YOLOv-tiny algoritması YOLOv4 algoritmasından türetilmiştir. Yolov4-tiny algoritmasında iki CSPBlock modülü yerine ResNet-D ağında iki adet ResBlock-D modülü kullanılmaktadır ve bu da hesaplama karmaşıklığını azaltmaktadır. YOLOv4-tiny algoritmasında, YOLOv4 algoritmasında kullanılan CSPDarknet53 ağı yerine

Backbone bölümünde CSPDarknet53-tiny ağını kullanılmaktadır. Yolov4-tiny ile YOLOv4 ağı arasında farklardan biride ağ boyutu ve parametreleri YOLOv4-tiny büyük ölçüde küçültülmüştür [22].

3.3. MKS Robotu

MKS yazılımının hareketli olarak çalıştırabilmesi ve maske kontrolü yapılacak insanlara etkili geri dönütler verilebilmesi için hareket edebilen 85 cm uzunluğunda 27 cm genişliğinde bir robot geliştirilmiştir. Robotun dış kabuğu 3d yazıcılar ile 3 parça halinde hazırlanmıştır. Robotun enerjisi 4 adet 3.7V 6000 mAh li-ion piller ile sağlanmıştır. MKS yazılımının çalışmasını sağlayan kütüphanelerin kurulabilmesi ve yapay zekâ modellerinin çalışabilmesi için robotta Latte Panda Delta mini bilgisayarı kullanılmıştır. Mini bilgisayarda işletim sistemi olarak Windows Inspire kullanılmıştır. Robotunu hareketi bluetooth donanımı ve d.c. (doğru akım) motorları ile mobil uygulama üzerinden cep telefonu ile sağlanmaktadır [25].

4. Deneysel Çalışmalar ve Bulgular

Bu bölümde metotlar bölümünde anlatılan Model 1 ve Model 2'ye ait yapay zekâ modellerinin geliştirilmesi aşamasında, kişisel bilgisayarlarda çalıştırılması, mini bilgisayarlarda çalıştırılması ve robotun üzerinde uygulanmasıyla elde edilen bulgular yer almaktadır.

Tablo 2 de YOLOv4-tiny ve YOLOv4 derin öğrenme algoritmaları kullanılarak geliştirilen sırasıyla Model 1 ve Model 2 için metrikleri görülmektedir. Her iki model için geliştirilen ortam ve iterasyon sayısı aynı olarak belirlenmiştir.

Tablo 2. Modellere ait hassasiyet, geri çağırma, F1-skor, tespit zamanı IoU ve mAP değerleri.

Modeller	Hassasiyet (Precision)	Geri Çağırma (Recall)	F1-Skoru	Tespit Zamanı(ms)	IoU	mAP	Boyut (Mb)
YOLOv4-tiny	0,93	0,93	0,93	3-4	%74,74	%96,58	22,4
YOLOv4	0,94	0,97	0,96	20-21	%78,65	%98,45	244,2

Tablo 2 de Model 1 ve Model 2 için tespit zamanları Google Colab platformunda P100-PCIE-16GB GPU' lu bilgisayar üzerinde hesaplanmıştır. Model 1'in test verileri üzerinde 3-4 milisaniyede tespit işlemini gerçekleştirdiği, Model 2'nin ise 20-21 milisaniyede tespit işlemini gerçekleştirdiği görülmüştür. Model 1 ve Model 2'ye ait hassasiyet, geri çağırma, F1-skoru metrikleri sırasıyla 0,93 - 0,94, 0,93 – 0,97, 0,93 – 0,96 olarak tespit edilmiştir.

Nesne tespiti algoritmalarında modelinin başarısının tespitinde kullanılan IoU ve mAP metrikleri kullanılmaktadır [26].

$$IoU = \frac{Gerçekte\ olması\ gereken\ kutu\ alanı\ \cap Tahmin\ edilen\ kutu\ alanı}{\text{Gerçekte olması gereken kutu alanı } \cup\ Tahmin\ edilen\ kutu\ alanı}$$

Şekil 2'te veri setinde mor renkte belirtilen kutu alanı maskesiz kişinin modeller tarafından tahmin edilmeden önce etiketlenmesine ait alanı yani gerçekte olması gereken referans alanı (ground truth) göstermektedir. Yeşil renkte belirtilen kutu alanı geliştirilen yapay zekâ modellerinden Model 1'e ait tahmin alanıdır [25].

Bu çalışmada Model 1 ve Model 2'ye ait 3 sınıfa ait ortalama IoU değerleri sırasıyla %74,74 ve %78,65 olarak elde edilmiştir. Doğru pozitif, yanlış pozitif değerleri nesne tespiti yapan algoritmalarının performanslarını belirleyen metriklerinin hesaplamalarında kullanılmaktadır. Çalışmada IoU değeri 0,5'e eşit veya büyükse doğru pozitif olarak kabul edilmektedir. IoU değeri 0,5'in altında olursa yanlış pozitif olarak kabul edilmektedir. Yanlış negatif değeri ise güven eşiği olarak bu çalışmada belirlenen 0,25 in altında IoU değerine sahip tahminler olarak alınmaktadır. Doğru pozitif, yanlış pozitif ve yanlış negatif değerleri elde edildikten sonra nesne tespitinde kullanılan başarı metrikleri hesaplanmaktadır:

Şekil 3. IoU Metriği Kullanımı.

$$Hassasiyet(P) = \frac{Doğru\ Pozitif}{\text{Doğru Pozitif} + \text{Yanlış Pozitif}} \quad (1)$$

$$Geri\ çağırma(R) = \frac{Doğru\ Pozitif}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (2)$$

$$F1\ Skoru = \frac{2 * Geri\ çağırma * Hassasiyet}{\text{Geri çağırma} + \text{Hassasiyet}} \quad (3)$$

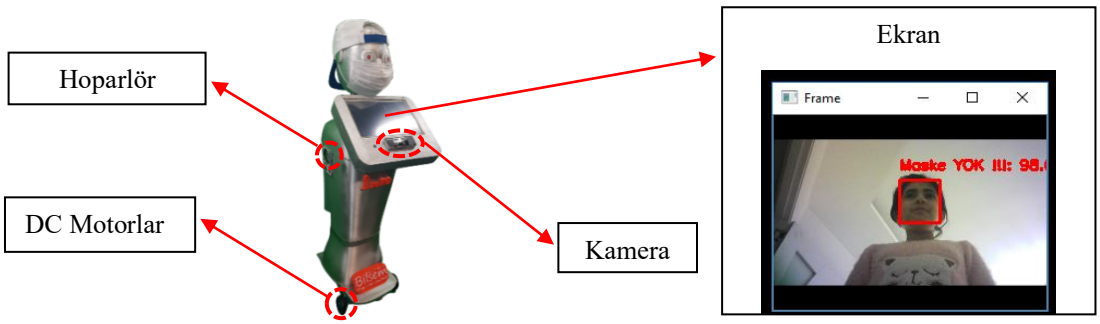
$$Ortalama\ Hassasiyet\ (AP) = \int_0^1 P(R)dR \quad (4)$$

$$Toplam\ Ortalama\ Hassasiyet\ (mAP) = \frac{\sum_{i=1}^{N} AP_i}{N} \quad (5)$$

AP değerleri çalışmada kullanılan 3 sınıfa ait tüm nesnelerin hassasiyet ve geri çağırma eğrisinin altında kalan alanlarının hesaplanması Denklem 4 ile elde edilir. Denklem 5 ile Denklem 4 ile elde edilen AP değerlerinin kullanılarak toplam ortalama hassasiyet değeri (mAP) hesaplanır [27]. Model 1 ve Model 2 için mAP değerleri sırasıyla %96,58 ile %98,45 elde edilmiştir.

4.1. MKS Robotu ve Bulguları

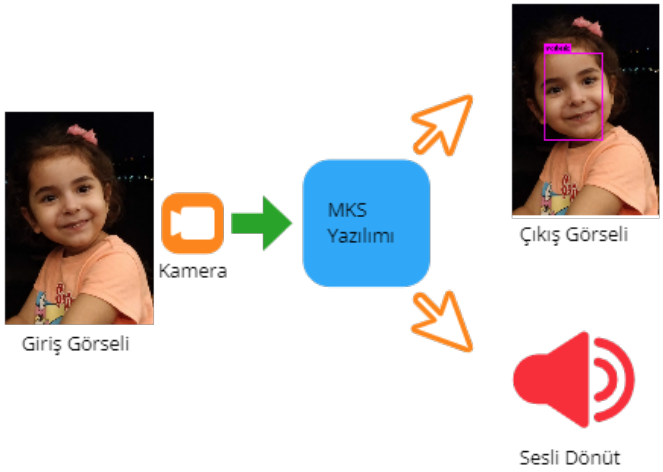
Çalışmada üretilen prototip MKS Robotu Şekil 4'te görülmektedir. Robotun ekranının altında yer alan kameradan görüntüler alınarak Model1 kullanılarak, MKS yazılımı ile maske kontrolü gerçekleştirilmektedir. Sesli dönütler robotun her iki yanında yer alan hoparlörler ile sağlanmıştır [25]. Robotun maske kontrolü sırasında sesli



Şekil 4. Geliştirilen Yapay Zekâ Destekli MKS Robot Prototipi.

dönütler ile beraber robotun ön tarafında yer alan 10 inch ekran ile kişilerin anlık olarak maske kontrollerinin sonuçları gösterilmektedir.

Şekil 5' te görüldüğü gibi dış dünyadan kamera ile anlık alınan görüntüler MKS yazılımı içerisinde bu çalışma kapsamında geliştirilen modellerden geçirilerek gerçek zamanlı maske kontrolü gerçekleştirerek ses çıkışı yapabilen donanımlar sayesinde sesli dönütler verebilmektedir. Robot üzerinde çalıştırılan MKS yazılımı ile maskesiz kişiye sesli olarak "Lütfen maskenizi takınız " şeklinde hoparlörler ile geri dönüt verilmiştir. Aynı şekilde maskesini doğru olarak kullanan kişilere "Maskenizi taktığınız için teşekkür ederim" mesajı verilirken, maskesini yanlış takan kişi tespit ettiğinde "Lütfen maskenizi doğru kullanınız" şeklinde sesli geri dönütler verilmektedir [25].



Şekil 5. MKS yazılımının çalışma şeması.

4.2. Modellere Ait Bilgisayar Bulguları

Model 1 ve Model 2' nin gerçek zamanlı çalıştırılması ve kişilere sesli uyarılar verilebilmesi için MKS yazılımı geliştirilerek farklı donanımlar üzerinde çalıştırılmış tahmin sonuçları ve FPS değerleri elde edilmiştir.

MKS sistemi kişisel bilgisayarlar, Raspberry Pi 4 ve Latte Panda Delta, Nvidia Jetson Nano mini bilgisayarlar üzerinde çalışabilmektedir. MKS sistemi Google Colab üzerinde eğitilen Model 1 ve Model 2 olarak isimlendirilen derin öğrenme modellerinin bir kamera vasıtasıyla çalıştırılmasına dayanmaktadır [25].

Çalışmada geliştirilen yapay zekâ modelleri Tablo 3'te gösterilen donanımlar üzerinde çalıştırılmıştır. MSI GP72 ve Nvidia Jetson Nano bilgisayarlarında GPU (Grafik İşlemci Birimi) donanımları varken diğer donanımlarda GPU bulunmamaktadır. MSI GP72 bilgisayarı diz üstü bilgisayar iken diğer donanımlar mini bilgisayar olarak sınıflandırılmaktadır. Mini bilgisayarlar MSI GP72 gibi diz üstü bilgisayarlar ve masaüstü bilgisayarlara göre daha az enerji tüketen bilgisayarlar olmakla beraber büyüklükleri daha küçüktür. MKS yazılımın mini bilgisayarlar, masaüstü bilgisayarlar, dizüstü bilgisayarlarda kullanımı ön görüldüğü için birçok donanım üzerinde çalıştırılmıştır. Mini bilgisayarlarda GPU donanımına sahip Nvidia Jetson Nano geliştirilen yapay zekâ modelleri üzerinde çalışma performansı Tablo 4'te dikkat çekmektedir. Düşük enerji tüketimi ve kapladığı alana itibari ile Nvidia Jetson Nano mini bilgisayarın robot projelerinde kullanımı önerilmektedir [25].

Tablo 3. Çalışmada kullanılan bilgisayar donanımları.

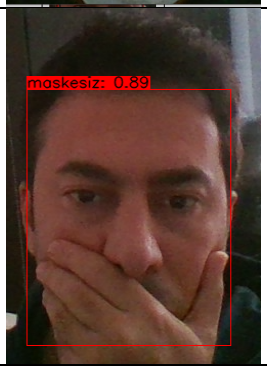
Donanım	Özellikleri
MSI GP72	**CPU:** Intel i7 7700 HQ 2.8 GHz **Ram:** 16 GB **GPU:** Nvidia Geforce GTX 1050 4GB
Nvidia Jetson Nano	**CPU:** Quad-core ARM A57 @ 1.43 GHz **Ram:** 4 GB 64 bit LPDDR4 **GPU:** 128 Çekirdek Maxwell GPU

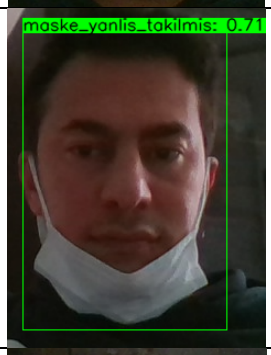
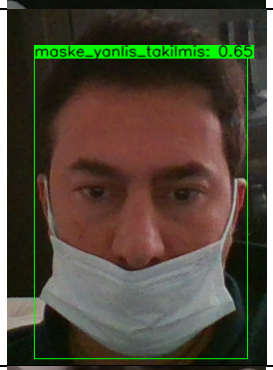
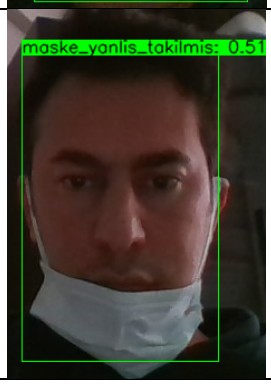
Latte Panda Delta	**CPU:** Intel 8. Nesil Celeron İşlemci N4100 **Ram:** 4 GB LPDDR4 2400MHz **GPU:** Yok
Raspberry Pi 4- 8GB	**CPU:** 1.5 GHz Broadcom BCM2711 dört çekirdekli ARM Cortex-A72 **Ram:** 8 GB LPDDR4 -3200 **GPU:** Yok
Raspberry Pi 4- 4GB	**CPU:** 1.5 GHz Broadcom BCM2711 dört çekirdekli ARM Cortex-A72 **Ram:** 4 GB LPDDR4-3200 **GPU:** Yok

MKS yazılımı kişilerin yüzünde maskesiz, maskeli ve yanlış maske kullanımı yapan kişilerin gerçek zamanlı tespitini yaparak kişileri sesli olarak uyarabilmektedir. MKS yazılımının uygulanması için Raspberry Pi 4 ve Nvidia Jetson Nano mini bilgisayarlara sırasıyla Linux tabanlı Raspberry Pi Os ve Ubuntu işletim sistemi yüklenmiştir. Raspberry Pi 4 ve Nvidia Jetson Nano mini bilgisayarlara işletim sistemine ssh kabuk kontrol yazılımı olan Git Bash ve VNC Viewer adlı uzaktan kontrol yazılımı ile bağlanılmış, gerekli kütüphaneler olan opencv, lxml, tqdm, tensorflow, absl-py, easydict, matplotlib, pillow, omxplayer VNC Viewer üzerinden yüklenmiştir. MKS sisteminin maske tanıma ve sesle uyarı yapabilmesi için ilgili kütüphaneler Raspberry Pi 4 ve Nvidia Jetson Nano mini bilgisayarlara yüklenmiştir. MSI GP72 kişisel bilgisayara Windows 10 Pro. ve Latte Panda Delta mini bilgisayara Windows inspire işletim sistemleri yüklenmiştir. Raspberry Pi 4 ve Nvidia Jetson Nano mini bilgisayarlara yüklenen yazılımların Windows işletim sistemine ait kütüphane ve yazılımları MSI GP72 kişisel bilgisayara ve Latte Panda Delta mini bilgisayarına GPU ve CPU sürümleri yüklenmiştir. Python dili ile hazırlanan Maske Kontrol Sistemi(MKS) yazılımı Raspberry Pi 4 ve Nvidia Jetson Nano, Latte Panda Delta ve MSI GP72 model bilgisayarda çalıştırılmıştır. MKS yazılımının uygulanması ile elde edilen tahmin görselleri ve FPS değerleri Tablo 4' te görülmektedir.

Tablo 4. MKS yazılımının farklı donanımlar üzerindeki tahminleri ve FPS sonuçları.

Model	**Donanımlar**	**MKS Çıktıları**	**Çalışma Zamanı Türü**	**FPS Minimum ve Maksimum Değerleri**
Model 1	MSI GP72		GPU	13,62 -21,42

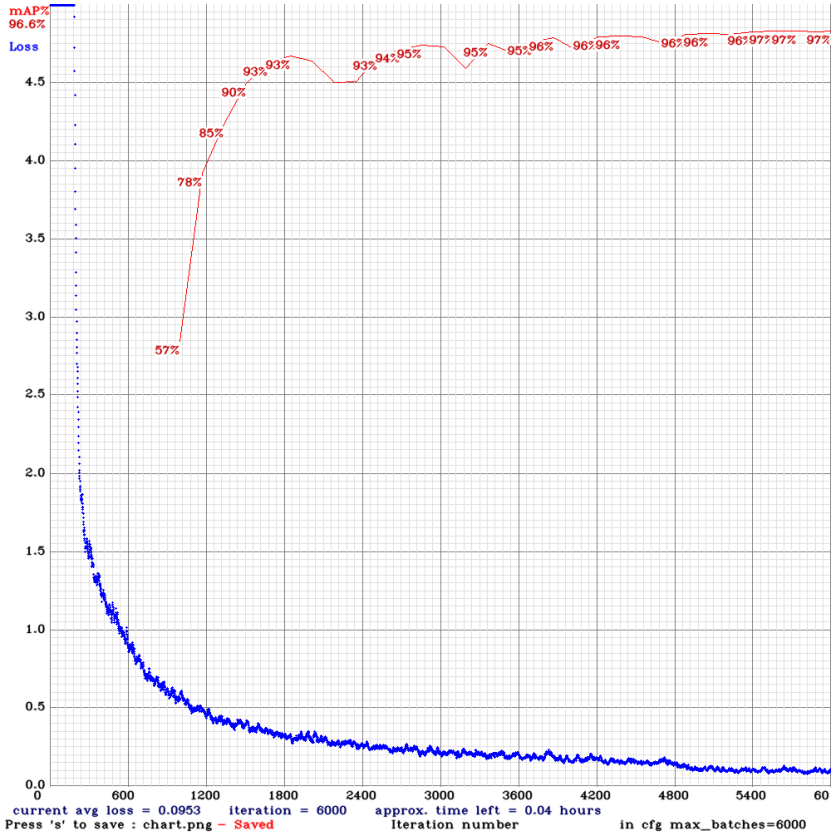
Model 2	MSI GP72		GPU	1,9 - 2,03
Model 1	Latte Panda Delta 4 GB		CPU	1,82 - 2,06
Model 2	Latte Panda Delta 4 GB		CPU	0,19 – 0,22
Model 1	Raspbery Pi 4 – 4 GB		CPU	0,59 – 1,54

Model 2	Raspbery Pi 4 – 4 GB		CPU	0,11 – 0,16
Model 1	Raspbery Pi 4 – 8 GB		CPU	0,84 – 1,72
Model 2	Raspbery Pi 4 – 8 GB		CPU	0,15 – 0,17
Model 1	Nvidia Jetson Nano 4 GB		GPU	8.2 – 12,3

Model 2	Nvidia Jetson Nano 4 GB		GPU	1,1 – 1,8

4.3. Google Colab Bulguları

Şekil 6'da YOLOv4-tiny ile geliştirilen Model 1'e ait mAP, kayıp (loss) ve iterasyon değerlerine ait grafik görülmektedir. Bu grafik Google Colab üzerinde eğitim sonunda elde edilmiştir. Şekil 6'da 6000 iterasyon sonunda mavi renk ile gösterilen kayıp (loss) değerlerinin azaldığı ve kırmızı renk ile gösterilen mAP değerlerinin arttığı görülmüştür.



Şekil 6 - YOLOV4-tiny ile geliştirilen Model 1'e ait mAP, kayıp (loss) / İterasyon Grafiği

5. Sonuçlar ve Tartışma

Bu çalışmada 3 sınıflı (maskeli, maskesiz ve maskesini yanlış takan) maske kontrolünü gerçekleştiren 2 adet yapay zekâ modeli geliştirilmiştir. Geliştirilen yapay zekâ modelleri Google Colab platformu üzerinde GPU'lu bilgisayarlar üzerinden eğitilmiştir. Model 1 ve Model 2 yapay zekâ modellerine ait mAP ve ortalama IoU değerleri sırasıyla %96,58 ile %98,45 - %74,74 ve %78,65 olarak elde edilmiştir. Çalışma için geliştirilen Model 1 ve Model 2 MKS yazılımı ile farklı donanımlara sahip bilgisayarlar üzerinde çalıştırılmıştır. Tablo 4'te MKS yazılımına ait FPS ve tahmin görselleri görülmektedir. Bu çalışmada geliştirilen yapay zekâ modellerinin GPU' lu bilgisayarlarda CPU' ya sahip bilgisayarlara oranla daha yüksek FPS değerlerine ulaştığı Tablo 4' te görülmektedir. Model 2'nin Model 1' e göre daha başarılı olmasının nedeni Model 2'nin YOLOv4 ile geliştirilmiş olmasıdır. Yine Tablo 4'te Model 1'in FPS değerlerinin Model 2'ye göre daha yüksek olduğu görülmektedir. Model 1'in Model 2'ye göre daha hızlı çalışmasını gösteren FPS değer farkının oluşmasındaki neden modellerin geliştirilme sürecinde kullanılan YOLO algoritmalarından kaynaklanmaktadır. Model 1 için kullanılan YOLOv4-tiny, Model 2 için kullanılan YOLOv4 ağına göre daha az katmana sahiptir. Tablo 2 de Modellere ait hassasiyet, geri çağırma, F1-skoru metrikleri sırasıyla 0,93 - 0,94, 0,93 – 0,97, 0,93 – 0,96 olarak tespit edilmiştir. Ayrıca Tablo 2'de Model1 ve Model2'nin dosya boyutlarının sırasıyla 22,4 – 244,2 Mb (Megabayt) olduğu görülmektedir.

MKS sisteminin halka açık alanlarda robot üzerinde uygulanması ile maske takmayan kişiler sesli olarak uyarılmıştır. MKS sistemiyle maske takmayan kişiler kolaylıkla tespit edilerek, bu kontroller için kullanılacak iş gücünden tasarruf sağlanmıştır. MKS sistemi ile maske takan kişilere sesli teşekkür mesajları iletilerek maske kullanımı pekiştirilmek hedeflenmiştir. MKS sistemi geliştirilen robot üzerinde çalıştırılarak maske kontrolü gerçekleştirilmiştir.

MKS sistemi donanımsal olarak çok fazla bileşenden oluşmadığı gibi uygulanabilirlik olarak istenilirse devlet kurumları veya özel kurumların girişlerinde sistem sabit kameralara ve hoparlörlere bağlanarak turnike girişlerinde maske kullanımın kontrolü sağlanabilir. MKS sistemi mini bilgisayar ve kişisel bilgisayarlar üzerinde çalışabilmektedir. Çalışmada elde edilen prototipin farklı senaryolarla, yeni eklenecek modüllerle daha işlevsel ve yenilikçi bir ürün haline getirilmesi ve üretilmesi amaçlanmaktadır.

Teşekkür

Çalışmada gerçekleştirilen prototip robot Teknofest 2021'de İnsanlık Yararına Teknoloji Yarışması Sağlık ve İlk Yardım kategorisinde üçüncülük elde etmiştir. Desteklerinden dolayı Teknofest düzenleme kuruluna teşekkür ederiz.

Kaynaklar

[1] Neeltje van Doremalen ve diğerleri, "Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1," (2020).
[2] Qiu J., Covert coronavirus infections could be seeding new outbreaks. 2020.
[3] Şirin H., Özkan S., "Dünyada ve Türkiye'de COVID-19 Epidemiyolojisi," Kulak Burun Boğaz ve Baş Boyun Cerrahisi Dergisi, vol. 28, (2020).
[4] Kumar A. ve diğerleri, "Scaling up face masks detection with YOLO on a novel dataset," Optik, vol. 239, (2021).
[5] Das A. ve diğerleri, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," in 2020 IEEE 17th India Council International Conference, INDICON 2020, 2020.
[6] Yu J., Zhang W., "Face mask wearing detection algorithm based on improved yolo-v4," Sensors, vol. 21, no. 9, (2021).
[7] Said Y., "Pynq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic era," International Journal of Advanced Computer Science and Applications, vol. 11, no. 9, (2020).
[8] Mahurkar R. R., Gadge N. G., "Real-time Covid-19 Face Mask Detection with YOLOv4," in Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021, 2021.
[9] Oumina A. ve diğerleri, "Control the COVID-19 Pandemic: Face Mask Detection Using Transfer Learning," in 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS 2020, 2020.
[10] Loey M. ve diğerleri, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," Sustainable Cities and Society, vol. 65, (2021).
[11] Bhuiyan M. R. ve diğerleri, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," in 2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020, 2020.
[12] Abbasi S. ve diğerleri, "A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset," in 26th International Computer Conference, Computer Society of Iran, CSICC 2021, 2021.
[13] Gedik O., Demirhan A., "Comparison of the effectiveness of deep learning methods for face mask detection,"

Traitement du Signal, vol. 38, no. 4, (2021).
[14] Ieamsaard J. ve diğerleri, "Deep Learning-based Face Mask Detection Using YoloV5," in Proceeding of the 2021 9th International Electrical Engineering Congress, iEECON 2021, 2021.
[15] Nagrath P. ve diğerleri, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," Sustainable Cities and Society, vol. 66, (2021).
[16] Mercaldo F., Santone A., "Transfer learning for mobile real-time face mask detection and localization," Journal of the American Medical Informatics Association, vol. 28, no. 7, (2021).
[17] Kumar G. P., "Face Mask Detection with Raspberry Pi," International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. VI, (2021).
[18] Rani N. ve diğerleri, "Real-Time Face Mask Detection Using Raspberry Pi and Camera," in Lecture Notes in Networks and Systems, 2022, vol. 300 LNNS.
[19] Cabani A. ve diğerleri, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," Smart Health, vol. 19, (Mar. 2021).
[20] "Face Mask Detection ~12K Images Dataset | Kaggle," Online: https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset [Accessed: 09 January 2022]
[21] Aalami N. ve diğerleri, "Derin Öğrenme Yöntemlerini Kullanarak Görüntülerin Analizi." https://doi.org/10.1016/j.eswa.2018.09.022
[22] Jiang Z. ve diğerleri, "Real-time object detection method for embedded devices.".(2020). https://arxiv.org/abs/2011.04244
[23] Redmon J. ve diğerleri, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-December.
[24] Lin H. H. ve diğerleri, "Efficient cell segmentation from electroluminescent images of single-crystalline silicon photovoltaic modules and cell-based defect identification using deep learning with pseudo-colorization," Sensors, vol. 21, no. 13, (Jul. 2021).
[25] Göksu M., "Yapay Zeka Destekli İnteraktif Sağlık Robotu". (2022), Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tez Çalışması.
[26] Parico A. I. B., Ahamed T., "Real time pear fruit detection and counting using yolov4 models and deep sort," Sensors, vol. 21, no. 14, (2021).
[27] Kulshreshtha M. ve diğerleri, "Oatcr: Outdoor autonomous trash-collecting robot design using yolov4-tiny," Electronics (Switzerland), vol. 10, no. 18, (Sep. 2021).