



T.C.

KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY ZEKA DESTEKLİ İNTERAKTİF SAĞLIK
ROBOTU**

MUSTAFA GÖKSU

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

KAHRAMANMARAŞ 2022

T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY ZEKA DESTEKLİ İNTERAKTİF SAĞLIK
ROBOTU

MUSTAFA GÖKSU

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

KAHRAMANMARAŞ 2022

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada, alıntı yapılan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Mustafa GÖKSU



Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bilgilerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

YAPAY ZEKA DESTEKLİ İNTERAKTİF SAĞLIK ROBOTU

(YÜKSEK LİSANS TEZİ)

MUSTAFA GÖKSU

ÖZET

Korona virüsün insanlar arasında hızla yayılmasından dolayı dünya büyük bir sağlık krizi ile mücadele etmek zorunda kalmıştır. Salgının başında belirli bir tedavisi olmayan korana virüs, hava yoluyla bulaştığı için sağlık kuruluşlarında yığılmalara sebep olmuştur. Bu nedenle, Dünya Sağlık Örgütü (WHO) COVID-19 virüsünün insanlar arasında yayılmasını engellemek için maske kullanımının önemini vurgulamaktadır. Buna rağmen insanların maske kullanımından sıkılmasından veya maske kullanımını ciddiye almamasından dolayı maske kullanımını ihlal ettikleri görülmektedir. Bu ihlalin neticesinde sağlık kurumlarında hastalara yakınlaşarak verilen sağlık hizmetlerinden dolayı sağlık çalışanlarına virüs bulaşmıştır. Bu çalışmada salgın gibi koşullarda sağlık çalışanlarının hastalara yaklaşmadan bazı sağlık hizmetlerinin verilmesini sağlayan yapay zekâ destekli bir robot geliştirilmiştir. Robot ile sağlık çalışanlarının salgın koşullarından kötü etkilenmemesi için hastalara yaklaşmadan ateş ölçme, oksijen saturasyon ölçümü, nabız ölçümü ve hasta kayıt işlemleri yapabilmektedir. Ayrıca robot ile beraber Maske Kontrol Sistemi (MKS) adı verilen 3 sınıfa (maskesini doğru kullanan, maskesini yanlış kullanan ve maske kullanmayan) göre maske kontrolünü gerçekleştirebilen yapay zekâ destekli yazılım geliştirilmiştir. Geliştirilen yapay zekâ modellerinde mAP değerleri %97,63, %98,09, %98,37, %98,11 olarak elde edilmiştir.

Anahtar Kelimeler: Derin Öğrenme, COVID-19, YOLO, Yapay Zeka, Robot.

Kahramanmaraş Sütçü İmam Üniversitesi

Fen Bilimler Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı, Ağustos/2022

Danışman: Prof. Dr. Ahmet ALKAN

Sayfa Sayısı: 62

ARTIFICIAL INTELLIGENCE ASSISTED INTERACTIVE HEALTH ROBOT

(M.Sc. THESIS)

MUSTAFA GÖKSU

ABSTRACT

Due to the rapid spread of the corona virus among people, the world has had to deal with a major health crisis. Corona virus, which did not have a specific treatment at the beginning of the epidemic, caused agglomeration in health institutions because it was transmitted through the air. Therefore, the World Health Organization (WHO) emphasizes the importance of using masks to prevent the spread of the COVID-19 virus among humans. Despite this, it is seen that people violate the use of masks because they are bored with the use of masks or because they do not take the use of masks seriously. As a result of this violation, health workers were infected with the virus due to the health services provided by getting closer to the patients in health institutions. In this study, an artificial intelligence supported robot has been developed that enables healthcare professionals to provide some healthcare services without approaching patients in conditions such as epidemics. With the robot, it is possible to measure temperature, oxygen saturation measurement, pulse measurement and patient registration without approaching patients so that healthcare workers are not adversely affected by the epidemic conditions. In addition, artificial intelligence supported software has been developed with the robot, which can perform mask control according to 3 classes (using the mask correctly, using the mask incorrectly, and not using the mask) called the Mask Control System (MKS). In the developed artificial intelligence models, mAP values were obtained as 97.63%, 98.09%, 98.37%, 98.11%.

Keywords: Deep Learning, COVID-19, YOLO, Artificial Intelligence, Robot.

Kahramanmaraş Sütçü Imam University

Graduate School of Natural and Applied Sciences

Department of Electric-Electronic Engineering, August/2022

Supervisor: Prof. Dr. Ahmet ALKAN

Page number: 62

TEŞEKKÜR

Bu tez çalışması süresince kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve destek olan danışman hocam Sayın Prof. Dr. Ahmet ALKAN' a, ayrıca maddi ve manevi olarak her daim yanımda olan kıymetli eşim Şafak GÖKSU' ya ve kızlarım Çağla Ela GÖKSU ve Elif Simge GÖKSU' ya teşekkürü bir borç bilirim.

Mustafa GÖKSU



İÇİNDEKİLER

	Sayfa No
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ.....	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Kapsamı.....	1
1.2. Önceki Çalışmalar.....	2
1.3. Tezin Organizasyonu	5
2. MASKE KONTROL SİSTEMİ	6
3. YAPAY ZEKÂ.....	8
3.1. Yapay Sinir Ağları	8
3.1.1. Algılayıcılar.....	9
3.1.2. Aktivasyon Fonksiyonları	10
3.1.2.1. Sigmoid Fonksiyon.....	11
3.1.2.2. ReLU Fonksiyon	11
3.1.2.3. Hiperbolik Tanjant Fonksiyon.....	12
3.1.2.4. Softmax Fonksiyon.....	13
3.2. Derin Öğrenme	13
3.2.1. ESA	14
3.2.2. Nesne Algılama	15
3.2.2.1. YOLO	15
3.2.2.2. R-CNN.....	16
3.2.2.3. SSD.....	17
4. MATERYAL VE METOT	18
4.1. Materyal	18
4.2. Metot.....	19

4.2.1. YOLOv4.....	19
4.2.2. YOLOv4-tiny	20
4.2.3. MKS Robotu	20
4.2.3.1. MKS Yazılımı	22
4.2.3.2. MKS Robot Tasarımı ve Çıktısı	23
4.2.3.3. MKS Robot Hareket Bölümü	23
4.2.3.4. MKS Robot Ekranı	24
4.2.3.5. MKS Robotu Mini Bilgisayarı	25
4.2.3.6. Sesli Asistan	26
4.2.3.7. Kablosuz Programlama Devresi	27
4.2.3.8. Pulse Oksimetre	27
4.2.3.9. MSSQL Server	28
5. DENEYSEL ÇALIŞMA	29
5.1. MKS Robotu ve Bulguları	32
5.2. Modellere Ait Bilgisayar Bulguları	35
5.3. Google Colab Bulguları	40
6. SONUÇLAR VE ÖNERİLER	44
7. KAYNAKLAR.....	46

ŞEKİLLER DİZİNİ

Şekil 1. 1. Sağlık çalışanlarının salgın sürecinde kullandığı giysi ve ekipmanlar.....	1
Şekil 2. 1. Alışveriş mağazalarında yapılan ateş ölçümü ve maske kontrolü.....	6
Şekil 3. 1. Örnek YSA yapısı.	9
Şekil 3. 2. Tek katmanlı algılayıcı yapısı.	9
Şekil 3. 3. Çok katmanlı algılayıcı yapısı.	10
Şekil 3. 4. Sigmoid fonksiyon grafiği.....	11
Şekil 3. 5. ReLU fonksiyon grafiği.	12
Şekil 3. 6. Hiperbolik tanjant fonksiyon.....	12
Şekil 3. 7. Leaky ReLU fonksiyon.	13
Şekil 3. 8. ESA yapısı.....	14
Şekil 3. 9. YOLO ızgara yapısı ve tespit çerçevesi.	15
Şekil 3. 10. R-CNN mimarisi	16
Şekil 3. 11. SSD mimarisi	17
Şekil 4. 1. Veri setinde bulunan sınıflara ait görüntülerden bir bölüm	18
Şekil 4. 2. YOLOv4 mimarisi..	20
Şekil 4. 3. MKS Robotunun bölümleri ve genel görünümünün tasarımı.	22
Şekil 4. 4. L298N motor sürücü.	23
Şekil 4. 5. 6V 185 rpm DC motor.	24
Şekil 4. 6. MKS robot bilgilendirme yazılımı arayüzü.....	25
Şekil 4. 7. Latte Panda Delta mini bilgisayar yapısı.....	26
Şekil 4. 8. Kablosuz mikrodenetleyici programlayıcı devresi.	27
Şekil 4. 9. GY-MAX30100 sensörü	28
Şekil 5. 1. Tahmin ve referans kutu alanları.....	30
Şekil 5. 2. Robot üzerinde MKS sisteminin çalışma çıktısı.	32
Şekil 5. 3. MKS robotunun görevleri.	33
Şekil 5. 4. MKS sisteminin çalışma şeması.....	33

Şekil 5. 5. Kablosuz programlayıcı devresi.	34
Şekil 5. 6. MKS Robotu' nun farklı açılardan görüntüleri.	34
Şekil 5. 7. Model 1' e ait mAP ve kayıplar grafiği.	40
Şekil 5. 8. Model 2' ye ait mAP ve kayıplar grafiği.	41
Şekil 5. 9. Model 3'e ait mAP ve kayıplar grafiği.	42
Şekil 5. 10. Model 4'e ait mAP ve kayıplar grafiği.	42
Şekil 5. 11. 9000 iterasyon sonucunda elde edilen mAP ve kayıplar grafiği.	43



ÇİZELGELER DİZİNİ

Tablo 4. 1. Veri setinde yer alan 3 sınıfa ait görüntü sayıları.....	19
Tablo 5. 1. Geliştirilen yapay zekâ modellerine ait başarımlar değerleri.....	29
Tablo 5. 2. Çalışmada kullanılan donanımlar.....	35
Tablo 5. 3. Geliştirilen modellerin farklı donanımlardaki sonuçları	37



SİMGELER VE KISALTMALAR DİZİNİ

MKS	: Maske Kontrol Sistemi
ESA	: Evrişimsel Sinir Ağları
YSA	: Yapay Sinir Ağları
R-CNN	: Bölge Bazlı Evrişimsel Sinir Ağları
Fast R-CNN	: Hızlı Bölge Bazlı Evrişimsel Sinir Ağları
Faster R-CNN	: Daha Hızlı Bölge Bazlı Evrişimsel Sinir Ağları
SSD	: Tek Atış Dedektörü
YOLO	: Sadece Bir Kez Bak
GPU	: Grafik İşlem Birimi
SVM	: Destek Vektörleri
CPU	: Merkezi İşlem Birimi
DC	: Doğru Akım
IoU	: Kesiştirilmiş Bölgeler Ölçütleri
mAP	: Genel Ortalama Hassasiyet
AP	: Ortalama Hassasiyet
FPS	: Saniye Başına Kare
FP	: Yanlış Pozitif
TP	: Doğru Pozitif
GB	: Gigabayt
PLA	: Biopolimer Polilaktik Asit

1. GİRİŞ

COVID-19 salgını tüm dünyayı etkilemiş ve hizmet sektöründe büyük değişimlere neden olmuştur. COVID-19 virüsü solunum yoluyla insanlar arasında kısa sürede yayılmış ve çoğu insanın ölümüne neden olmuştur. İnsan sağlığını tehdit eden bu virüsten dolayı çoğu iş sektörü çalışma şekillerine kısa zamanlı veya uzaktan çalışma kararı alarak çalışanlarını korumaya çalışmıştır. Sağlık sektöründe ise verilen sağlık hizmetlerinin büyük bir çoğunluğunun yakından hizmet verme zorunluluğu olmasından dolayı sağlık çalışanları diğer sektörlere oranla en riskli çalışma koşullarına sahiptir. Dünyada ve ülkemizde salgın süresince yakından hizmet vererek çalışan birçok sağlık çalışanı enfekte olmuş ve salgından kötü etkilenmiştir (Çalışkan & Metintas, 2020).

Sağlık kuruluşlarında hastaların bilgilerini alırken ve ilk tetkiklerini yaparken mevcut uygulamalarda hastalara yakınlaşmadan bu işlemler yapılmamaktadır. Bu işlemler gerçekleştirilirken sağlık çalışanları maske, siperlik, bone ve tulum gibi malzemeler ile kendilerini korumaya çalışmaktadır (Gül ve arkadaşları, 2020). Sağlık çalışanlarının saatlerce maske ve siperlik ile çalışması onları yıpratmaktadır (Yüncü & Yılan, 2020). Şekil 1.1’ de sağlık çalışanlarının salgın sürecinde hizmet verirken salgından dolayı kullanmak zorunda olduğu giysi ve ekipmanlar görülmektedir.



Şekil 1.1. Sağlık çalışanlarının salgın sürecinde kullandığı giysi ve ekipmanlar.

1.1. Tezin Amacı ve Kapsamı

Bu tez çalışması ile ülkemizde sağlık çalışanlarının hastalara yaklaşmadan ilk kayıt işlemleri ve ilk tetkiklerinin robot ile yapılarak sağlık çalışanlarının çalışma koşullarını iyileştirmek hedeflenmiştir. Sağlık çalışanları salgın durumlarında ve diğer şartlarda sağlıklarını tehlikeye atmadan çalışma koşullarına kavuşacaklardır. Tez çalışmasında

önerilen robotun sağlık kuruluşlarında maske kontrolü, kayıt, danışmanlık, ateş ölçme, oksijen satürasyonu, nabız ölçümlerini gerçekleştirmesi hedeflenmektedir. Çalışmanın hayata geçirilmesi ile sağlık çalışanlarının izolasyonu sağlanarak danışmanlık, kayıt ve ilk tetkikleri yapan personelin başka alanlarda istihdam edilmesine bağlı olarak ülkemizde sağlık alanında iş gücü kazanımı öngörülmektedir. Robotun korona virüs salgını gibi gelecekte oluşabilecek salgın koşullarında da kullanılması sağlık çalışanlarının izolasyonunu sağlayacağı düşünülmektedir.

1.2. Önceki Çalışmalar

Literatürde 4 sınıfa ait (maskesini doğru kullanan, maske kullanmayan, maskesini yanlış kullanan ve maske bölgesi) kullanılarak elde edilen bir veri seti üzerinde 8 çeşit YOLO derin öğrenme ağı kullanılarak yapay zekâ modelleri geliştirilmiştir. Çalışmada en yüksek mAP değeri YOLOv4 ağı ile %71,69 olarak elde edilmiştir. Aynı veri seti üzerinde YOLOv4-tiny geliştirilen modelin %57 mAP değerine ulaştığı görülmüştür (Kumar ve arkadaşları, 2021).

COVID-19 virüsünden kaynaklanan hastalıklar ile yapılan mücadelede hastalık belirtisi göstermeden hastalığı geçiren vakaların yüksek sayıda olması önem arz etmektedir. Çalışmada enfeksiyonların %60 'ının hafif belirtili veya belirtisiz vakalara ait olduğu görülürken %60'lık bu kısma ait vakaların virüsü hasta olmayanlara bulaştırılabileceği ifade edilmiştir (Qiu, 2020).

Literatürde 1376 ve 873 adet görüntüden oluşan bir çalışmada görüntü üzerinde yüz tespiti işlemi gerçekleştirildikten sonra yüzlerde maske varlığı ESA ile kontrol edilmiştir. Veri setleri üzerinde yapılan çalışmalarda %95.77 ve %94.58 başarı değeri elde edildiği görülmüştür (Das ve arkadaşları, 2020).

YOLOv4 ağının görüntü ölçekleme ve hesaplama işlemlerini azaltabilen farklı bir çalışmada yüz maskesinin tespiti için eğitilen modelin mAP değerinin %98,3 değeri olarak elde edildiği görülmüştür (Yu & Zhang, 2021)

YOLOv4 ağı kullanılarak maske kontrolü gerçekleştirilen bir çalışmada 4000 iterasyon sonunda %98 mAP değeri elde edildiği görülmüştür (Mahurkar & Gadge, 2021).

1376 adet görüntüden oluşan bir veri seti üzerinde maske takan kişilerin tespiti için destek vektörleri (SVM) ve MobileNetV2 birlikte kullanılması ile elde edilen ağ üzerinde başarı oranının %97,1 elde edildiği görülmüştür (Oumina ve arkadaşları, 2020).

1415 adet görüntüden oluşan veri seti üzerinde görüntülerden özellik çıkarma işlemleri için ResNet50, görüntüler üzerinde nesne algılama için YOLOv2 derin öğrenme algoritmaları kullanılarak %81 mAP değerine ulaşıldığı görülmüştür (Loey ve arkadaşları, 2021).

Maske tespitinin gerçekleştiği farklı bir çalışmada YOLOv3 derin öğrenme algoritması kullanılarak 4000 iterasyon sonunda %96 başarı elde edilmiştir (Bhuiyan ve arkadaşları, 2020).

Maske tespiti için 3 adet derin öğrenme algoritmalarının kullanıldığı farklı bir çalışmada en yüksek başarı değerinin %97,82 olarak VGG16 algoritması elde edildiği görülmüştür (Gedik & Demirhan, 2021).

Maske tespitinin SSD ve MobileNetV2 ağlarının birlikte kullanılarak yapıldığı bir çalışmada %92,64 başarı elde edildiği görülmüştür. Ayrıca önerilen sistemin gömülü sistemler için uygun olduğu ifade edilmiştir (Nagrath ve arkadaşları, 2021).

4098 adet görüntüden oluşturulan bir veri seti üzerinde maskeli ve maskesiz kişiler MobileNetV2 ağı ile sınıflandırılmıştır. Yapay zekâ modelinin %98'lik doğruluğa ulaştığı görülmüştür (Mercaldo & Santone, 2021).

25.000 görüntüden oluşturulan bir veri setinde maske kontrolü gerçek zamanlı olarak Raspberry pi mini bilgisayar üzerinde gerçekleştirilmiştir. MobileNetV2 ve SSD ağları kullanılan çalışmada %96 doğruluk değerine ulaşılmıştır (Kumar ve arkadaşları, 2021).

MobileNetV2 derin öğrenme modeli kullanılarak geliştirilen bir çalışmada %95,85 doğruluk elde edilmiştir. Aynı çalışmada maskesini takan kişilere kapıyı otomatik açan bir sistem geliştirildiği görülmüştür (Rani ve arkadaşları, 2022).

Yapılan farklı bir çalışmada halka açık olan tren istasyonları, metro istasyonları, havaalanı girişleri gibi yerlerde kameradan alınan görüntüler üzerinde maske kontrolü gerçekleştirilmiş kişi maskesini doğru takana kadar kapılar açılmamıştır (Manhas ve arkadaşları, 2022).

Raspberry Pi kullanılarak geliştirilen bir çalışmada maske takan kişileri %94,71, maske takmayan insanları %97,7 ve insan vücut sıcaklığını %99,49 başarıyla ölçebilen bir sistem geliştirilmiştir (Najib ve arkadaşları, 2021).

Koronavirüs salgın döneminde gerçekleştirilen bir çalışmada hastaneye yatmak zorunda kalan kişilerin kandaki oksijen satürasyon değerleri incelenmiştir. Bu incelemelerde hastaneye başvuran 6194 kişide nefes darlığı, ateş ve öksürük olduğu hastaneye yatmak zorunda kalan kişilerin %89’unda oksijen satürasyon değerinin düşük olduğu görülmüştür (Akhavan ve arkadaşları, 2020).

Literatürde yapılan bir çalışmada Dünya Sağlık Örgütü’ ne (WHO) göre Kasım 2020’ nin sonuna gelindiğinde COVID-19 virüsünden 60 milyon kişinin etkilendiği vefat sayısının 1.410.378’ i aştığı ifade edilmiştir. Çalışmada korona virüsle mücadelede maske kullanımının önemi vurgulanırken, maske kontrolü için yapılan çalışmalarda maskesiz yüzler için yaklaşık olarak %99, maskeli yüzlerin tespiti için %95 olduğu DSFD ve RetinaFace ile mAP değerleri elde edilmiştir (Batagelj ve arkadaşları, 2021).

Maske takan ve maske takmayan kişileri tespit etmek için önerilen bir çalışmada hibrit olarak Faster R-CNN ağı ve InceptionV2 yapısı kullanılmış ve %91,13 genel başarıyla ölçeklenen bir model geliştirilmiştir (Wang ve arkadaşları, 2021).

Tez ile önerilen robota benzer olarak düşünebilecek robot Pepper, sağlık kuruluşuna gelen kişileri karşılamakta ve onları istedikleri yerlere gitmeleri için bilgilendirmektedir (URL 1, 2019). Literatürde yer alan Robox-m4mt isimli robot hareketli olarak üzerinde yer alan kamerası ile insan alın bölgesini algılayarak kızıl ötesi sensörleri ile ateş ölçümü gerçekleştirmektedir (URL 3, 2021). 2020 yılında SoftBank Robotics tarafından üretilen robot tezde yer alan robot gibi maskeli ve maskesiz kişileri tespit edebilmekte onları kibarca uyarabilmektedir (Reuters, 2020). Fakat maskesini yanlış takan kişileri tespit etmemekte, sağlık alanında tezde hedeflenen ateş ölçme, oksijen satürasyonu, nabız ölçümü ve hasta kayıt işlemlerini gerçekleştirmemektedir.

Bu tez çalışmasında kullanılan veri seti 1077 adet görüntüden oluşturulmuştur. Veri seti maskesini yanlış kullanan, maskesini doğru kullanan ve maskesini kullanmayan kişiler olmak üzere 3 sınıftan oluşturulmuştur. Yapay zekâ modellerinin eğitiminde veri setinin %70 kullanılırken %20’si doğrulama ve % 10’u ise test verisi için bölümlendirilmiştir. Oluşturulan veri seti üzerinde YOLOv4 ve YOLOv4-tiny derin öğrenme ağlarından Model 1, Model 2, Model 3 ve Model 4 olarak isimlendirilen yapay zekâ modelleri oluşturulmuştur. Oluşturulan 4 modele ait başarı metrikleri, grafikleri karşılaştırılmıştır. Oluşturulan yapay zekâ modelleri tez kapsamında oluşturulan robot üzerinde hareketli olarak denenmiştir. Ayrıca Maske Kontrol Sistemi (MKS) yazılımının sabit kullanımı için yapay zekâ modelleri çeşitli bilgisayar donanımlarında denenmiştir.

Literatür taramalarından görüldüğü gibi günümüzde maske kontrolü yapabilen birçok uygulamalar vardır. Ancak bu çalışmada geliştirilen maske kontrol sisteminin hareketli robotlar üzerinde kullanılmasına ek olarak robot üzerinde yer alan sensörler, diğer donanımlar ve çalışmaya özgü yazılımlar sayesinde sesli hasta kayıt, soru-cevap, oksijen satürasyonu, nabız ölçümü, ateş ölçümü işlemlerinin tek çatı altında toplanması ile literatürdeki birçok çalışmanın ilerisine geçilmiştir.

1.3. Tezin Organizasyonu

Önerilen tez çalışması yedi bölümden oluşmaktadır. Birinci bölüm olan giriş bölümünde tez çalışmasının amacı ve kapsamından bahsedilirken bu bölümde tez konusu ile alakalı olarak literatürde bulunan çalışmalara yer verilmiştir.

İkinci bölümde maske kontrol sistemine ilişkin bilgiler verilmiştir. Maske kontrol sistemini oluşturan bölümler ve nerelerde kullanılacağı hakkında bilgiler verilmiştir.

Üçüncü bölümde tez çalışmasında kullanılan yapay zekâ yöntemlerine ait teorik bilgiler yer almaktadır.

Dördüncü bölümde tez çalışmasında kullanılan yöntemler ve çalışmada kullanılan donanımlardan bahsedilmiştir.

Beşinci bölümde tez kapsamında yapılan deneysel süreçlerde elde edilen yapay zekâ modelleri ve yazılımlar hakkında bilgiler verilmiştir. Ayrıca bu bölümde yapılan çalışmalara ait bulgular ve elde edilen başarımların kriterleri verilmiştir.

Sonuçlar bölümünde geliştirilen yapay zekâ modellerinin başarımların değerleri ve robot üzerindeki yazılımların sonuçları değerlendirilmiştir. Tez kapsamında kullanılan donanımların geliştirilen yapay zekâ modelleri ile çalışma performansları değerlendirilmiştir.

En son bölüm olan kaynaklar bölümünde çalışmada yararlanılan kaynaklara yer verilmiştir.

2. MASKE KONTROL SİSTEMİ

COVID-19 virüsünün bulaşma riskini azaltan en önemli tedbirlerden birisi olan maske kullanımı, Dünya Sağlık Örgütü ve Sağlık Bakanlığı tarafından önerilmektedir. Bu kapsamda doğru maske kullanımı Şekil 2.1’ de görüldüğü gibi kurum ve kuruluşların giriş noktalarında bir veya daha fazla personel tarafından kontrol edilmektedir.



Şekil 2. 1. Alışveriş mağazalarında yapılan ateş ölçümü ve maske kontrolü.

COVID-19 virüs salgını sürecinde hizmet sektörü başta olmak üzere insan yoğunluğunun fazla olduğu kurum ve kuruluşlar kötü etkilenmiştir (Koyuncu ve arkadaşları, 2020). Özellikle sağlık çalışanları yakından verilen sağlık hizmetlerinden dolayı COVID-19 virüsüne karşı mücadelede en riskli grubu oluşturmaktadır. Sağlık kuruluşlarına gelen hastaların tetkik ve tedavileri sürecinde, hastalara yaklaşılarak verilen hizmetler sağlık çalışanlarının virüse maruz kalma riskini arttırmaktadır. Bu bağlamda bu çalışma ile sağlık kurumlarına girişte, sağlık çalışanlarının verdiği hizmetlerden olan maske kontrolü, ateş ölçme, saturasyon ölçümü, nabız ve hasta kayıt işlemlerinin bir robot tarafından gerçekleştirilmesi hedeflenmektedir. Sağlık kuruluşlarının giriş bölgelerinde görev yapabilecek hareketli robot ile verilecek hizmetler sayesinde hem personel tasarrufu sağlanacak hem de sağlık çalışanlarının bulaş riski azaltılmış olabilecektir. Ayrıca geliştirilen robot, sağlık kuruluşlarında, alışveriş merkezlerinde ve halka açık alanlarda salgın ile ilgili kuralları ve bilgilendirme mesajlarını sesli bildirimler şeklinde ifade ederek insanların bilinçlendirilmesine hizmet edecektir. Maske kontrollerinin herhangi bir çalışana ihtiyaç duyulmadan yapılabilmesi için teze özgü bir yapay zekâ modeliyle 3 sınıfa ait maske kontrolü (maskesini doğru kullanan, maskesini yanlış kullanan ve maske kullanmayan) gerçekleştirilmiştir. Maske kontrolleri robot üzerinde YOLOv4-tiny algoritması ile üretilen Model 1 ve Model 3 olarak isimlendirilen yapay zekâ modelleri kullanılarak

gerçekleştirilmiştir. YOLOv4-tiny algoritması ile üretilen yapay zekâ modelleri mini bilgisayarlarda hızlı çalışabilen performansı yüksek bir algoritmadır (Pestana ve arkadaşları, 2021).



3. YAPAY ZEKÂ

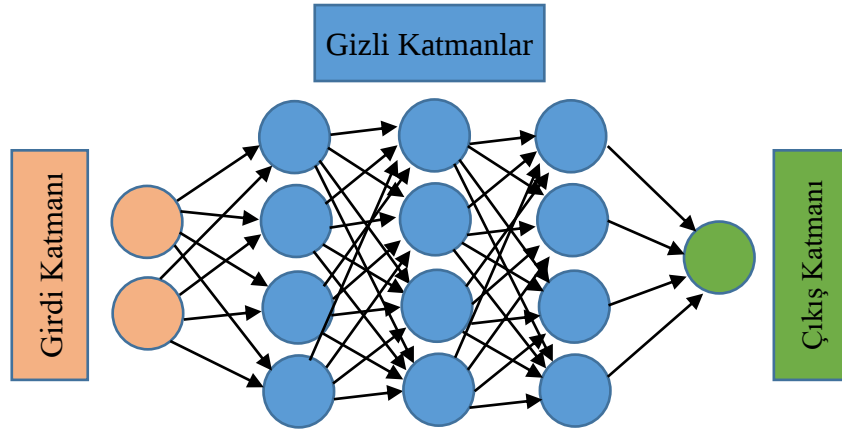
İnsanoğlu geçmişten günümüze doğada yaptığı gözlemler ile yaşama ait sorunları çözmeye çalışmıştır. Kuşların kanat yapılarından uçak üretimine, balıkların yüzgeç ve deri yapılarından balık adam giysisi gibi birçok alanda insanlar çözümler ve ürünler geliştirmişlerdir. Bilgisayar biliminin kurucusu olarak kabul edilen Alan Turing'in “Makineler düşünebilir mi?” ilk yapay zekâ fikrini ortaya atması sonrasında bilim insanlarının çabaları sayesinde insanların karar verebilme yeteneğinin makinelere aktarılabilmesi gerçekleştirilmiştir (Sarı, 2021). Yapay zekâ insanlara gerek olmadan makinelerin tıpkı insanlar gibi düşünüp karar verebilmesini algoritmalar ile sağlayan yapılar olarak ifade edilmektedir (Sucu & Ataman, 2020). Bir başka çalışmada yapay zekâ insanların çözemeyeceği karmaşık problemleri, insanın düşünme şekillerini taklit ederek çözebilen teknoloji olarak ifade edilmektedir (Uzun, 2020).

Yapay zekâ teknolojisinde gerçekleşen gelişmeler, büyük veri kümelerine erişim ve bulunan yeni yöntemlere göre alt dallara ayrılmıştır. Makine öğrenmesi, yapay zekâ kavramının ortaya atılmasından sonra 1959 yılında model tanıma çalışmaları ve sayısal öğrenme çalışmalarından geliştirilen yapay zekânın bir alt dalı olarak ortaya çıkmıştır (Öğüt, 2019). Derin öğrenme ise özellik ve dönüştürme için birçok katmanlı yapının kullanıldığı makine öğrenmesi sınıfıdır (Şeker ve arkadaşları, 2017).

3.1. Yapay Sinir Ağları

Biyolojik olarak insanın öğrenme yeteneği incelendiğinde sinir hücresi içerisinde gerçekleşen kimyasal olaylar ve bu olayların tetiklediği sinir hücreleri incelenerek matematiksel modeller ortaya konulmuştur. Ortaya konan matematiksel modellerde sinir hücresi içerisinde yer alan kimyasal işlemler matematikteki fonksiyonlar olarak modellenmiş, çıktılarda bir sonraki yapay sinir hücresine aktarılmıştır. Birden fazla sinir hücresinin birbirine bağlanması ve paralel işlem yapabilmesi yapay sinir ağlarının (YSA) ortaya çıkmasını sağlamıştır. YSA yapısında yapay sinir hücreleri katmanlar halinde kullanılmaktadır. Yapısal olarak yapay sinir hücreleri girdi katmanı, gizli katmanlar ve çıkış katmanı olarak dizilmektedir. Şekil 3.1’ de YSA yapısında girdi katmanında bilgi girişi yapılmakta, gizli katmanlarda yer alan mavi renkteki yapay sinir hücrelerinde bilgiler paralel olarak işlenerek çıkış katmanına iletilmektedir. Şekil 3.1 ‘de giriş katmanında öğrenilmesi istenilen 2 değer için tanımlanmışken, gizli katmanlar giriş katmanında girilen değerler üzerinde öğrenme işleminin gerçekleşmesi için hesaplamaların yapıldığı katmanlardır. Çıkış

katmanı ise gizli katmanlarda yapılan hesaplamalar sonunda ağın ürettiği tahmin sonucunun dış dünyaya aktarıldığı katmandır (Nabiyev, 2012).

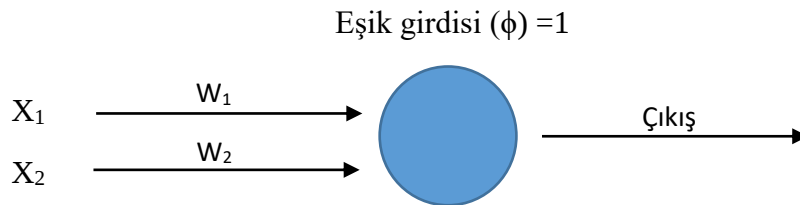


Şekil 3. 1. Örnek YSA yapısı (Arı ve arkadaşları, 2017).

Literatürde yapay sinir ağlarının insan beyninin çalışma yapısından esinlenilerek bilgi alabilen ve bilgi paylaşabilen, birbirine paralel olarak çalışabilen yapay sinir hücrelerinden oluşan bir yapıya sahip olduğu ifade edilmiştir (Keskenler, 2017).

3.1.1. Algılayıcılar

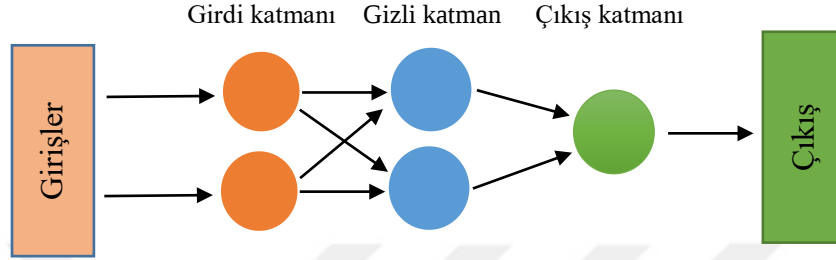
Tek katmanlı algılayıcılar genellikle doğrusal problemlerin çözümünde kullanılmaktadır. Giriş ve çıkış bölümlerinde tek katmandan oluşan doğrusal sinir ağlarıdır. Şekil 3.2 'de tek katmanlı algılayıcı görülmektedir. Eşik girdisi değeri girişlerden alınan değerlere göre ağın çıktısının 0 olmasını engellemek için kullanılmaktadır (Arı ve arkadaşları, 2017). Şekil 3.2 'de X_1 ve X_2 giriş değerlerini temsil ederken, W_1 ve W_2 değerleri giriş değerlerine göre üretilen ağın ağırlık değerleridir. Çıkış değeri ise ağırlıklandırılmış giriş değerlerinin eşik değerleri ile toplanması ile elde edilmektedir (Arı ve arkadaşları, 2017).



Şekil 3. 2. Tek katmanlı algılayıcı yapısı (Arı ve arkadaşları, 2017).

Tek katmanlı algılayıcıların çözümünde başarısız olduğu doğrusal olmayan problemlerin çözümü için çok katmanlı algılayıcılar geliştirilmiştir. Çok katmanlı algılayıcılarda giriş katmanı, gizli katmanlar ve çıkış katmanları yer almaktadır. Bu

katmanlar ileri ve geriye yayılım algoritmaları ile birbirine bağlıdır. İleriye doğru yapılan işlemler çıkış bölümde değerlendirilerek hataların azaltılması için geriye yayılım algoritması kullanılır. Geri yayılımda ağırlıklar tekrar güncellenerek ileriye yayılım işlemi gerçekleştirilir. Bu işlem hatalar en aza indirilene kadar devam eder (Arı ve arkadaşları, 2017). Şekil 3.3' de çok katmanlı algılayıcı yapısı görülmektedir. Gizli katman ve çıkış katmanlarında eşik değerler ağırlıklandırılmış giriş değerlerine eklenmektedir.



Şekil 3.3. Çok katmanlı algılayıcı yapısı (Kabalıcı, 2014).

3.1.2. Aktivasyon Fonksiyonları

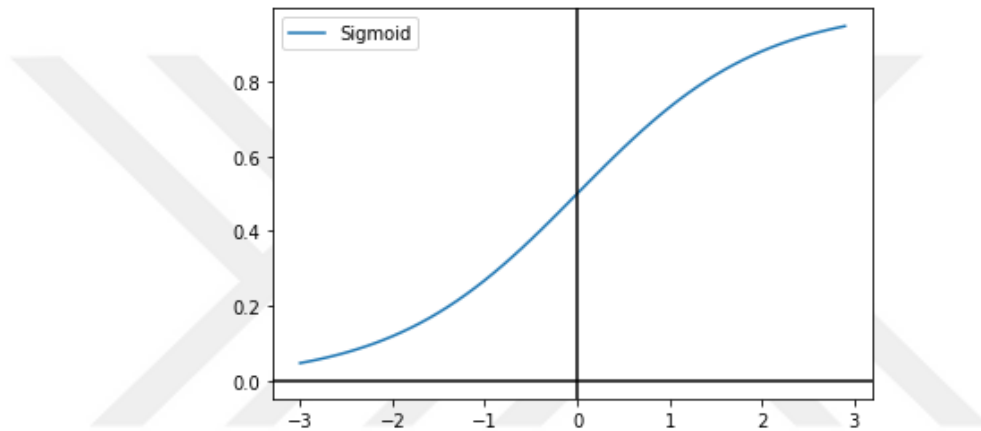
YSA matematiksel modellemelerden yararlanılarak oluşturulmuş ileri ve geri yayılım özelliklerine sahip algoritmalarından oluşmaktadır. YSA' da ilk zamanlarda giriş değerleri ağırlıklar ile çarpıldıktan sonra eşik değerleri ile toplanarak çıkış olarak diğer yapay sinir hücresinin girişine verilmekteydi (Ataseven, 2013). Bu şekilde giriş değerlerinin doğrusal olarak işlenmesi sadece doğrusal problemlerin çözümünü sağlamıştır. Karmaşık problemlerin çözülmesi için aktivasyon fonksiyonları geliştirilmiştir (Kılıçarslan ve arkadaşları, 2021). Aktivasyon fonksiyonun kullanımı genel olarak yapay sinir hücresinin giriş katmanında gelen değerlerin ağırlık değerleri ile çarpılarak bir eşik değeri ile toplandıktan sonra doğrusal olmayan bir fonksiyondan geçirilme işlemidir (Küçük ve arkadaşları, 2021). Aktivasyon fonksiyonu işlemi sonucunda elde edilen değer çıkış katmanına veya diğer yapay sinir hücrelerine giriş değeri olarak verilmektedir. YSA' da öğrenmenin gerçekleşmesi için ağırlık değerlerinin çıkış katmanında elde edilen değerlere göre güncellenebilmesi için geri yayılım algoritmaları kullanılmaktadır (Ataseven, 2013). Geri yayılım algoritmalarının kullanılabilmesi için aktivasyon fonksiyonları karmaşık ve doğrusal olmayan türevlenebilir fonksiyonlardan seçilmektedir (Kılıçarslan ve arkadaşları, 2021).

3.1.2.1. Sigmoid Fonksiyon

Yaygın olarak kullanılan aktivasyon fonksiyonudur. Giriş alarak aldığı girdi değerini $[0,1]$ aralığına dönüştürmektedir. Denklem 3.1 ‘den anlaşılacağı gibi parametre olarak aldığı x değeri 1 bölü $1 + e^x$ olduğu için x değeri ne olursa olsun 1 ile 0 arasında bir değer elde edilecektir (Yang & Yang, 2018).

$$f(x) = \frac{1}{1 + e^x} \quad (3.1)$$

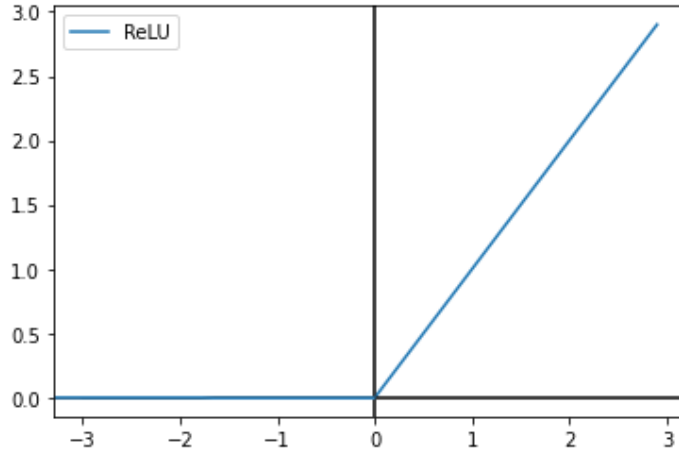
Şekil 3.4 ‘ de sigmoid fonksiyonun giriş değerlerine karşılık 0 ile 1 değerinde çıktı ürettiği değerler grafikte mavi renkle gösterilmektedir.



Şekil 3. 4. Sigmoid fonksiyon grafiği.

3.1.2.2. ReLU Fonksiyon

ReLU aktivasyon fonksiyonu giriş değeri olarak negatif değerlerde çıkışa 0 değeri atarken pozitif sayı değerlerinde doğrusal fonksiyon gibi davranmaktadır. Burada dikkat edilmesi gereken nokta negatif değere sahip değerlere 0 değeri atamasından dolayı ağıın hızlı çalışmasına katkıda bulunurken bu değerlere sahip değerlerin öğrenmede dikkate alınmamasına neden olmaktadır (Yang & Yang, 2018). Şekil 3.5’ de ReLU fonksiyon grafiğinde mavi renk ile gösterilen çıkış değerlerinde; negatif giriş değerlerine karşılık çıkışın 0 değeri aldığı pozitif değerlerde ise aynı değeri çıktı olarak verdiği görülmektedir.

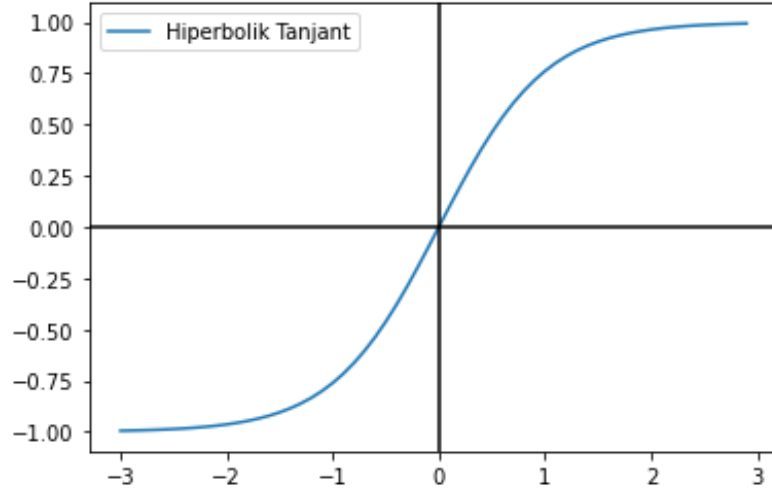


Şekil 3. 5. ReLU fonksiyon grafiği.

3.1.2.3. Hiperbolik Tanjant Fonksiyon

Hiperbolik tanjant fonksiyonu, giriş değerlerini $[-1, 1]$ aralığına dönüştürmektedir. Denklem 3.2’ de görüleceği gibi x parametresi olarak girilen negatif ve pozitif değerler için sonuç -1 ile $+1$ arasında elde edilmektedir. Şekil 3.6 ‘ da hiperbolik tanjant fonksiyonun giriş değerlerine karşılık mavi renk ile gösterilen çıkış değerlerinde -1 ile $+1$ değerleri aldığı görülmektedir (Yang & Yang, 2018).

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.2)$$



Şekil 3. 6. Hiperbolik tanjant fonksiyon.

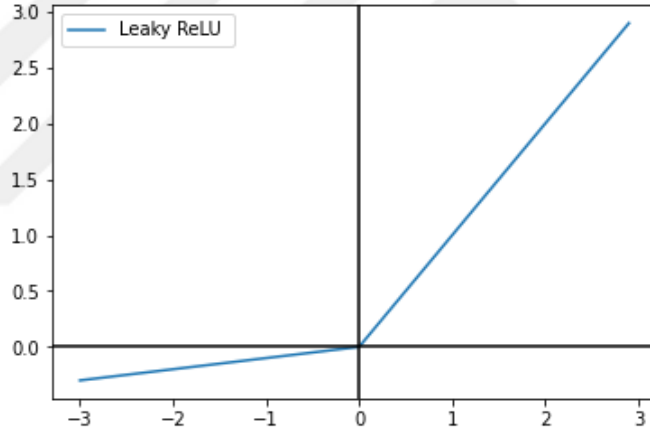
3.1.2.4. Softmax Fonksiyon

Softmax aktivasyon fonksiyonu genellikle çıkış katmanında birden fazla sınıflandırma yapılan ağlarda kullanılmaktadır. Denklem 3.3’ de görüleceği gibi verilen değerler 0 ile 1 arasında bir değere dönüştürülmektedir (Karakuş, 2018).

$$f(z_j) = \frac{e^{z_j}}{\sum_{j=1}^K e^{z_j}} \quad (3.3)$$

3.1.2.5. Leaky ReLU Fonksiyon

Leaky ReLU fonksiyonu ReLU fonksiyonunun negatif giriş değerlerini 0 sıfıra eşitlemesi ve çıkış değerinin türevinin de sıfır olması dezavantajını gidermek için geliştirilmiştir (Xu ve arkadaşları, 2015). Şekil 3.7’ de Leaky ReLU fonksiyonun grafiğinde görüldüğü gibi negatif giriş değerleri mavi renk ile belirtilen çıkış değerlerinde sıfır değeri yerine negatif değerler almaktadır.



Şekil 3. 7. Leaky ReLU fonksiyon.

3.2. Derin Öğrenme

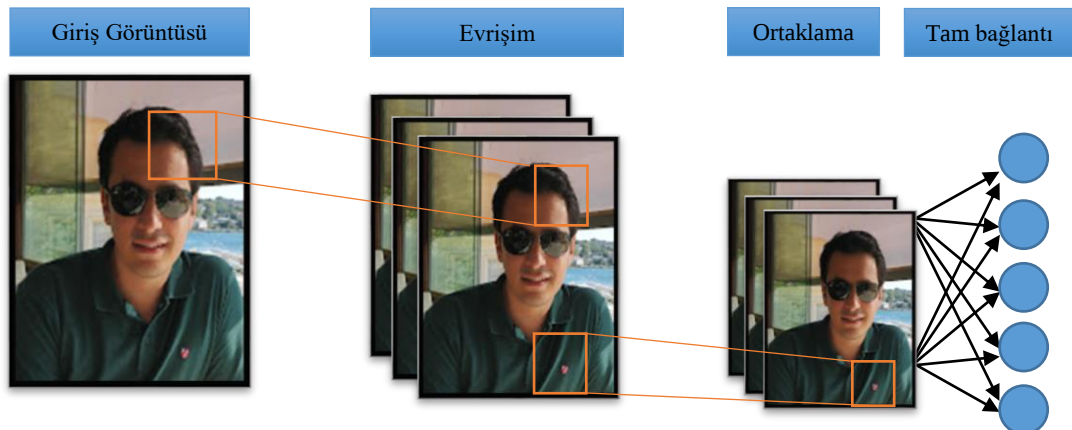
YSA’ lar insan vücudunda yer alan sinir hücrelerinden esinlenilerek oluşturulmuştur (Aydın, 2019). Grafik işlem birime (GPU) sahip kartların ortaya çıkması, işlemci (CPU) ve ram kapasitelerinin artması çok katmanlı yapay sinir ağlarının günümüzde kullanımını sağlamıştır. Derin öğrenme büyük veri kümelerinden çok katmanlı yapay sinir ağları kullanılarak yapay zekâ modellerinin geliştirildiği makine öğrenmesinin alt dalıdır (Seyyarer ve arkadaşları, 2020). Derin öğrenme makine öğrenmesi yöntemi olmasına rağmen dış müdahale olmadan tahmin sonuçlarının doğru olup olmadığını algılayabilen algoritmalardır (Aalami, 2019). Tahmin sonuçlarını iyileştirmek için geri yayılım metodu kullanılması ile

günümüzdeki derin öğrenme algoritmalarının temeli oluşmuştur. Geri yayılım algoritması yapay sinir ağlarında yer alan sinir hücrelerindeki ağırlık değerlerini güncelleyerek ileriye doğru tekrar yayılım yapılmasını sağlamaktadır (Güldemir, 2022).

Günümüzde derin öğrenme algoritmaları ile tıp alanında erken teşhis, savunma sanayide hedef tayini, taşıtlarda otonom sürüş gibi birçok alanda yapay zekâ uygulamaları geliştirilmektedir. İnsanın görme yeteneğinden esinlenerek üretilen evrişimsel sinir ağları (ESA) görüntü sınıflandırma, doğal dil işleme, nesne tanıma gibi birçok alanda başarı ile uygulanmaktadır. Özellikle büyük veri setleri üzerinde ESA kullanılarak geliştirilen yapay zekâ modellerinde sınıflandırma başarısı dikkat çekmektedir (Onan, 2020). Nesne tespiti görüntüler üzerinde yer alan canlı-cansız varlıkların sınıflandırılması ve görüntü üzerinde konumlarının kutucuklar ve maskeler ile belirtilmesi işlemidir. Nesne tespitinde YOLO, R-CNN, Maske R-CNN, SSD - MobileNet yöntemleri kullanılmaktadır (Saygılı & Özmen, 2022).

3.2.1. ESA

ESA' lar genel olarak özellik çıkartma, sınıflandırma, nesne tanıma ve görüntü segmentasyon işlemlerinde kullanılmaktadır (İnik & Ülker, 2017). ESA görüntü dosyaları üzerinde özniteliklerin elde edilmesi için görüntüleri sayısallaştırarak ızgara filtreleri ile işleyebilen derin öğrenme yöntemidir (Onan, 2020). Fotoğraf ve video görüntüleri ESA'lara matris formatında 3 ve 1 kanallı olarak verilmektedir. 1 kanallı görüntüler siyah beyaz görseller için kullanılırken 3 kanallı görüntüler renkli görseller için kullanılmaktadır. Şekil 3.8 ' de evrişim katmanında ızgara filtreleri uygulanarak görüntünün öznitelikleri çıkarılmaktadır. Örnek olarak uygulanan filtreler 3x3 , 5x5'lik matrisler halinde seçilebilir. Filtrenin tüm görüntü üzerinde uygulanması sonunda elde edilen matris öznitelik haritası olarak isimlendirilmektedir (Güldemir, 2022).



Şekil 3. 8. ESA yapısı.

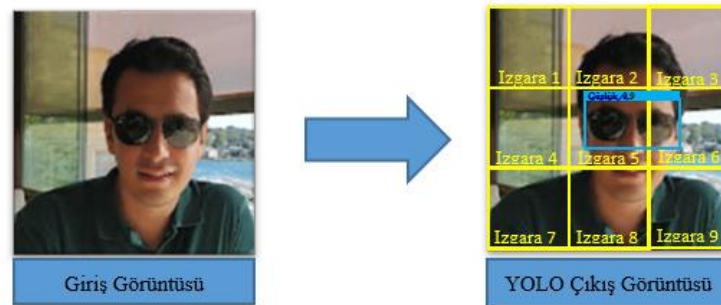
Özniteliklerinin çıkarılmasından sonra ortaklama katmanında elde edilen matrisi belirlenen filtrelerle göre tekrar küçültülür. Genel olarak ortaklama katmanında maksimum ortaklama ve ortalama ortaklama işlemleri gerçekleşir. Burada maksimum ortaklama uygulanan filtreler vasıtasıyla filtrenin uygulandığı matriste en büyük değerlerden yeni bir matris elde edilir. Ortalama ortaklamada filtrenin kapsadığı alanda tüm değerlerin ortalaması alınarak yeni bir matris elde edilmektedir. Tam bağlantı katmanında matris halinde ortaklama katmanındaki veriler vektörel hale dönüştürülmektedir (İnik & Ülker, 2017).

3.2.2. Nesne Algılama

Nesne tespiti, makinelerin görüntüler üzerinde nesnelerin konumları ile birlikte sınıflandırarak tespit edebilen teknolojilerdir. Nesne algılama günümüzde güvenlik, askeri, ulaşım ve tıp alanı gibi birçok alanda kullanılmaktadır (Jiao ve arkadaşları, 2019). Dünyada ve ülkemizde marketlerde ürün sayımı, mağazalarda reyon yoğunluk ısı haritalarının belirlenebilmesi için nesne algılama teknolojileri kullanılmaktadır (Xu ve arkadaşları, 2020;URL 8, 2022).

3.2.2.1. YOLO

Sadece Bir Kez Bak (YOLO) günümüzde çokça kullanılan nesne algılama algoritmasıdır. Hız konusunda oldukça iddialı bir algoritmadır. Hızlı olmasının altında yatan neden giriş olarak verilen görüntünün belirlenen ağ yapısından sadece bir kez geçirilmesinden kaynaklanmaktadır. YOLO algoritması nesnelerin sınıflandırma ve konum tespiti yapıldıktan sonra nesnenin algılandığı alanı çerçeve içine almaktadır. YOLO algoritması kendisine verilen görüntüyü SxS şeklinde ızgaralara bölmektedir. Izgaralara ayrılmış bölgelerde nesne olup olmadığına baktıktan sonra nesnenin merkez konumuna göre nesnenin yükseklik ve genişliğine göre tespit çerçevesi çizmektedir (URL 5, 2021). Şekil 3.9 'da örnek bir görüntü üzerinde YOLO derin öğrenme algoritmasının 3x3 ızgara yapısı görülmektedir.

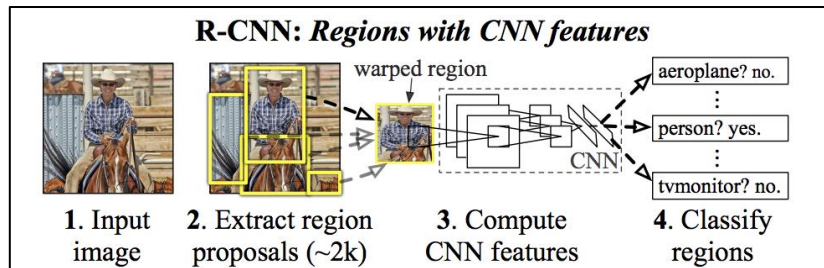


Şekil 3. 9. YOLO ızgara yapısı ve tespit çerçevesi.

Giriş olarak verilen görüntü ızgaralara ayrılmış her bir ızgarada gözlük nesnesinin var olup olmadığına bakılmıştır. 5 ve 6 nolu ızgaralara denk gelen alanda %90 oranında gözlük nesnesi algılanarak tespit çerçevesi mavi renk olarak çizilmiştir. YOLO tek bir ESA ile aynı anda birden çok sınırlayıcı çerçeve ve bu çerçeve kutuları için sınıf olasılıklarını tahmin etmektedir. İlk YOLO modeli Joseph Redmon tarafından 2016 yılında yayınlanan “You Only Look Once: Unified, Real-Time Object Detection” isimli makale ile tanıtılmıştır. Gerçek zamanlı olarak oluşturulan YOLO ağının saniyede 45 kare işleyebildiğini ifade edilmiştir. DPM (Deformable Parts Model) ve R-CNN (Region Based Convolutional Neural Networks) algoritmaları ile karşılaştırıldığında nesne algılamada daha iyi performans göstermektedir (Redmon ve arkadaşları, 2016).

3.2.2.2. R-CNN

ESA’ lar ile bir görüntü üzerinde sınıflandırma işlemi başarıyla yapılabildiği halde görüntüler üzerinde birden fazla nesne algılama gerçekleştirememektedir. Bu eksikliği gidermek için Bölge Bazlı Evrişimsel Sinir Ağları (R-CNN) geliştirilmiştir. R-CNN yapısında görüntü üzerindeki sınıfları ve nesnelere ait çerçeve kutularını tespit etmek amacıyla kullanılmaktadır. Modelde her bir bölge teker teker ESA’ dan geçirilmektedir. Her bir görüntüde yaklaşık olarak 2000 bölgede işlem yapıldığı için yavaştır. R-CNN ’nin bu yönden yavaş olmasından dolayı Hızlı Bölge Bazlı Evrişimsel Sinir Ağları (Fast R-CNN) geliştirilmiştir. Hesaplama hızı ve mAP değeri açısından R-CNN ve Fast R-CNN’den daha başarılı olan Faster R-CNN öne çıkmaktadır. Şekil 3.10’ da R-CNN mimarisi görülmektedir. Bu mimaride giriş olarak verilen görüntüde 2. aşamada yaklaşık olarak 2000 bölge önerisi oluşturulmaktadır. Önerilen bölgeler teker teker ESA’dan geçirilerek öz nitelikler oluşturulmaktadır. Oluşturulan öz nitelikler Destek Vektör Makineleri (SVM) ile sınıflandırılırken regresyon modelleri ile çerçeve kutuları (bounding box) belirlenmektedir (Serkan, 2021).

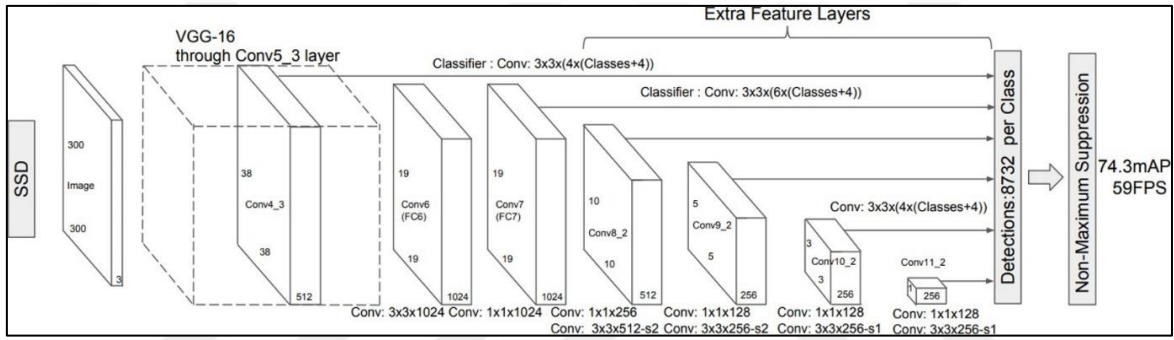


Şekil 3. 10. R-CNN mimarisi (Girshick ve arkadaşları, 2014).

3.2.2.3. SSD

Tek Atış Dedöktörü (SSD) derin öğrenme algoritması isminden de anlaşılacağı üzere tek bir derin öğrenme ağı kullanarak görüntüler üzerinde nesne tespiti yapmaktadır. SSD'nin Faster R-CNN'e göre daha hızlı çalıştığı görülmüştür (Liu ve arkadaşları, 2016). SSD Öznitelik haritalarının çıkarılması ve nesneleri algılamak için evrişim filtrelerinin uygulanması olarak iki aşamada gerçekleşmektedir (Tan ve arkadaşları, 2021).

Şekil 3.11' de SSD mimarisinde birinci aşamada 300x300 çözünürlükte giriş görüntüsü VGG-16 ESA ağından geçirilerek öz nitelikleri çıkarılmaktadır. İkinci aşamada öz nitelik matrislerinde sınırlayıcı kutular ile nesnenin varlığı incelenmektedir. Her sınırlayıcı kutu ve sınıf için olasılık puanları hesaplanmaktadır (Eriş, 2018).



Şekil 3. 11. SSD mimarisi (Liu ve arkadaşları, 2016).

Literatürde SSD ile yapılan bir çalışmada Faster R-CNN ile karşılaştırıldığında SSD derin öğrenme ağının daha hızlı çalıştığı görülmüştür. Aynı çalışmada R-CNN ve Fast R-CNN ağlarının ilk etapta nesneleri sınıflandırdıktan sonra ESA ağından geçirmesinden dolayı yavaş çalıştığı görülmüştür (Tan ve arkadaşları, 2021).

4. MATERYAL VE METOT

Bu bölümde tez çalışmasında kullanılan veri seti ve kullanılan yöntemler detaylı olarak sunulmuştur.

4.1. Materyal

Tez çalışmasında yer alan veri seti internetten ücretsiz olarak erişilebilen veri setleri ve çevreden alınan görüntülerden oluşturulmuştur (Cabani ve arkadaşları, 2021; URL 2, 2020). Kullanılan veri seti Şekil 4.1’ de JPG ve PNG formatında görüntülerden oluşmaktadır. Veri setinde yer alan maskesini doğru kullanan, maskesini yanlış kullanan ve maske kullanmayan görüntüler etiketlenmiştir. MKS sistemi için Şekil 4.1’ de bir bölümü görülen 1077 görüntü Label Image yazılımı ile etiketlenmiştir.



Şekil 4. 1. Veri setinde bulunan sınıflara ait görüntülerden bir bölüm.

Etiketlenen veri setine ait sınıf sayıları Tablo 4.1’ de gösterilmektedir. Veri setinde yer alan görüntülerin %70’i eğitim verisi için, %20’si doğrulama verisi için, %10’u ise test verisi için ayrılmıştır. YOLO ağı giriş çözünürlük değeri olarak 416*416 yükseklik ve genişlik değerine göre tüm resimler yeniden boyutlandırılmıştır.

Tablo 4. 1. Veri setinde yer alan 3 sınıfa ait görüntü sayıları.

Görüntü sayısı	Maske Kullanmayanlar	Maskesini Doğru Kullanan	Maskesini Yanlış Kullananlar	Toplam
Eğitim seti %70	200	265	289	754
Doğrulama seti %20	58	76	83	217
Test seti %10	28	38	40	106
Toplam	286	379	412	1077

Veri setinde Model 1 ve Model 2 olarak isimlendirilen yapay zekâ modellerinin oluşturulmasından sonra, veri setinin eğitim için ayrılan %70' lik kısmı yatay yönde aynalama veri çoğaltma tekniği ile 3 katına çıkarılarak Model 3 ve Model 4 olarak isimlendirilen yapay zekâ modelleri oluşturulmuştur.

4.2. Metot

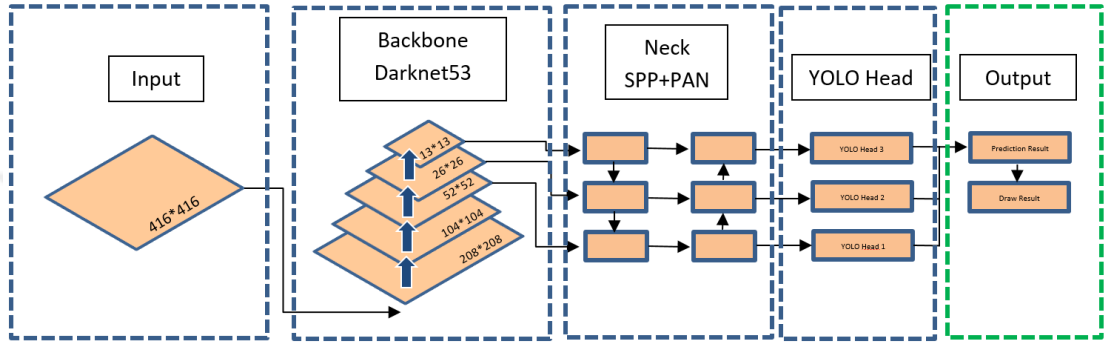
Bu bölümde tez çalışmasında kullanılan metotlar ve kullanılan donanımlar açıklanmaktadır. Tez kapsamında oluşturulan yapay zekâ modellerinde derin öğrenme tekniklerinden YOLO' nun 4. sürümlerine ait YOLOv4-tiny ve YOLOv4 kullanılmıştır. YOLOv4-tiny algoritması YOLOv4'e göre kıyasla katman sayısı ve modüller değiştirilerek düşük donanıma sahip cihazlar için daha yüksek FPS değerleri elde etmek için geliştirilmiştir (Jiang ve arkadaşları, 2020).

Çalışmada oluşturulan veri seti üzerinde YOLOv4 ve YOLOv4-tiny algoritmaları ile Google Colab web ortamında Tesla T4 GPU donanımı ile 4 adet yapay zekâ modeli elde edilmiştir. Çalışmada Model 1, Model 3 ile ifade edilen modeller YOLOv4-tiny ile geliştirilirken, YOLOv4 ile geliştirilen modeller Model 2 ve Model 4 olarak isimlendirilmiştir.

4.2.1. YOLOv4

Alexey Bochkovskiy, Chien -Yao Wang ve Hong-Yuan Mark Liao tarafından oluşturulan YOLOv4 performans açısından EfficientNet' ten 2 kat daha fazla hızlıdır (URL 5, 2021). Şekil 4.2 'de yer alan YOLOv4 mimarisinde görüldüğü gibi Input bölümünde 416*416 boyutundaki görüntüler giriş verisi olarak kullanılmaktadır. Backbone bölümünde

ise giriş olarak verilen görüntülerde özellik çıkarım işlemleri gerçekleşmektedir. Neck bölümünde daha fazla bilgi çıkarımları gerçekleştirilerek modelin daha iyi tahmin yapabilmesi gerçekleştirilmektedir. Neck bölümü PAN ağı ile SPP katmanının birleşiminden oluşmaktadır. Neck bölümünde yer alan PAN ağı daha fazla özellik toplamasını gerçekleştirirken, SPP havuzlama katmanı PAN ağı ile çıkarılan özelliklerin boyutlarından bağımsız olarak sabit boyutlu olmasını sağlamaktadır. YOLO Head bölümünde tespit edilen nesnelere ait sınırlayıcı kutuların koordinatları, güven skorları ve sınırlayıcı kutuların sınıf tahmin işlemleri gerçekleşmektedir (Lin ve arkadaşları, 2021).



Şekil 4. 2. YOLOv4 mimarisi (Lin ve arkadaşları, 2021).

4.2.2. YOLOv4-tiny

YOLOv4-tiny algoritması YOLOv4 algoritmasında yer alan 2 CSPBlock modülü yerine ResNet-D ağına 2 adet ResBlock-D modülü kullanılması ile elde edilmiştir. Bu şekilde hesaplama karmaşıklığı azaltılmak hedeflenmiştir. YOLOv4-tiny algoritmasında, YOLOv4 algoritmasında kullanılan CSPDarknet53 ağı yerine Backbone bölümünde CSPDarknet53-tiny ağını kullanılmaktadır. Yolov4-tiny ağı YOLOv4 ağına göre ağ boyutu ve parametreleri açısından büyük ölçüde azaltılmıştır (Jiang ve arkadaşları, 2020). YOLOv4-tiny algoritmasını kısaca YOLOv4'ün sıkıştırılmış hali olarak ifade edebiliriz. Bu ağ daha çok mobil ve gömülü sistemlerde hız problemlerini aşmak için tercih edilmektedir. FPS açısından kıyaslandığında YOLOv4-tiny YOLOv4'e göre 8 kat daha hızlıdır (URL 5, 2021).

4.2.3. MKS Robotu

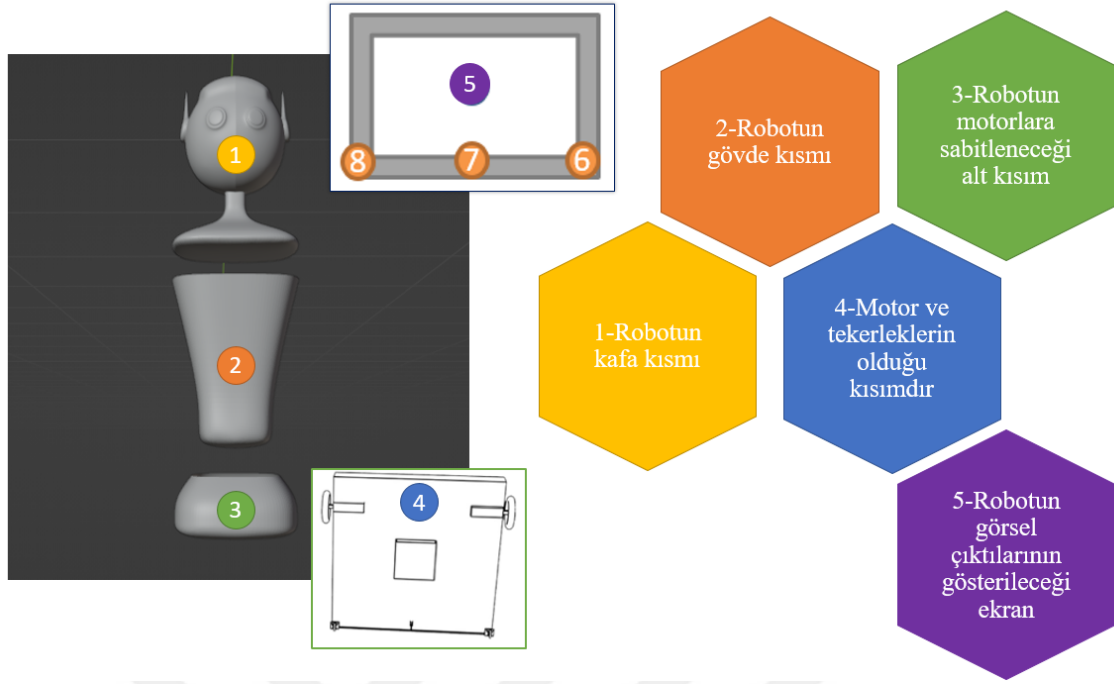
MKS yazılımının hareketli olarak çalıştırılabilmesi için 85x27 cm ebatlarında hareket edebilen bir robot geliştirilmiştir. Robotun dış kabuğu mevcut 3D yazıcının çıktı boyutunun 40x30x30 cm olmasından dolayı robot kabuk tasarımı 3 parçaya ayrılarak elde edilmiştir. 3D yazıcı ile çıktı alınırken 1,75 mm PLA filament kullanılmıştır. Elde edilen dış kabuklar sprey boya ile boyanarak metal görünüm verilmiştir. Robotun kafa ve gövde birleşim yerleri

için 3D yazıcıdan kuşak kabuğu elde edilmiştir. Elde edilen kuşak kuvvetli yapıştırıcı, sıcak silikon ve vidalar ile kafa kısmı kuşak kısmına sabitlenmiştir. Robotun uzaktan kontrolü Android işletim sistemleri için hazırlanan Bluetooth RC Car isimli uygulama ile yapılmaktadır (URL 6, 2021). Mobil telefon üzerinden gönderilen hareket komutları bluetooth modülüne bağlı mikrodenetleyiciye gelmekte daha sonra motor sürücü üzerinden robotun altında yer alan motorlara iletilmektedir. Hastalardan nabız, ateş ölçümü yapabilmek için robotta GY-MAX30100 kalp/nabız sensörü, MLX90614 sıcaklık sensörü kullanılmıştır.

Robotta kullanılacak yapay zekâ algoritmaları ve sensör okuma algoritmalarının hazırlanmasında; açık kaynaklı kütüphanelere sahip olması ve seçilen mini bilgisayarda hızlı çalışması özelliklerinden dolayı python dili tercih edilmiştir. Motorların kontrolleri için L298N motor sürücü kullanılmasına karar verilmiştir. L298N motor sürücü robotun hareketini sağlayacak olan motorları sürebilecek yeterlilikte olması ve python kütüphane desteği olmasından dolayı tercih edilmiştir. Şekil 4.3' te MKS robotunun blender yazılımındaki tasarımı ve genel görünümü görülmektedir. Robotun hareketi sırasında etrafındaki nesnelere çarpmaması için robotun gövdesinin ön, sağ ve sol bölümlerine 3 adet HC-SR04 ultrasonik mesafe sensörü yerleştirilmiştir.

Şekil 4.3' te gösterilen robot tasarımı 5 temel parçadan oluşmaktadır. Bu parçalar ve anlamları aşağıda özetlenmiştir:

1. Robotun kafa kısmı,
2. Robotun gövdesi,
3. Robotun motorlara sabitlendiği alt kısım,
4. Robotun hareketini sağlayan motor ve tekerleklerin olduğu kısımdır,
5. Robotun görsel çıktılarının gösterildiği ekran yeri ve sensörlerin takıldığı bölümdür.



Şekil 4. 3. MKS Robotunun bölümleri ve genel görünümünün tasarımı.

Şekil 4.3' te 5 numaralı kısım 1 numaralı kısım üzerine monte edilmiştir. Robotun maske kontrolünü yapmasını sağlayacak olan kamera 7 (turuncu renkli) numaralı alana, ateş ölçüm sensörü 8 numaralı alana, oksijen satürasyon ve nabız ölçümü yapan donanım 6 numaralı alana yerleştirilmiştir. 4 numara ile robotun taşıyıcı alt bölümü gösterilmiştir. Robotun mini bilgisayarı ve mikrodenetleyici kısmı 2 numaralı alan olan gövdeye yerleştirilmiştir. Sistemin enerjisini verecek güç kaynağı 3 numara ile gösterilen alt bölüme yerleştirilmiştir. Sesli iletişim için giriş donanımı olan mikrofona 7 numaralı alan üzerinde kameranın yan kısmına yerleştirilmiştir. Sesli uyarılar ve dönütler için kullanılacak hoparlörler 2 numara ile gösterilen gövdenin her iki yanına yerleştirilmiştir.

4.2.3.1. MKS Yazılımı

MKS yazılımı hareketli maske kontrolü ve sabit maske kontrolü için tasarlanmıştır. COVID-19 gibi virüs salgınlarında maske kontrolü yapılacak sabit noktalar veya hareketli robotlar üzerinde MKS maske kontrolünü 3 sınıfa göre yapabilmektedir. Günümüzde salgına bağlı olarak gerçekleştirilen maske kontrol noktalarında insana ihtiyaç duymadan MKS yazılımının sabit veya hareketli kullanımı ile maske kontrolü gerçekleştirilmesi hedeflenmiştir. MKS yazılımı python dili ile hazırlanmış yapay zekâ destekli bir yazılımdır. MKS' de yapay zekâ modeli olarak 4 adet model önerilmektedir. Model 1, Model 2, Model 3 ve Model 4 olarak isimlendirilen yapay zekâ modellerinde GPU ve CPU'su yüksek

cihazlar için Model 2 ve Model 4 önerilirken, düşük donanıma sahip bilgisayarlarda Model 1 ve Model 3' ün kullanılması önerilmektedir. Geliştirilen yapay zekâ modelleri YOLO algoritmaları ile eğitilmiş ve mini bilgisayar üzerinde performanslı çalıştırılabilmesi için Tensorflow-Lite araç setleri kullanılarak çalıştırılmıştır (David ve arkadaşları, 2021).

4.2.3.2. MKS Robot Tasarımı ve Çıktısı

MKS robot tasarımı 3 boyutlu Blender yazılımında tasarlanmıştır. Blender gelişmiş özellikleri ve kolay kullanımının yanında ücretsiz ve açık kaynaklı bir yazılım olmasından dolayı tercih edilmiştir. Blender yazılımında tasarlanan model 3 boyutlu yazıcıdan çıktı alınabilmesi için 3 parça halinde stl dosya formatına dönüştürülmüştür. 3 parça halinde elde edilen stl dosya formatındaki dosyalar Creality Slicer programında Gcode dosya formatına dönüştürülerek Creality Cr-10s Pro marka 3d yazıcıdan çıktı alınmıştır.

4.2.3.3. MKS Robot Hareket Bölümü

MKS robotunun hareketini Şekil 4. 3 bölümünde gösterilen 4 numaralı bölümde yer alan 2 adet DC motor ve 2 adet çok yönlü tekerleklerle sağlanmıştır. Motorların kontrol edilebilmesi için 1 adet L298N motor sürücü kullanılmıştır. Motor sürücünün kontrolü Arduino Uno model mikrodenetleyici ile sağlanmıştır. Robotun kontrolü Mikrodenetleyiciye bağlantısı yapılmış HC-06 Bluetooth modülü ile kablosuz olarak kontrol gerçekleştirilmektedir. Şekil 4.4' te çalışmada kullanılan motor sürücü görülmektedir.



Şekil 4. 4. L298N motor sürücü.

L298N Motor sürücünün çalışmada kullanılma nedeni piyasada kolayca bulunabilmesi, maliyet ve çalışmada kullanılan mikrodenetleyiciye ait kütüphane desteğinin olmasıdır. L298N motor sürücüsünün datasheet dosyası incelendiğinde 2 adet doğru akım

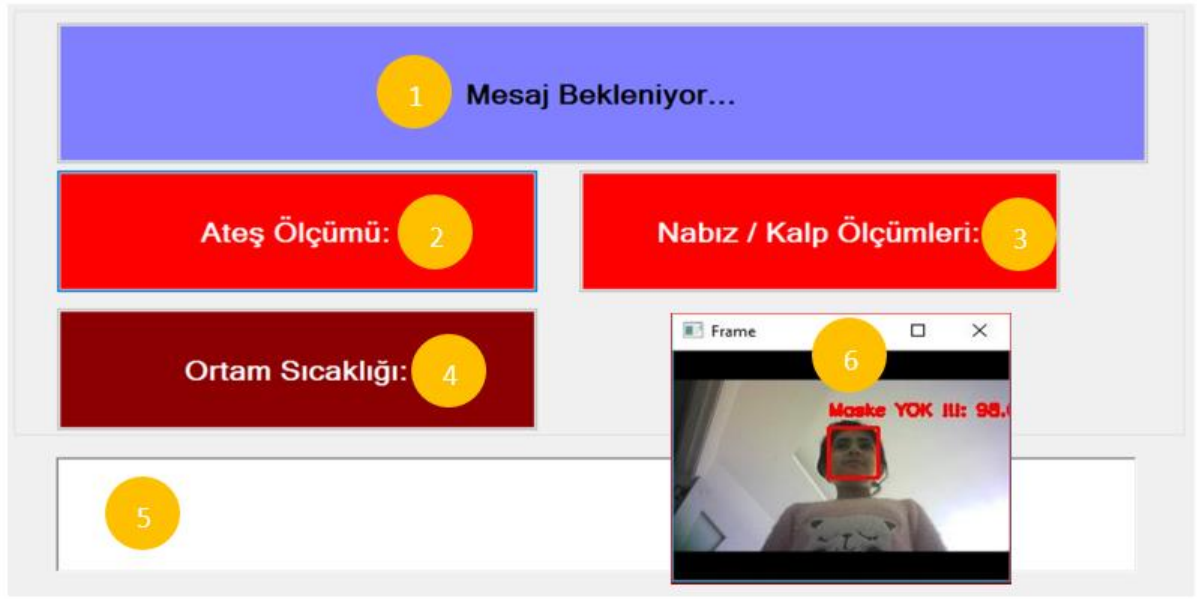
motorunun bağımsız olarak sürebildiği, her bir kanaldan 2 ampere kadar sürekli çıkış akımı verebildiği, giriş besleme geriliminin 5V ile 35V değerleri arasında olduğu görülmüştür (URL 7, 2021). Robotun hareketi için 2 adet 6V 185 rpm DC motor kullanılmıştır. Robotun hareketini kolaylaştırmak ve robotun dengesini sağlayabilmek için 2 adet 20 mm yükseklikte çok yönlü tekerlekler kullanılmıştır. Şekil 4.5’ te MKS robotunda kullanılan DC motor görülmektedir.



Şekil 4. 5. 6V 185 rpm DC motor.

4.2.3.4. MKS Robot Ekranı

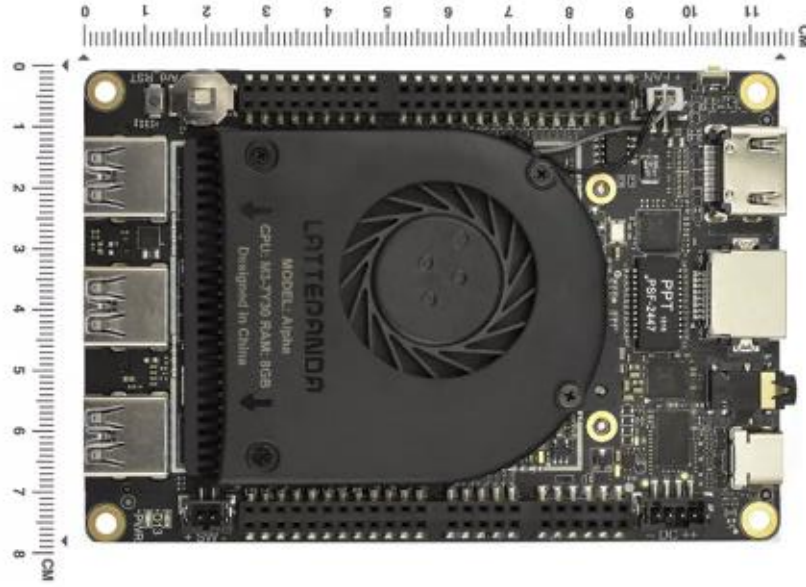
Robotun görsel çıktılarının kullanıcılara iletilmesi için 10.1 inch boyutunda WaveShare marka LCD ekran kullanılmıştır. Bu ekran mini bilgisayara hdmi portundan bağlanmıştır. Ekranın sabitlenmesi ve dış darbelerden korunması için strafor malzemeden çerçeve hazırlanmıştır. Bu çerçeve aynı zamanda kamera, ateş ölçme sensörü, mikrofon ve satürasyon sensörlerini üzerinde konumlanmasını sağlamaktadır. MKS Robot ekranında Şekil 4.6’ da gösterilen C# dilinde hazırlanmış olan yazılım ile robot tarafından üretilen geri bildirimler 1 numaralı alanda, kişilere ait anlık ateş ölçümü 2 numaralı alanda, oksijen satürasyonu ve nabız bilgileri 3 numaralı alanda, ortam sıcaklığı 4 numaralı alanda gösterilmektedir. 5 numaralı alanda robot tarafından üretilen log kayıtları gösterilirken, 6 numaralı alanda anlık maske kontrolüne ait tahmin sonucu gösterilmektedir.



Şekil 4. 6. MKS robot bilgilendirme yazılımı arayüzü.

4.2.3.5. MKS Robotu Mini Bilgisayarı

MKS Robotunun yapay zekâ destekli yazılımlarının çalıştırılabilmesi için gerekli donanım ihtiyacını karşılaması, sensörlerden gelen bilgilerin okunup istenilen formatlara dönüştürülme imkânının olması ve kart büyüklüğünün küçük boyutlarda olmasından dolayı Latte Panda Delta mini bilgisayar tercih edilmiştir. Mini bilgisayar üzerinde yer alan Arduino Leonardo model mikrodnetleyicinin tümleşik olarak kullanıcılara sunulması sensörlerden gelen bilgilerin kolayca seri port üzerinden okunmasını sağlamaktadır. Seri port üzerinden mikrodnetleyiciden bilgilerin alınması için arduino kütüphanelerinden Standartfirmdata kullanılmıştır. Şekil 4.7' de mini bilgisayarın yapısı ve boyutu görülmektedir. Mini bilgisayar üzerinde yer alan mikrodnetleyici sürekli olarak arka planda çalıştığı için sensörlerden gelen bilgiler sesli asistan ve MKS yazılımı ile eş zamanlı olarak görev yapabilmektedir. Bunlara ek olarak Latte Panda Delta mini bilgisayarda kullanımı kolay olan Windows işletim sisteminin kullanılabilir olması mini bilgisayar seçiminde belirleyici olmuştur.



Şekil 4. 7. Latte Panda Delta mini bilgisayar yapısı.

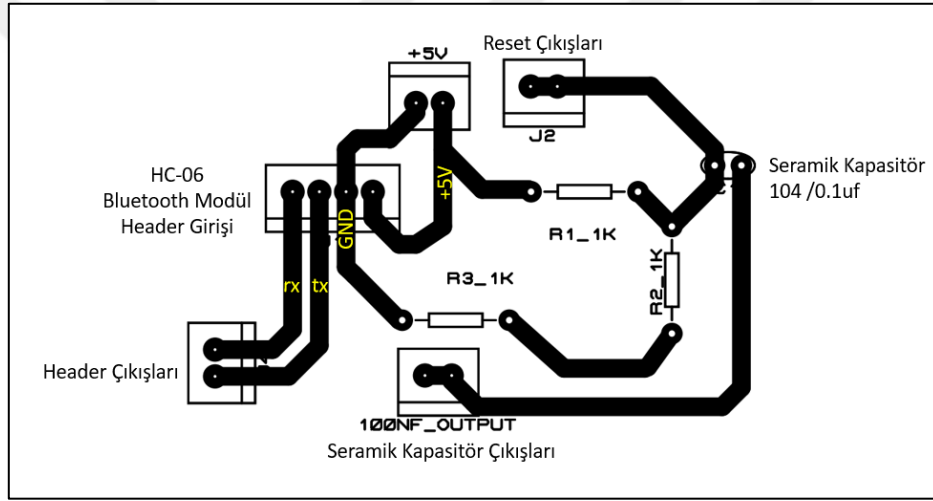
Latte Panda Delta mini bilgisayarı 45 Watt'lık bir güç beslemesine ihtiyaç duymaktadır. Bu güç 4 adet 6000 mAh 3.2V LifePO4 pil ile sağlanmıştır. Mini bilgisayarın harici besleme bölümünden güç verilmiştir. LifePO4 pilin uzun ömürlü kullanılabilmesi ve pillerin şarj edilebilmesi için batarya yönetim sistemi devresi kullanılmıştır.

4.2.3.6. Sesli Asistan

Robotun insanlar ile sesli iletişime geçerek hasta kayıt, yönlendirme ve uzman kişilere soru yönetebilmesi için python SpeechRecognition kütüphanesinden sesli asistan geliştirilmiştir. Robotun sesli olarak üzerinde yer alan hoparlör donanımları ile “sizlere nasıl yardımcı olabilirim” mesajını iletmesinin ardından bip sesi ile robot sesli komut beklemektedir. Sesli asistan yazılımında verilen sesli komutlardan muayene, sesli sorular, yönlendirme işlemlerine göre insanları yönlendirmektedir. Eğer kayıt yapmak isteniyorsa robot kişinin adını soyadını ve T.C. numarasını onaylı olarak almakta ve veri MSSQL Server veri tabanına kaydedilmektedir. Eğer robota kişi sesli olarak soru sormak veya danışmak isterse sesli asistan üzerinde soru ile ilgili kişi bulunarak e-mail ile anında soru iletilmekte 2 dakika içinde kişinin soruya cevap vermesi ile gelen cevap robot vasıtası ile sesli olarak okunmaktadır. Robottan sesli olarak yönlendirme istendiğinde sesli asistanda yüklü olan yönlendirme bilgilerine göre robot sesli olarak kişileri yönlendirmektedir. Robot sesli mesajlarını gTTS (Google Text to Speech) kütüphanesi ile mp3 dosya formatında üretmektedir. Üretilen mp3 ses dosyaları pygame kütüphanesi ile seslendirilmektedir. Sesli asistan yazılımı MKS yazılımına paralel çalışan bir yazılımdır.

4.2.3.7. Kablosuz Programlama Devresi

MKS Robotun hareketini sağlayan mikrodnetleyici kartına ait yazılımın kablosuz olarak programlanabilmesi için Şekil 4.8’ de pcb baskı devresi gösterilen elektronik bir devre kartı geliştirilmiştir. Bu devre ile robotun hareketini sağlayacak kodların kablo ile güncelleme zorunluluğunun ortadan kaldırılması ile daha pratik güncelleme işlemleri gerçekleştirilmiştir. Şekil 4.8’ deki seramik kapasitör çıkışı HC-06 bluetooth modülünün 32. pinine bağlanmıştır. Reset çıkışı ise mikrodnetleyicinin reset pinine bağlanmıştır. Devrede reset ve seramik kapasitör çıkışlarında ikişer adet çıkış konulma sebebi yedek çıkışların olmasından kaynaklanmaktadır. Devre blutetooth üzerinden aldığı bilgileri seri bağlantı ile mikrodnetleyiciye aktarmakta ve reset pini üzerinden mikrodnetleyiciyi yeniden başlatmaktadır (Shubham, 2018).



Şekil 4. 8. Kablosuz mikrodnetleyici programlayıcı devresi.

Şekil 4.8’ de yer alan header çıkışları bluetooth modülüne ait alıcı ve verici bağlantılarının kablo ile mikrodnetleyiciye bağlantısını sağlamaktadır. +5V etiketli alan ise kablosuz programlayıcı devresinin enerji besleme giriş bölümünü işaret etmektedir.

4.2.3.8. Pulse Oksimetre

Sağlık kuruluşlarında hastaların el ve ayak tırnağı üzerine Pulse Oksimetrenin ışık kaynağı yerleştirilerek hastanın oksijen satürasyon değeri hesaplanabilmektedir (Hakverdioğlu, 2007). MKS robotu üzerinde kişilerin kırmızı kan hücrelerinin taşıdığı oksijen miktarını gösteren oksijen satürasyon değeri, Şekil 4.9’ da gösterilen GY-MAX30100 sensörü ile elde edilmiştir. Ayrıca sensör üzerinden nabız bilgileri de alınmaktadır.



Şekil 4. 9. GY-MAX30100 sensörü.

Sensörde yer alan kızıl ötesi ve kırmızı ışığın kandaki oksijen miktarına göre emiliminden kaynaklanan yansıma oranına göre oksijen saturasyonunu ölçmektedir. Şekil 4.9 ‘ da gösterilen GY-MAX30100 sensöre ait veriler mini bilgisayar üzerinde bütünleşik olarak yer alan mikrodenetleyici üzerinden alınmaktadır. GY-MAX30100 sensörü MAX30100 kütüphanesi üzerinden kullanılmıştır. Kütüphane ile oksijen saturasyonu ve nabız bilgileri result nesnesi üzerinden alınabilmektedir. GY-MAX30100 sensörü 5V giriş beslemesine ihtiyaç duymaktadır.

4.2.3.9. MSSQL Server

Microsoft firması tarafından oluşturulan Windows işletim sistemlerinde çalışan bir veri tabanıdır. Web üzerinden ve yerel olarak çalıştırılabilen MSSQL Server birden fazla kullanıcının aynı anda erişebildiği ve ilişkisel veri yapılarının kullanıldığı bir veri tabanıdır. Veri tabanını oluşturmak ve tabloları yönetmek için SQL Server Management Studio kullanılmaktadır (URL 4, 2012). Bu çalışmada sağlık kuruluşlarında MKS robotu üzerinden sesli kayıt bilgilerinden oluşturulan hasta kayıt bilgileri MSSQL Server veri tabanına kaydedilmiştir.

5. DENEYSEL ÇALIŞMA

Bu bölümde metotlar bölümünde açıklanan Model 1, Model 2, Model 3 ve Model 4' e ait yapay zekâ modellerinin oluşturulması, yapay zekâ modellerinin çeşitli donanımlarda çalıştırılması ve tez ile önerilen robotta uygulanması ile elde edilen sonuçlar bulunmaktadır. Tablo 5.1' de YOLOv4, YOLOv4-tiny derin öğrenme algoritmaları ile oluşturulan Model 1, Model 2, Model 3 ve Model 4 için başarımleri metrikleri görülmektedir. YOLOv4-tiny algoritması kullanılan eğitimlerde sınıf başına eğitim için 2000 iterasyon belirlenirken, YOLOv4 için toplam 2000 iterasyon sayısı belirlenmiştir (AlexeyAB, 2018).

Tablo 5. 1. Geliştirilen yapay zekâ modellerine ait başarımleri değerleri.

Modeller	Hassasiyet (Precision)	Geri Çağırma (Recall)	F1- Skoru	Tespit Zamanı(ms)	IoU	mAP	Boyut (Mb)
Model 1	0,85	0,97	0,90	5-6	%71,27	%97,63	22,4
Model 2	0,95	0,98	0,96	20-21	%81,77	%98,09	244,2
Model 3	0,86	0,98	0,92	5-6	%72,13	%98,37	22,4
Model 4	0,93	0,98	0,95	20-21	%81,90	%98,11	244,2

Tablo 5.1' de Model 1, Model 2, Model 3 için tespit zamanları Google Colab Tesla T4 GPU' lu bilgisayar üzerinden hesaplanmıştır. Geliştirilen Modellere ait mAP ve IoU değerleri Tablo 5.1' de görülmektedir. Model 1 ve Model 3' ün test görüntülerinde 5-6 ms tahmin gerçekleştirebildiği, Model 2 ve Model 4' ün ise 21-22 ms tahmin işlemlerini gerçekleştirdiği görülmüştür.

Model 1'in maskesini yanlış kullanan sınıfı için %97,52, maskesini doğru kullanan sınıfı için %96,70 ve maske kullanmayan sınıfı için %98,67 AP değerleri elde edilmiştir. Sınıflara ait TP ve FP değerleri maskesini yanlış kullanan sınıfı için TP = 78 FP = 19, maskesini doğru kullanan sınıfı için TP = 55 FP = 9 ve maske kullanmayan sınıfı için TP = 50 FP = 5 değerleri elde edilmiştir.

Model 2' nin maskesini yanlış kullanan sınıfı için %97,34, maskesini doğru kullanan sınıfı için %97,39 ve maske kullanmayan sınıfı için %99,54 AP değerleri elde edilmiştir. Sınıflara ait TP ve FP değerleri maskesini yanlış kullanan sınıfı için TP = 76 FP = 1, maskesini doğru kullanan sınıfı için TP = 58 FP = 6 ve maskesini kullanmayan sınıfı için TP = 51 FP = 3 değerleri elde edilmiştir.

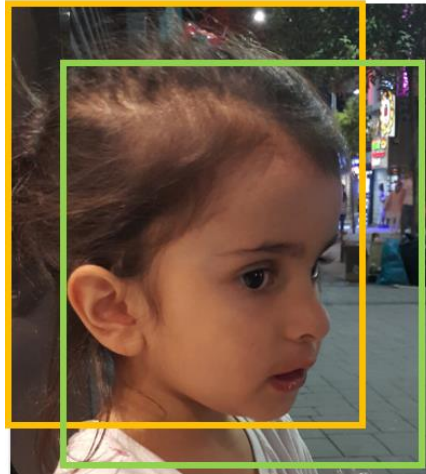
Model 3' ün maskesini yanlış kullanan sınıfı için %97,88, maskesini doğru kullanan sınıfı için % 98,07 ve maskesini kullanmayan sınıfı için % 99,15 AP değerleri elde edilmiştir. Sınıflara ait TP ve FP değerleri maskesini yanlış kullanan sınıfı için TP = 78 FP = 21, maskesini doğru kullanan sınıfı için TP = 58 FP = 4 ve maskesini kullanmayan sınıfı için TP = 50 FP = 6 değerleri elde edilmiştir.

Model 4' ün maskesini yanlış kullanan sınıfı için %98,73, maskesini doğru kullanan sınıfı için % 95,83 ve maske kullanmayan sınıfı için % 99,77 AP değerleri elde edilmiştir. Sınıflara ait TP ve FP değerleri maskesini yanlış kullanan sınıfı için TP = 77 FP = 2, maskesini doğru kullanan sınıfı için TP = 58 FP = 9 ve maskesini kullanmayan sınıfı için TP = 51 FP = 4 değerleri elde edilmiştir.

Derin öğrenme ile nesne tespiti gerçekleştiren yapay zekâ modellerinin başarı değerlendirilmesinde IoU ve mAP metrikleri kullanılmaktadır. IoU değeri Denklem 5.1' e göre hesaplanmaktadır (Parico & Ahamed, 2021).

$$IoU = \frac{\text{Gerçek çerçeve alanı} \cap \text{Tahmin edilen çerçeve alanı}}{\text{Gerçek çerçeve alanı} \cup \text{Tahmin edilen çerçeve alanı}} \quad (5.1)$$

Şekil 5.1 'de veri setinde sarı renkte belirtilen çerçeve (kutu) alanı maskesiz kişilerinin veri setinde yer alan görüntülerin etiketlendirilmesinde kullanılan alanı yani gerçek referans alanını (ground truth) belirtmektedir. Yeşil renkte belirtilen çerçeve alanı ise oluşturulan yapay zekâ modelinin nesne tahmin kutusunu göstermektedir.



Şekil 5. 1. Tahmin ve referans kutu alanları.

Bu çalışmada 3 sınıfa ait IoU değerleri %71,27, %81,77, %72,13, %81,90 olarak elde edilmiştir. Doğru pozitif sayıları, yanlış pozitif sayıları nesne tespiti yapan algoritmalarının

performanslarını belirleyen metriklerinin hesaplamalarında kullanılmaktadır. Çalışmada kullanılan IoU değerleri hesaplanırken eşik değeri olarak 0,5'e eşit veya büyükse TP olarak belirlenmiştir. IoU değeri 0,5 değerinin altında yer alırsa FP olarak kabul edilmiştir. Yanlış FN ise güven eşiği olarak bu çalışmada belirlenen 0,25 in altında IoU değerine sahip tahmin sonuçları olarak alınmaktadır.

TP, FP ve FN değerleri elde edildikten sonra nesne tespitinde kullanılan hassasiyet, geri çağırma, F1 skoru, ortalama hassasiyet ve genel ortalama hassasiyet değerleri 5.2 - 5.6 denklemlerine göre hesaplanmaktadır (Parico & Ahamed, 2021). AP değeri geliştirilen yapay zekâ modelinin bir sınıftaki doğruluğunu temsil ederken, mAP değeri ise tüm sınıfların ortalama doğruluğunu göstermektedir (Fu ve arkadaşları, 2021).

$$\text{Hassasiyet(P)} = \frac{\text{Doğru Pozitif (TP)}}{\text{Doğru Pozitif (TP)} + \text{Yanlış Pozitif (FP)}} \quad (5.2)$$

$$\text{Geri çağırma(R)} = \frac{\text{Doğru Pozitif (TP)}}{\text{Doğru Pozitif (TP)} + \text{Yanlış Negatif (FN)}} \quad (5.3)$$

$$\text{F1 Skoru} = \frac{2 * \text{Geri çağırma} * \text{Hassasiyet}}{\text{Geri çağırma} + \text{Hassasiyet}} \quad (5.4)$$

$$\text{Ortalama Hassasiyet (AP)} = \int_0^1 P(R) dR \quad (5.5)$$

$$\text{Genel Ortalama Hassasiyet (mAP)} = \frac{\sum_{i=1}^N AP_i}{N} \quad (5.6)$$

Oluşturulan yapay zekâ modellerinin AP değerleri Denklem 5.5 ile elde edilir. AP değeri veri setinde yer alan tüm sınıflara ait R, P eğrisinin alt bölümde kalan alanın hesaplanması ile elde edilmektedir. Genel ortalama hassasiyet değeri (mAP) Denklem 5.6 ile hesaplanır (Lippi ve arkadaşları, 2021).

Denklem 5.2 ' ye göre Model 1 için 3 sınıfın ortalama hassasiyet değeri hesaplanıp ortalaması alınırsa:

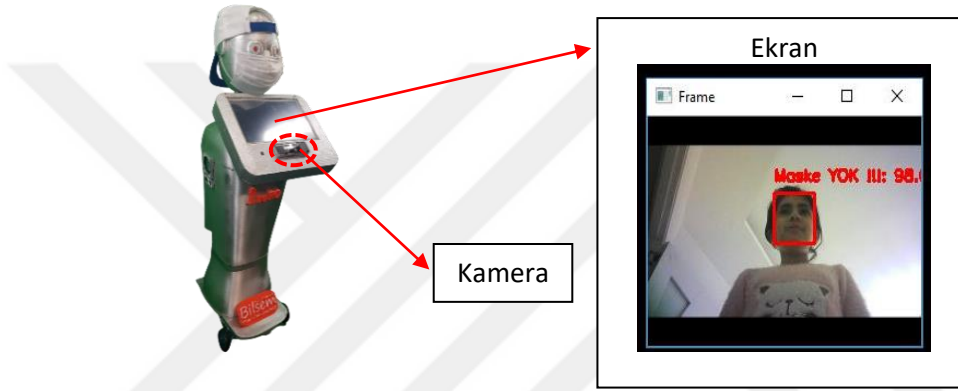
$$\text{Hassasiyet(P)} = \frac{TP}{TP+FP}$$

$$= \left(\frac{78}{78+19} + \frac{55}{55+9} + \frac{50}{55+5} \right) * \frac{1}{3} \cong 0,85 \text{ değeri elde edilmektedir. Bu değer Tablo 5.1' e}$$

bakıldığında algoritma tarafından üretilen hassasiyet değerinin aynı olduğu görülmektedir.

5.1. MKS Robotu ve Bulguları

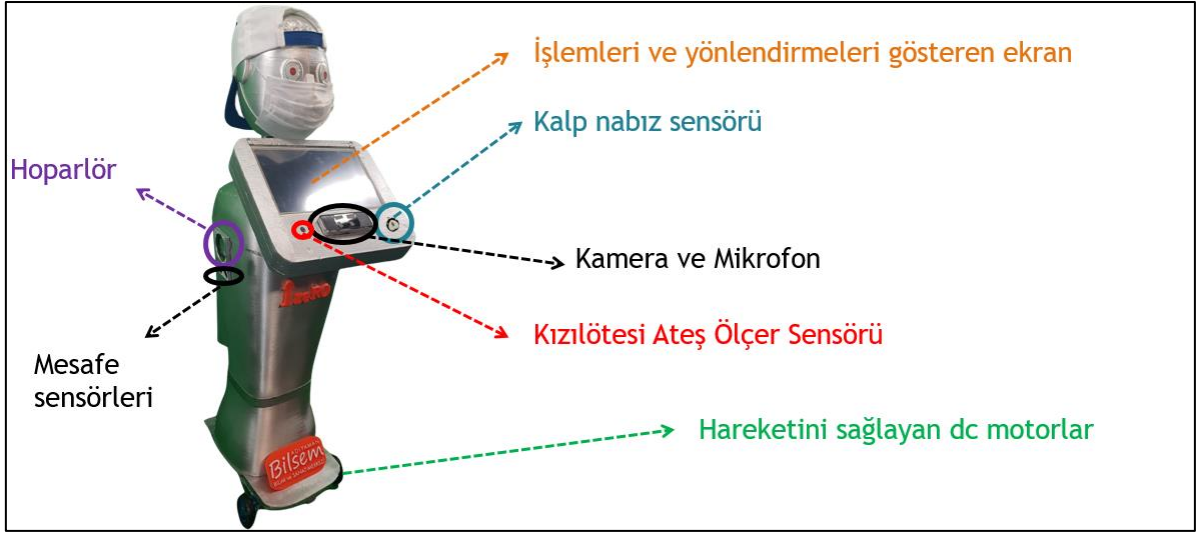
Tez çalışmasında önerilen MKS Robot prototipi Şekil 5.2' de görülmektedir. Robotun ön kısmında yer alan ekran bölümünün alt kısmında ortalanmış olarak kamera donanımı yer almaktadır. Kameradan alınan görüntüler geliştirilen yapay zekâ modellerine bütünleştirilmiş MKS yazılımından geçirilerek maske kontrolü gerçekleştirilmektedir.



Şekil 5. 2. Robot üzerinde MKS sisteminin çalışma çıktısı.

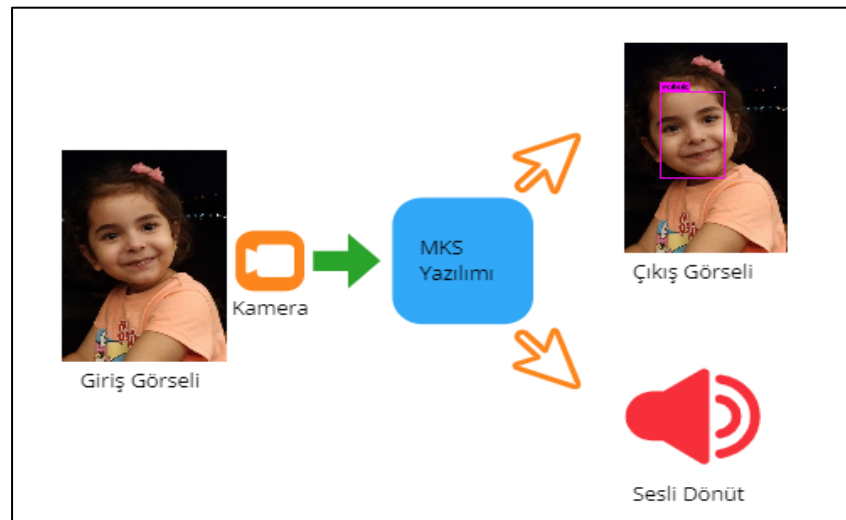
Maske kontrolü 3 sınıfa göre gerçekleştirildikten sonra robotun gövde kısmında sağ ve sol bölümünde yer alan hoparlör donanımları vasıtası ile sesli geri bildirimler kişilere iletilmektedir.

Robotun maske kontrolü sırasında sesli dönütler ile beraber robotun ön bölümünde yer alan 10,1 inch boyutundaki ekran ile kişilerin anlık olarak maske tahmin sonuçları gösterilmektedir. Şekil 5.3' te MKS prototipinin bölümleri ve görevleri görülmektedir.



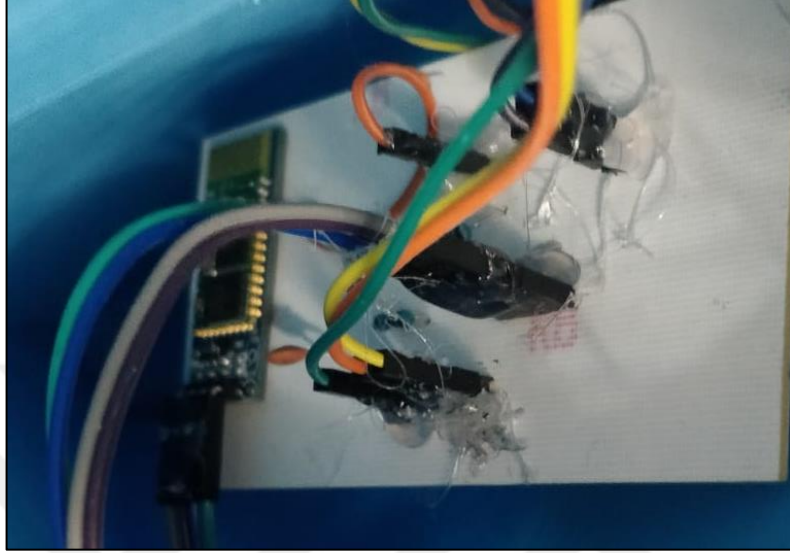
Şekil 5. 3. MKS robotunun görevleri.

Şekil 5.4 ‘ te MKS yazılımının çalışma şekli görülmektedir. Robotta yer alan mini bilgisayar üzerine bağlı kameradan alınan görüntüler üzerinde maske tespiti gerçekleştirilmiştir. MKS yazılımında sesli geri dönüt olarak maskesini kullanmayan kişilere “Lütfen maskenizi takınız” şeklinde uyarılar verilirken, maskesini doğru kullananlara “Maskenizi taktığınız için teşekkür ederim” sesli dönüt mesajı verilmiştir. Maskesini yanlış kullananlara “Lütfen maskenizi doğru takınız” şeklinde sesli uyarı mesajı verilmiştir. MKS robotuna yönlendirilen sesli olarak iletilen soru cümleleri sesli asistan yazılımı tarafından algılanarak soru sorulmak istenen kişiye soru e-mail ile yöneltmiş ve soruya verilen cevap robot tarafından sesli olarak okunmuştur. Ayrıca robota sesli olarak iletilen kayıt ve muayene isteklerinde robot kişinin T.C. no bilgisi ve adı ve soyadını alarak MSSQL Server veri tabanına pyodbc kütüphanesinden yararlanılarak kayıt işlemi gerçekleştirilmiştir.



Şekil 5. 4. MKS sisteminin çalışma şeması.

Şekil 5.5’ de MKS robotunun motor sürücüsünü denetleyen mikrodenetleyicinin kablosuz olarak programlanmasını sağlayan devrenin, bakır plaka üzerine baskı edilmiş ve bağlantıları yapılmış hali görülmektedir. Bu devre sayesinde MKS robotunun hareket kontrolünü sağlayan kodlar kablosuz olarak programlanmıştır.



Şekil 5. 5. Kablosuz programlayıcı devresi.

Tez kapsamında geliştirilen MKS Robotu’ nun tamamlanmış prototipi farklı açılardan Şekil 5.6’ da görülmektedir. Şekil 5.6.a-d’ de sırasıyla robotun ön profili, robotun gövdesine ait yan profili, robotun arka profili ve robotun üst çapraz profilinden çekilen fotoğraflar yer almaktadır.



(a)

(b)

(c)

(d)

Şekil 5. 6. MKS Robotu’ nun farklı açılardan görüntüleri.

5.2. Modellere Ait Bilgisayar Bulguları

Tez kapsamında üretilen yapay zekâ modellerinin MKS yazılımına bütünleştirilmesi sonucunda anlık olarak maske kontrolü gerçekleştirilerek sesli geri bildirimler oluşturulmuştur. Üretilen yapay zekâ modellerinin tahmin sonuçları Tablo 5.2’ de gösterilen donanımlarda elde edilmiştir. Tablo 5.3’ te yapay zekâ modellerinin tahmin sonuçları ve FPS değerleri tablo olarak listelenmiştir.

Tablo 5.2’ de Lenovo Z50-70, Nvidia Jetson Nano ve Google Colab Pc bilgisayarlarında GPU donanımı mevcut iken diğer donanımlarda GPU donanımı yer almamaktadır. Lenovo Z50-70 donanımı dizüstü bilgisayar iken diğer donanımlar mini bilgisayar olarak kategorize edilmektedir. Mini bilgisayarlar genel olarak Lenovo Z50-70 gibi diz üstü bilgisayarlar ve masaüstü bilgisayarlara göre daha az enerji tüketen bilgisayarlar olmakla birlikte boyutları masaüstü ve dizüstü bilgisayarlara göre daha küçüktür. MKS yazılımının mini bilgisayarlar ve dizüstü bilgisayarlarda kullanımı hedeflendiğinden birçok donanımda çalıştırılmıştır. Tablo 5.2’ de yer alan mini bilgisayarlardan Nvidia Jetson Nano diğer mini bilgisayarlara kıyas ile daha iyi performans göstermiştir. Nvidia Jetson Nano’ da diğer mini bilgisayarlarda bulunmayan GPU desteği bu farkı oluşturmaktadır. Ayrıca Raspberry pi 8 GB mini bilgisayarının ram donanım kapasitesinin Nvidia Jetson Nano’ ya göre 2 kat daha fazla olmasına rağmen daha düşük FPS değerlerde yapay zeka modelleri çalıştırdığı Tablo 5.3’de görülmektedir. Bu bağlamda yapay zekâ projelerinde Raspberry pi 4 GB – 8 GB mini bilgisayarlarına göre Nvidia Jetson Nano mini bilgisayarlarının robot çözümlerinde kullanılması önerilmektedir.

Tablo 5. 2. Çalışmada kullanılan donanımlar.

Donanımlar	Donanım Görseli	Donanım Özellikleri
Lenovo Z50-70		CPU: Intel i5 1.7 Ghz - 8Gb-1Tb 15.6"-4Gb Ram: 12 GB GPU: Nvidia Geforce GT840M 4GB
Google Colab PC		CPU: 2.30 GHz GPU: Tesla T4 GPU RAM: 16GB high-bandwidth memory (GDDR6)

Nvidia Jetson Nano		CPU: Quad-core ARM A57 @ 1.43 GHz RAM: 4GB 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: 8GB LPDDR4 -3200 GPU: Yok
Raspberry Pi 4- 4GB		CPU: 1.5 GHz Broadcom BCM2711 dört çekirdekli ARM Cortex-A72 RAM: 4GB LPDDR4-3200 GPU: Yok

MKS yazılımı maskesini doğru kullanan, maskesini yanlış kullanan ve maske kullanmayan kişilerin gerçek zamanlı tespitini yaparak kişileri sesli olarak uyarma işlemini yapabilmektedir. MKS yazılımının test edilmesi için Tablo 5.2’de belirtilen donanımlardan Latte Panda Delta mini bilgisayara Windows 10 Inspire işletim istemi, Nvidia Jetson Nano mini bilgisayara Ubuntu işletim istemi, Raspberry pi 4 GB – 8 GB mini bilgisayarlar rasbian işletim istemi, Lenovo Z50-70 dizüstü bilgisayara Windows 10 Pro. işletim sistemi yüklenmiştir. Tablo 5.2’ de yer alan donanımlara işletim sistemleri yüklendikten sonra güncelleme, paket yönetimi işlemlerinin daha hızlı yapılabilmesi için uzaktan kontrol programı olan VNC Viewer programı kurulmuştur. Bilgisayar donanımlarında geliştirilen yapay zekâ modellerinin çalışması için gerekli olay python alt yapısı ve kütüphaneleri kurulmuştur. Ayrıca robot üzerinde kullanılan Latte Panda Delta mini bilgisayar üzerinde

tümleşik olarak yer alan Arduino Leonardo mikrodnetleyicinin kodlanması ve seri iletişimi için gerekli Arduino kütüphaneleri ve sürücülerini yüklenmiştir. Latte Panda Delta mini bilgisayarında seri iletişim sağlanması için StandartFirmata Kütüphanesi indirilerek mikrodnetleyiciye yüklenmiştir.

MKS yazılımının Tablo 5.2’ de yer alan donanımlar üzerinde çalıştırılması ile elde edilen sonuçlar Tablo 5.3 ’ de görülmektedir.

Tablo 5. 3. Geliştirilen modellerin farklı donanımlardaki sonuçları.

Yapay Zekâ Modeli	Donanım	MKS Çıktısı	Sınıf	Doğruluk	Donanım Hızlandırıcı	FPS Minimum ve Maksimum Değerleri
Model 1	Raspbery Pi 4 – 4 GB		Maskesini yanlış kullanan	%95	CPU	0,56 - 1,48
Model 2	Lenovo Z50-70		Maskesini doğru kullanan	%96	GPU	8,62 -9,80
Model 3	Latte Panda Delta 4 GB		Maske kullanmayan	%92	CPU	1,71 - 2,08

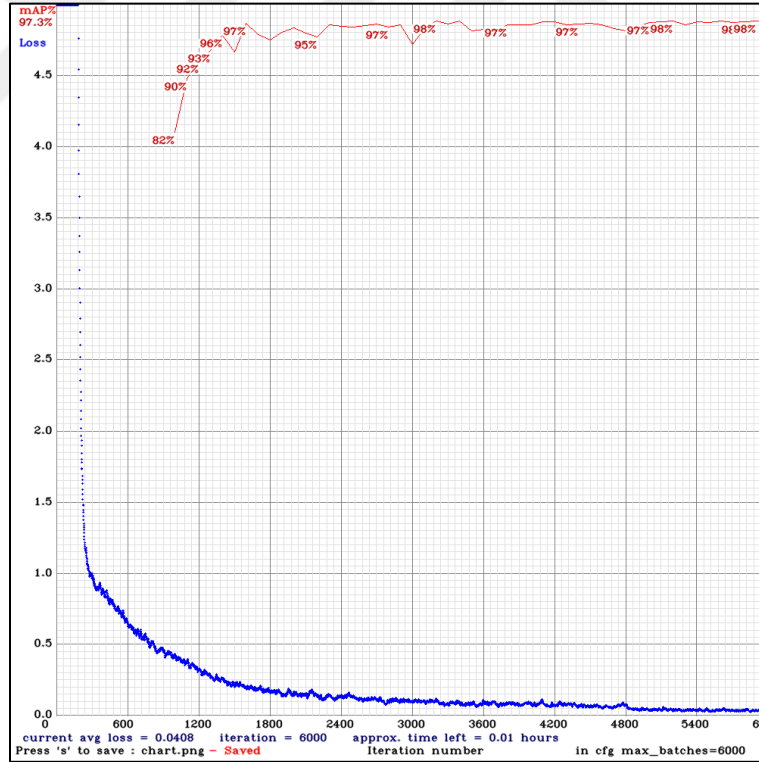
Model 4	Latte Panda Delta 4 GB		Maskesini doğru kullanan	%97	CPU	0,17 - 0,20
Model 1	Google Colab Pc		Maskesini doğru kullanan	%99	GPU	47,1 - 55,4
Model 2	Lenovo Z50-70		Maskesini yanlış kullanan	%94	GPU	8,53 -9,72
Model 3	Raspbery Pi 4 – 8 GB		Maskesini yanlış kullanan	%100	CPU	0,17 - 1,74

Model 4	Raspbery Pi 4 – 4 GB		Maskesini doğru kullanan	%94	CPU	0,11 - 0,15
Model 1	Google Colab Pc		Maskesini yanlış kullanan	%100	GPU	46,8 - 57,2
Model 2	Raspbery Pi 4 – 8 GB		Maskesini doğru kullanan	%99	CPU	0,12 - 0,16
Model 3	Nvidia Jetson Nano 4 GB		Maskesini yanlış kullanan	%89	GPU	8.6 - 12,9

Model 4	Nvidia Jetson Nano 4 GB		Maskesini yanlış kullanan	%97	GPU	1,1 - 1,7
------------	----------------------------------	---	---------------------------------	-----	-----	-----------

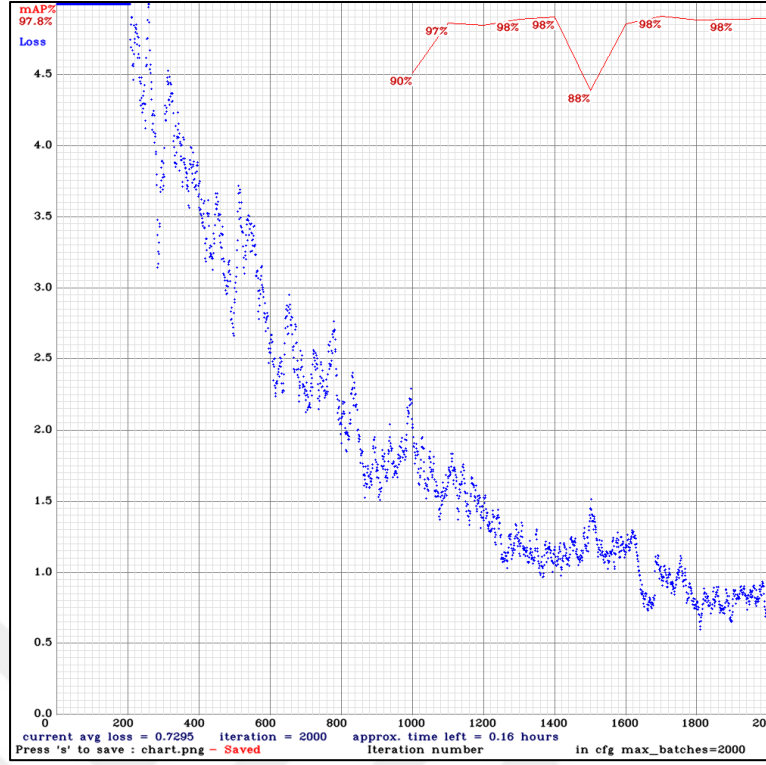
5.3. Google Colab Bulguları

Şekil 5.7’ de YOLOv4-tiny derin öğrenme algoritması ile elde edilen Model 1’e ait ortalama hassasiyet (mAP) ve kayıplar (loss) grafiği görülmektedir. Şekil 5.7’ de sınıf başına 2000 iterasyon sonucunda elde edilen mAP değeri kırmızı renk ile gösterilmiştir. Mavi renk ile gösterilen değer eğitim sırasında oluşan kayıpları göstermektedir. mAP değerinin eğitim boyunca arttığı görülmektedir. Kayıp değerlerinin eğitim boyunca azaldığı görülmektedir.



Şekil 5. 7. Model 1’ e ait mAP ve kayıplar grafiği.

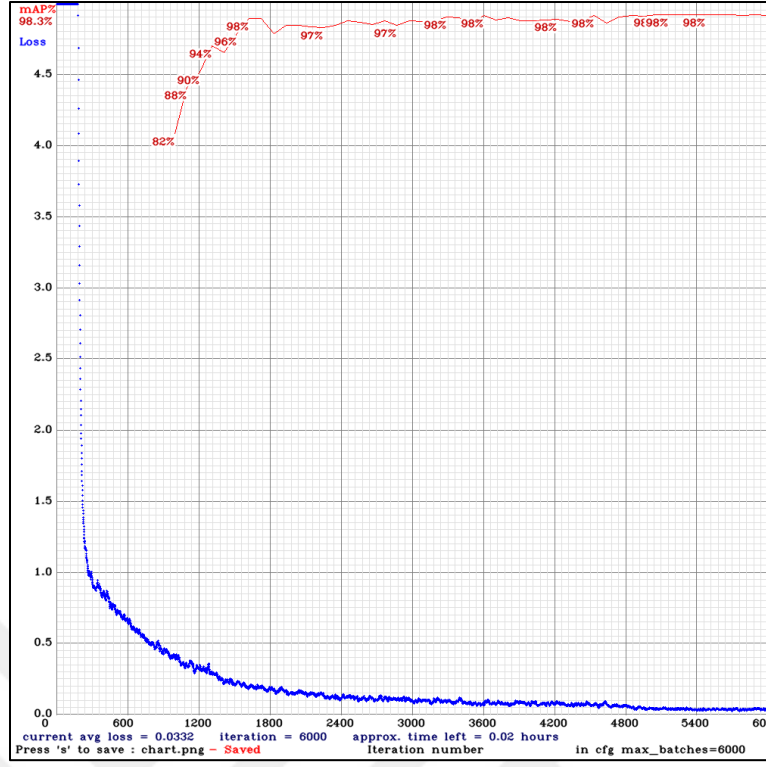
Şekil 5.8’ de YOLOv4 derin öğrenme modeli ile 2000 iterasyon sonunda elde edilen Model 2’ ye ait ortalama hassasiyet (mAP) ve kayıplar (loss) grafiği görülmektedir.



Şekil 5. 8. Model 2' ye ait mAP ve kayıplar grafiği.

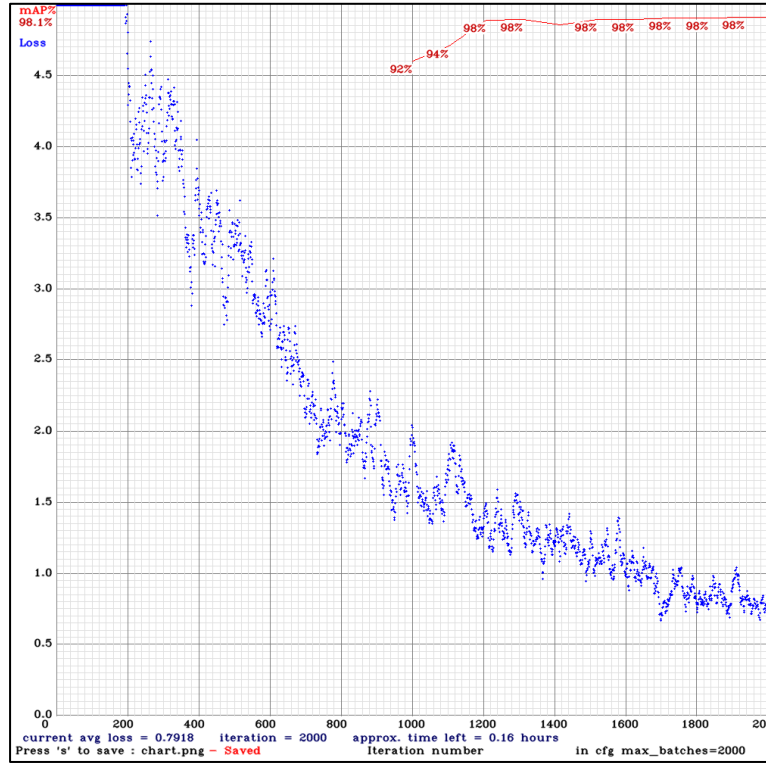
Şekil 5.7 - 5.11' de yer alan grafikler Google Colab platformundan oluşturulmuştur. Model 2' nin eğitiminde YOLOv4 derin öğrenme ağı kullanması ile daha az iterasyon ile daha yüksek mAP değeri elde edildiği görülmektedir.

Veri setinin %70'lik bölümünde yer alan görüntüler yatay yönde aynalama veri çoğaltma yöntemi ile çoğaltılarak 3 katına çıkarılmıştır. Veri setinin eğitim bölümünde yapılan bu değişiklik sonrası YOLOv4-tiny derin öğrenme algoritması ile oluşturulan Model 3'e ait ortalama hassasiyet (mAP) ve kayıplar (loss) grafiği Şekil 5.9' da görülmektedir.



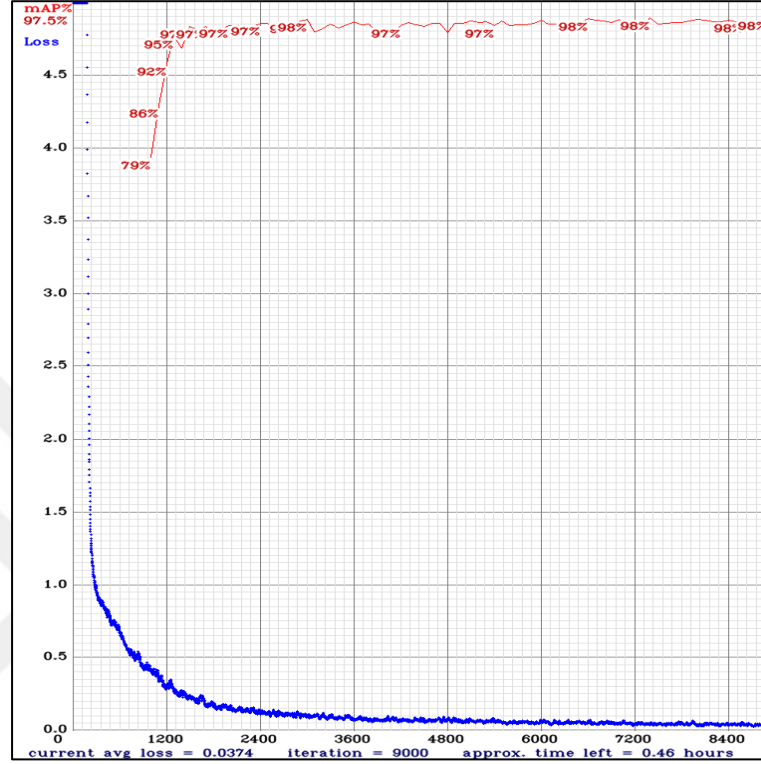
Şekil 5. 9. Model 3'e ait mAP ve kayıplar grafiği.

Şekil 5.10' da veri setinin çoğaltılması sonrasında YOLOv4 derin öğrenme algoritması kullanılarak elde edilen Model 4'e ait ortalama hassasiyet (mAP) ve kayıplar (loss) grafiği görülmektedir.



Şekil 5. 10. Model 4'e ait mAP ve kayıplar grafiği.

İterasyon sayıları YOLOv4-tiny ile eğitilen model için 9000, YOLOv4 için eğitilen model için 4000' e çıkarılarak Model 1 ve Model 2 eğitiminde kullanılan veri seti üzerinden modeller tekrar eğitilmiştir. Şekil 5.11' de YOLOv4-tiny derin öğrenme algoritmasıyla 9000 iterasyon ile gerçekleştirilen model eğitime ait mAP ve kayıplar grafiği görülmektedir.



Şekil 5. 11. 9000 iterasyon sonucunda elde edilen mAP ve kayıplar grafiği.

Şekil 5.7 ve Şekil 5.11 incelendiğinde mAP ve eğitim kayıplarında 6000. iterasyondan sonra sabit dalgalanmalar yaptığı görülmektedir. Ayrıca 9000. iterasyon sonunda elde edilen mAP değeri 6000. iterasyonda elde edilen mAP değerinden sadece %0,12, ortalama eğitim kaybı olarak 0,0034, IoU değeri olarak %0,22 gibi küçük iyileşmeler gösterdiği görülmüştür. 4000 iterasyon ile eğitilen model ile 2000 iterasyon ile yapılan modelin eğitim sonuçları değerlendirildiğinde IoU, mAP ve eğitim kayıplarında büyük değişimler olmadığı görülmüştür. 4000 iterasyon ile gerçekleştirilen modelin mAP değeri %0,17, eğitim kaybı olarak 0,0097, IoU değeri olarak %0,41 değeri gibi küçük iyileşmeler gösterdiği görülmüştür. Bu bağlamda, eğitimlerde IoU değerlerinde büyük değişimlerin olmaması, mAP ve eğitim kayıplarını gösteren grafiklerde çalışmada belirlenen iterasyon sayılarından sonraki değerlerde sabit dalgalanmaların görülmesinden dolayı iterasyon sayıları YOLOv4 için 2000, YOLOv4-tiny için 6000 olarak uygulanması bu çalışma için yeterli olarak görülmüştür (Ukhwah ve arkadaşları, 2019; AlexeyAB, 2018).

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında hareketli ve sabit noktalarda maske kontrolünü gerçekleştirebilen yapay zekâ destekli MKS yazılımı geliştirilmiştir. Ayrıca hareketli maske kontrolü, ateş ölçümü, hasta kaydı, oksijen satürasyonu, nabız ölçümü, yönlendirme ve soru-cevap işlemlerini gerçekleştirebilen bir robot geliştirilmiştir. Maske kontrolünü 3 sınıfa (maskesini doğru kullanan, maskesini yanlış kullanan ve maske kullanmayan) göre gerçekleştirebilen 4 adet yapay zekâ modeli oluşturulmuştur. Oluşturulan yapay zekâ modelleri tflite formatına çevrilerek Tablo 5.2’ de yer alan donanımlar üzerinde çalıştırılmıştır. Geliştirilen yapay zekâ destekli MKS yazılımı sabit noktalarda ve robot üzerinde hareketli olarak çalıştırılmıştır. MKS yazılımında yer alan sesli mesajlar ile maske denetlemesi yapılırken maske takmayan ve maskesini yanlış takan kişiler kibarca uyarılmış, maskesini takan kişilere teşekkür edilmiştir. Ayrıca robotta yer alan sesli asistan yazılımı ile sağlık kuruluşlarında kullanılmak üzere planlanan kayıt, yönlendirme, soru-cevap, ateş ölçme, oksijen satürasyonu ve nabız ölçüm işlemlerine ait testler başarıyla gerçekleştirmiştir. Geliştirilen robot IzeRO20 projesi ile TEKNOFEST 2021 yılında üniversite ve üzeri seviyesi sağlık ve ilk yardım kategorisinde 3.lük derecesi elde etmiştir.

Tez kapsamında oluşturulan yapay zekâ modelleri Google Colab web ortamında üzerinde ücretsiz olarak yüksek kapasiteli GPU birimlerine sahip Tesla T4 donanımı üzerinden eğitilmiştir. Model 1, Model 2, Model 3 ve Model 4 yapay zekâ modellerine ait mAP değerleri sırası ile %97,63, %98,09, %98,37, %98,11 olarak elde edilmiştir. Model 1, Model 2, Model 3 ve Model 4 yapay zekâ modellerinin IoU değerleri sırası ile %71,27, %81,77, %72,13, %81,90 olarak elde edilmiştir. Model 1, Model 2, Model 3 ve Model 4 Tablo 5.2’ de gösterilen donanımlarda çalıştırılmıştır. Tablo 5.3’ de yapay zekâ modellerinin GPU’ ya sahip bilgisayarlarda daha iyi FPS değerleri elde ettiği görülmüştür. Model 2’nin Model 1’ e göre daha doğru tahminler üretmesinin nedeni Model 2’nin YOLOv4 derin öğrenme algoritması ile geliştirilmiş olmasıdır. YOLOv4 derin öğrenme algoritması YOLOv4-tiny’ göre ağ yapısı gereği daha iyi öğrenen modeller üretirken hız açısından yavaş modeller ürettiği görülmüştür. Yine Tablo 5.3 ’de eğitim sürecinde tercih edilen YOLO algoritmalarından dolayı Model 1 ve Model 3’ün FPS değerlerinin Model 2 ve Model 4’e göre daha yüksek olduğu görülmektedir. Veri seti üzerinde çoğaltma işlemi sonrasında geliştirilen modellerden Model 3 Model 1’ e göre mAP ve IoU değerleri açısından daha başarılı olduğu Tablo 5.3’ de görülmektedir. Model 4 ise Model 2’ ye göre mAP, IoU metrikleri açısından daha başarılıdır. YOLOv4-tiny algoritması ile geliştirilen Model 3 mAP

değeri açısından YOLOv4 ile geliştirilen Model 4' ten % 0,26 daha önde olsa da IoU değeri açısından %9,77 geridedir. Tablo 5.1' de yer alan Model 3 ve Model 4 değerlerinde hassasiyet değeri olarak Model 4'ün daha iyi olduğu görülmektedir. Bu bağlamda nesne tespitinde yapay zekâ modellerinin başarısını kıyaslamak için yalnızca mAP gibi bir değerin baz alınması bu çalışmada da görüldüğü gibi yeterli olmamaktadır.

Geliştirilen robotun gelecekte otonom olarak hareket edebilmesi için LIDAR donanımının robota eklenmesi ve robotun ergonomik yapısının geliştirilmesi hedeflenmektedir.



7. KAYNAKLAR

- Arı, A. & Berberler, M. E. (2017). Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı . *Acta Infologica* , 1 (2) , 55-73 . Retrieved from <https://dergipark.org.tr/tr/pub/acin/issue/33868/335553>
- Kabalıcı, E. (2014). Yapay Sinir Ağları. Ders Notları, <https://ekblc.files.wordpress.com/2013/09/ysa.pdf>
- Sarı, F. (2021). Cahit Arf'in "Makine Düşünebilir mi ve Nasıl Düşünebilir?" Adlı Makalesi Üzerine Bir Çalışma . *TRT Akademi* , 6 (13) , 812-833 . DOI: 10.37679/trta.962940
- Ataseven, B. (2013). YAPAY SİNİR AĞLARI İLE ÖNGÖRÜ MODELLEMESİ . *Öneri Dergisi* , 10 (39) , 101-115 . Retrieved from <https://dergipark.org.tr/tr/pub/maruoneri/issue/17900/187806>
- Batagelj, B., Peer, P., Štruc, V., & Dobrišek, S. (2021). How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Applied Sciences*, 11(5). <https://doi.org/10.3390/app11052070>
- Bhuiyan, M. R., Khushbu, S. A., & Islam, M. S. (2020). A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3. 2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020. <https://doi.org/10.1109/ICCCNT49239.2020.9225384>
- Cabani, A., Hammoudi, K., Benhabiles, H., & Melkemi, M. (2021). MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health*, 19. <https://doi.org/10.1016/J.SMHL.2020.100144>
- Çalışkan P, S., & Metintas, S. (2020). COVID-19 Pandemisinde Sağlık Çalışanları. Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Halk Sağlığı Dergisi. <https://doi.org/10.35232/estudamhsd.789806>
- Das, A., Wasif Ansari, M., & Basak, R. (2020). COVID-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. 2020 IEEE 17th India Council International Conference, INDICON 2020. <https://doi.org/10.1109/INDICON49873.2020.9342585>
- David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., & Warden, P. (2021). Tensorflow Lite Micro: Embedded Machine Learning on Tinyml Systems.
- Eriş, M. Derin Öğrenme Yöntemleri Kullanılarak Adli Bilişim İncelemelerinde Delil Çıkarımının Gerçekleştirilmesi. Yüksek Lisans Tezi, Fırat Üniversitesi, 2018.
- Gedik, O., & Demirhan, A. (2021). Comparison of the effectiveness of deep learning methods for face mask detection. *Traitement Du Signal*, 38(4). <https://doi.org/10.18280/ts.380404>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5). <https://arxiv.org/abs/1311.2524>
- Koyuncu, T., Meçik, O. (2020). COVID-19 Pandemisinin Türkiye’de Ekonomik Büyümeye Sektörel Ve Sektörler Arası Etkileri. <https://doi.org/10.11611/yead.793257>
- Gül, H., Gülüm, M., İlter, B., Hasde, M., Çiğdem Şimşek, A., Emre Bulut, Y., Şirin, H.,

- Bozcuk Güzeldemirci, G., Yılmaz Koç, E., Felekoğlu, Z., Yavuz Leblebici, S., Topaç, O. (2020). Covid-19 sürecinde sağlık çalışanları için alınan koruyucu önlemlerin etkisi. Araştırma Makalesi Ankara Med J, 4, 1000. <https://doi.org/10.5505/amj.2020.09825>
- Güldemir, N. H. Yaşa bağlı makula dejenerasyonunun optik koherens tomografi görüntüleri kullanılarak derin öğrenme yöntemleri ile tespiti. Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi, 2022.
- Tan, G. F., Yüksel, S. A., Aydemir, E., Ersoy, M., (2021). Derin Öğrenme Teknikleri İle Nesne Tespiti Ve Takibi Üzerine Bir İnceleme. European Journal of Science and Technology, 25, 159–171. <https://doi.org/10.31590/ejosat.878552>
- Hakverdioğlu, G. (2007). Oksijen Saturasyonunun Değerlendirilmesinde Pulse Oksimetre Kullanımı. <http://eskidergi.cumhuriyet.edu.tr/makale/1624.pdf>
- Jiang, Z., Zhao, L., Li, S., Jia, Y., & Liquan, Z. (2020). Real-time object detection method for embedded devices. <https://arxiv.org/abs/2011.04244>
- Jiao, L., Zhang, F., Liu, F., Member, S., Yang, S., & Feng, Z. (2019). A Survey of Deep Learning-Based Object Detection. <https://doi.org/10.1109/ACCESS.2019.2939201>
- Kılıçarslan, S., Adem, K. & Çelik, M. (2021). An overview of the activation functions used in deep learning algorithms . Journal of New Results in Science , 10 (3) , 75-88 . DOI: 10.54187/jnrs.1011739
- Kumar, G.P., Shashank, Y.S., Rohit, S., Ramani, S., (2021). Face Mask Detection with Raspberry Pi. International Journal for Research in Applied Science and Engineering Technology, 9(VI). <https://doi.org/10.22214/ijraset.2021.35778>
- Lin, H. H., Dandage, H. K., Lin, K. M., Lin, Y. T., & Chen, Y. J. (2021). Efficient cell segmentation from electroluminescent images of single-crystalline silicon photovoltaic modules and cell-based defect identification using deep learning with pseudo-colorization. Sensors, 21(13). <https://doi.org/10.3390/s21134292>
- Lippi, M., Bonucci, N., Carpio, R. F., Contarini, M., Speranza, S., & Gasparri, A. (2021). A YOLO-based pest detection system for precision agriculture. 2021 29th Mediterranean Conference on Control and Automation, MED 2021. <https://doi.org/10.1109/MED51440.2021.9480344>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9905 LNCS, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
- Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). 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, 65. <https://doi.org/10.1016/j.scs.2020.102600>
- Mahurkar, R. R., & Gadge, N. G. (2021). Real-time COVID-19 Face Mask Detection with YOLOv4. Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021. <https://doi.org/10.1109/ICESC51422.2021.9533008>
- Manhas, M., Sanduja, D., Vashisth, R., & Aggarwal, N. (2022). Automated Gate Control System using Face Mask Detection. 2022 International Mobile and Embedded

- Technology Conference, MECON 2022, 184–188.
<https://doi.org/10.1109/MECON53876.2022.9752417>
- Mercaldo, F., & Santone, A. (2021). Transfer learning for mobile real-time face mask detection and localization. *Journal of the American Medical Informatics Association*, 28(7). <https://doi.org/10.1093/jamia/ocab052>
- Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 66. <https://doi.org/10.1016/j.scs.2020.102692>
- Najib A. F., Fauzan, M. N., & Riza, N. (2021). Multiple Linear Regression and Deep Learning in Body Temperature Detection and Mask Detection. *IT Journal Research and Development*. <https://doi.org/10.25299/itjrd.2022.7424>
- Öğüt, G. E. Makine Öğrenmesi ve Derin Öğrenmede CPU ve GPU Hızlandırma Tekniklerinin Karşılaştırması. Yüksek Lisans Tezi, Muğla Sıtkı Koçman Üniversitesi, 2019.
- Onan, A. (2020). Evrimsel Sinir Ağı Mimarilerine Dayalı Türkçe Duygu Analizi. *European Journal of Science and Technology Special Issue*, 374–380. <https://doi.org/10.31590/ejosat.780609>
- Oumina, A., El Makhfi, N., & Hamdi, M. (2020, December 2). Control the COVID-19 Pandemic: Face Mask Detection Using Transfer Learning. 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS 2020. <https://doi.org/10.1109/ICECOCS50124.2020.9314511>
- Parico, A. I. B., & Ahamed, T. (2021). Real time pear fruit detection and counting using yolov4 models and deep sort. *Sensors*, 21(14). <https://doi.org/10.3390/s21144803>
- Pestana, D., Miranda, P. R., Lopes, J. D., Duarte, R. P., Vestias, M. P., Neto, H. C., & De Sousa, J. T. (2021). A Full Featured Configurable Accelerator for Object Detection with YOLO. *IEEE Access*, 9, 75864–75877. <https://doi.org/10.1109/ACCESS.2021.3081818>
- Aalami N., “Derin Öğrenme Yöntemlerini Kullanarak Görüntülerin Analizi.” <https://dergipark.org.tr/tr/pub/estudambilisim/issue/45482/566828>
- Xu, B., Wang, N., Kong, H., Chen, T., & Li, M. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network. <https://doi.org/10.48550/arxiv.1505.00853>
- Qiu, J. (2020). Covert coronavirus infections could be seeding new outbreaks. In *Nature*. <https://doi.org/10.1038/d41586-020-00822-x>
- Rani, N., Jain, R., Patel, S., Ruhela, T., Kumari, P., & Kumawat, R. (2022). Real-Time Face Mask Detection Using Raspberry Pi and Camera. *Lecture Notes in Networks and Systems*, 300 LNNS. https://doi.org/10.1007/978-3-030-84760-9_65
- Aydın, M. R. (2019). Yapay Sinir Ağları İle Talep Tahmini: Perakende Sektöründe Bir Uygulama . *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi* , 18 (35) , 43-55 . Retrieved from <https://dergipark.org.tr/tr/pub/ticaretfbid/issue/55970/565281>
- Reuters. (2020). COVID robocop: the droid that reminds you to wear your mask. Reuters. <https://www.reuters.com/article/us-health-coronavirus-france-robot/covid-robocop->

the-droid-that-reminds-you-to-wear-your-mask-idUSKBN25Z2IN

- Saygılı, Z., Özmen, G. (2022). İnsansız Hava Araçlarında Gömülü Sistem Üzerinden Derin Öğrenme ile Nesne Tespiti. *European Journal of Science and Technology Special Issue*, 34, 292–298. <https://doi.org/10.31590/ejosat.1081713>
- Şeker, A. , Diri, B. & Balık, H. H. (2017). Derin Öğrenme Yöntemleri Ve Uygulamaları Hakkında Bir İnceleme . *Gazi Mühendislik Bilimleri Dergisi* , 3 (3) , 47-64 . Retrieved from <https://dergipark.org.tr/tr/pub/gmbd/issue/31064/372661> Serkan, Y. (2021). Ağız dış ve çene radyolojisi anabilim dalı karşılaştırılmalı olarak incelenmesi.
- Seyyarer, E., Ayata, F., Uçkan, T. & Karci, A. (2020). Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması Ve Kıyaslanması . *Computer Science* , 5 (2) , 90-98 . Retrieved from <https://dergipark.org.tr/tr/pub/bbd/issue/57870/752132>
- Shubham, S. (2018). Wireless Arduino Programming Shield | Full Prototype Project. <https://www.electronicsforu.com/electronics-projects/prototypes/wireless-arduino-programming-shield>
- Sucu, İ. & Ataman, E. (2020). Dijital Evrenin Yeni Dünyası Olarak Yapay Zeka Ve Her Filmi Üzerine Bir Çalışma . *Yeni Medya Elektronik Dergisi* , 4 (1) , 40-52 . Retrieved from <https://dergipark.org.tr/tr/pub/ejnm/issue/51095/666171>
- URL 1. (2019). İnsansı Robot Pepper Türkiye’nin sağlık turizminde kullanılmaya başlandı - Yaşam Haberleri. <https://www.milliyet.com.tr/pembenar/insansi-robot-pepper-turkiye-nin-saglik-turizminde-kullanilmaya-baslandi-2855223>
- URL 2. (2020). Face Mask Detection ~12K Images Dataset | Kaggle. <https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset>
- URL 3. (2021). Covid19 – IASP TUS PHID System& Robot for Human Body Temperature Measurement. <https://www.iasp.ws/covid19>
- URL 4. (2012). SQL Server Management Studio’yu Kullanma | Microsoft Docs. [https://docs.microsoft.com/tr-tr/previous-versions/sql/sql-server-2012/ms174173\(v=sql.110\)?redirectedfrom=MSDN](https://docs.microsoft.com/tr-tr/previous-versions/sql/sql-server-2012/ms174173(v=sql.110)?redirectedfrom=MSDN)
- URL 5. (2021). YOLOv4 vs YOLOv4-tiny, <https://medium.com/analytics-vidhya/yolov4-vs-yolov4-tiny-97932b6ec8ec>
- URL 6. (2021). Bluetooth RC Car, <https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&gl=TR>
- URL 7. (2021). L298N Motor Driver Module, <https://components101.com/modules/l293n-motor-driver-module>
- URL 8. (2022). AKES - Görüntü İşleme İle Ürün Sayımı, <https://www.akesltd.com/kameradan-urun-sayma/>
- Karakuş, A. B., Derin Öğrenme Ve Büyük Veri Yaklaşımları İle Metin Analizi. Doktora Tezi, Fırat Üniversitesi, 2018
- Uzun, T. (2020). Yapay Zeka Ve Sağlık Uygulamaları . *İzmir Katip Çelebi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* , 3 (1) , 80-92 . Retrieved from <https://dergipark.org.tr/tr/pub/ikacuiibfd/issue/54124/710604> Wang, B., Zhao, Y., & Chen, C. L. P. (2021). Hybrid Transfer Learning and Broad Learning System for

- Wearing Mask Detection in the COVID-19 Era. IEEE Transactions on Instrumentation and Measurement, 70. <https://doi.org/10.1109/TIM.2021.3069844>
- Yang, J., & Yang, G. (2018). Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer. Algorithms, 11(3), 1–15. <https://doi.org/10.3390/A11030028>
- İnik, Ö. & Ülker, E. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri . Gaziosmanpaşa Bilimsel Araştırma Dergisi , 6 (3) , 85-104 . Retrieved from <https://dergipark.org.tr/tr/pub/gbad/issue/31228/330663> Nabiye V (2012). Yapay Zeka. Seçkin Yayınları, Ankara, 776 s.
- Küçük, K., Kahraman, F., Kamaşak, M. E. & Adalı, E. (2021). Öğrenen ve Öngören Varlık Yonetimi . Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi , 14 (2) , 120-136 . DOI: 10.54525/tbbmd.1018536Yu, J., & Zhang, W. (2021). Face mask wearing detection algorithm based on improved yolo-v4. Sensors, 21(9). <https://doi.org/10.3390/s21093263>
- Keskenler, M. F. & Keskenler, E. F. (2017). Geçmişten Günümüze Yapay Sinir Ağları ve Tarihçesi . Takvim-i Vekayi , 5 (2) , 8-18 . Retrieved from <https://dergipark.org.tr/tr/pub/takvim/issue/33375/346279>
- Ukhwah, E. N., Yuniarno, E. M., & Suprpto, Y. K. (2019). Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network. Proceedings - 2019 International Seminar on Intelligent Technology and Its Application, ISITIA 2019, 35–40. <https://doi.org/10.1109/ISITIA.2019.8937176>
- Yüncü, V. & Yılan, Y. (2020). COVID-19 Pandemisinin Sağlık Çalışanlarına Etkilerinin İncelenmesi: Bir Durum Analizi . Iğdır Üniversitesi Sosyal Bilimler Dergisi , (EK SAYI (2020)) , 373-402 . Retrieved from <https://dergipark.org.tr/tr/pub/igdirsosbilder/issue/66832/1045333>
- Fu, H., Song, G. ve Wang, Y. (2021). Improved YOLOv4 Marine Target Detection Combined with CBAM. Symmetry, 13(4), 623. doi:10.3390/sym13040623
- AlexeyAB. (2018). AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet). <https://github.com/AlexeyAB/darknet>
- Xu, Q., Zheng, W., Liu, X., & Jing, P. (2020). Applied Artificial Intelligence Deep Learning Technique Based Surveillance Video Analysis for the Store. <https://doi.org/10.1080/08839514.2020.1784611>