



**GÖRÜNTÜ İŞLEME
TEKNİKLERİ ile İHA TESPİTİ**

Ensar KOŞATEPE

**Yüksek Lisans Tezi
Elektrik Elektronik Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Nur Hüseyin KAPLAN**

**2021
Her hakkı saklıdır.**



**FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZİ**

GÖRÜNTÜ İŞLEME TEKNİKLERİ ile İHA TESPİTİ

Ensar KOŞATEPE

Tez Danışmanı: Doç. Dr. Nur Hüseyin KAPLAN

Anabilim Dalı: Elektrik Elektronik Mühendisliği

Erzurum

2021

Her hakkı saklıdır

T.C.
ERZURUM TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
TEZ ONAY FORMU

GÖRÜNTÜ İŞLEME TEKNİKLERİ VE İHA TESPİTİ

Doç. Dr. Nur Hüseyin KAPLAN danışmanlığında, Ensar KOŞATEPE tarafından hazırlanan bu çalışma 14 / 01 / 2021 tarihinde aşağıdaki jüri tarafından Elektrik Elektronik Mühendisliği Ana Bilim Dalı'nda Yüksek Lisans tezi olarak **Oy birliği ile (3/3)** ile kabul edilmiştir.

Başkan	: Doç.Dr. NUR HÜSEYİN KAPLAN	İmza	:
Üye	: Dr.Öğr.Üyesi EMİN ARGUN ORAL	İmza	:
Üye	: Dr.Öğr.Üyesi NURBANU GÜZEY	İmza	:

Yukarıdaki sonucu onaylıyorum

Prof. Dr. Arzu GÖRMEZ
Enstitü Müdürü

Bu tez çalışması tarafından nolu proje ile desteklenmiştir.

ETİK KURALLARA UYGUNLUK BEYANI

Erzurum Teknik Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki tüm bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

14 / 01 / 2021

İmzası

Ensar KOŞATEPE

ÖZET

YÜKSEK LİSANS TEZİ

GÖRÜNTÜ İŞLEME TEKNİKLERİ ve İHA TESPİTİ

Ensar KOŞATEPE

Erzurum Teknik Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Ana Bilim Dalı

Danışman: Doç. Dr. Nur Hüseyin KAPLAN

İnsanlık tarihinin başlangıcından günümüze kadar belirli evrelerle gelişen teknoloji, insanların ihtiyaç ve istekleri doğrultusunda, doğadan örnek alınarak ilerlemektedir. Gelişen teknolojinin insan hayatını kolaylaştırmasının yanında ülkelerin kendi öz kaynaklarını kullanabilmesi halinde savunma sanayi ile birlikte ülkelerin caydırıcılık etkisi artmaktadır. Ancak bu avantajlarının yanı sıra diğer ülke veya kötü niyetli insanların da bu teknolojilere sahip olarak doğrudan ya da dolaylı bir şekilde tehdit oluşturmaları kaçınılmazdır. Bu amaçla dünyada gelişen teknolojiyi takip edip uygulamalara katılarak daha iyisini üretmenin yanında olası tehdit içeren kullanımlara karşı önlemler almak, ülkemiz için hayati öneme sahiptir.

Bu çalışmada, özellikle son dönemde gelişen İnsansız Hava Aracı (İHA) tespit ve tanımlanması için uygulama gerçekleştirilmiştir. Doğadan örnek alınarak, görme sisteminin bir benzeri donanımsal olarak kurulmuştur. Kurulan donanım üzerinden İHA tanımlanması ve tespiti için, son yıllarda kullanım alanı artan derin öğrenme yöntemlerinden SSD Mobilnet yöntemi ve Google alt yapısını kullanan TensorFlow kütüphanesi kullanılmıştır. Misyonu gereği belli saatlerde belli yükseklikte bulunan İHA, sürekli hareket halindedir. Bundan dolayı hızlı ve gerçek zamanlı nesne tanımlaması yapabilmek için seçilen SSD modeli bu çalışmanın ana temasını oluşturmuştur.

Bu yazılım ve programların kullanımları pratik, hızlı ve ulaşılabilir donanımlarla desteklenmiştir.

Yıl 2021, 76 sayfa

Anahtar Kelimeler: Görüntü İşleme, İHA, TensorFlow, SSD mobilnet

ABSTRACT

MS. Thesis

IMAGE PROCESSING TECHNIQUES and DRONE DETECTION

Ensar KOŞATEPE

Erzurum Technical University
Graduate School of Natural and Applied Sciences
Department of Electrical Electronic Engineering

Supervisor: Assoc. Prof. Dr. Nur Hüseyin KAPLAN

Technology, which has developed in certain stages from the beginning of human history to the present, can be produced by taking samples from nature in line with the needs and desires of people. In addition to facilitating human life due to the developing technology, the ability of countries to use their own resources and deterrent effect increases with defence. However, in addition to these advantages, other countries or people with bad intentions also pose a threat in all situations and conditions, directly or indirectly, by having these technologies. For this purpose, it is of vital importance for our country to take various measures besides producing better ones by following the developing technology in the world and participating in applications.

In this study, an application has been made for the detection and identification of the UAV that has developed recently. A visual system in a digital environment similar to that of nature was tried to be established. For the identification and detection of UAVs on the installed hardware, SSD Mobilnet method which is among the deep learning methods whose usage area has increased in recent years and the TensorFlow library using Google infrastructure, have been used. Due to its mission, the UAV, which is at a certain height at certain hours, is constantly in motion. For this reason, the SSD model chosen for fast and real-time object identification was the main theme of this study.

These software and programs are supported with practical, fast and accessible hardware.

Year 2021, 76 page

Keywords: Image processing, UAV, TensorFlow, SSD mobilnet

TEŞEKKÜR

Yapmış olduğum bu çalışmada bilgi ve tecrübelerini benimle paylaşan, çalışmalarımda bana içtenlikle yardımcı olan, yoluma ışık tutup her daim yanımda bulunan, onunla birlikte çalışmaktan onur duyduğum kıymetli hocam Doç. Dr. Nur Hüseyin KAPLAN' a en içten teşekkürlerimi sunarım.

Laboratuvar imkânlarını bu çalışmayı yapmamız için bize sunan Erzurum Teknik Üniversitesi yönetimi ve Mühendislik Fakültesi Elektrik Elektronik Mühendisliği bölümü hocalarımıza teşekkürlerimi sunarım.

Hayat akışı içinde bana güvenen ve inanan annem Fadime KOŞATEPE' ye ve babam Aziz KOŞATEPE' ye ve kardeşlerime teşekkürlerimi sunarım. Bu çalışmada tecrübelerinden faydalandığım, maddi ve manevi olarak her zaman yanımda olan abim Öğr. Gör. Abdulhadi KOŞATEPE' ye teşekkür ederim. Ayrıca çalışmalarım süresinde yanımda olan kıymetli arkadaşlarım Ramazan ÖZCAN ve Muhammed Sertaç FIRAT'a ve Ayşe BAĞIRAN'a teşekkür ederim.

Ensar KOŞATEPE

Ocak 2021

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
SİMGELER ve KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	ix
1. GİRİŞ	1
1.1 Tezin Amacı	4
1.2 Tezin İçeriği	5
2. KAYNAK ÖZETLERİ	7
3. MATERYAL ve YÖNTEM	12
3.1 Sistemlerde Yapay Zekâ	12
3.2. Evrişimsel Sinir Ağları ve Modelleri	14
3.2.1 Evrişimsel sinir ağlarını oluşturan katmanlar	14
3.2.2 AlexNet modeli	19
3.2.3. ResNet Modeli	20
3.2.4. GoogLeNet modeli.....	20
3.2.5 Lenet-5	21
3.2.6 VGG	22
3.3 Sistem Veri Setleri	23
3.3.1 MS COCO veri seti	23
3.3.2 MNIST veri seti	23
3.3.3 ImageNet veri seti	24
3.4 Veri İşleme Modelleri	25
3.4.1 Nesne tanıma	25
3.4.2 Label map.....	26
3.4.3 TensorFlow model eğitimi	27
3.4.4 SSD mobilnet	29
3.5 Donanımsal Yapılar	36
3.5.1 Step motor	37

3.5.2 Step motor sürücü TB6600	37
3.5.3 Arduino uno	38
3.5.4 Raspberry pi4	39
3.5.5 Pi kamera V2.....	39
3.5.6 Motor için adaptör.....	40
3.5.7 Kapasitif dokunmatik 7 inç ekran	40
4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....	42
5. SONUÇ ve ÖNERİLER.....	56
KAYNAKLAR	58
ÖZGEÇMİŞ.....	63



SİMGELER ve KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
İHA	İnsansız hava araçları
SDD	Single Shot MultiBox Detector
HOG	Yönlendirilmiş gradyanlar histogram
LBP	Yerel iki örüntü
FBTP- NN	Çift katmanlı aşağı ve yukarı yol birleştirilmeli yapay sinir ağı
FF-NN	Üç katmanlı ileri beslemeli yapay sinir ağı
CNN	Convolutional Neural Network
CV	Bilgisayar görüntüsü
ESA	Evrişimsel Sinir Ağı
ReLu	Rectified Linear Unit
R-CNN	Region Convolutional Neural Network
Fast R-CNN	Fast Region Convolutional Neural Network
TF	TensorFlow
MNIST	Modified National Institute of Standards and Technology database
MS COCO	Microsoft Common Objects in Context
VOC	Visual Object Classes
GPU	Grafic Process Unit
ResNet	Residual Network
YSA	Yapay Sinir Ağları
RPN	Region Proposal Network
FC	Fully Connected

ŞEKİLLER DİZİNİ

Şekil 3.1. Yapay zekâ uyarlaması (Adalı vd 2017)	12
Şekil 3.2. Yapay zekânın insan beyni gibi uyarlanması (Adalı vd 2017).....	13
Şekil 3.3.Yapay sinir hücresinin yapısı (Ataseven 2015)	14
Şekil 3.4.Yapay sinir ağlarının matematiksel modeli (Adalı 2018).....	15
Şekil 3.5. Yapay sinir ağlarının genel yapısı (Yurtoğlu 2005)	16
Şekil 3.6. Bir ve birden fazla katmanlı yapay sinir ağı modeli (Lecun 1998)	17
Şekil 3.7. Bir adım kaymalı 5x5'lik giriş görüntüsüne 2x2'lik filtre uygulanarak havuzlama işlemi (Boureau 2010)	18
Şekil 3.8. (a) Standart ESA ağı (b) ESA ağına drop uygulanması (Srivastaya 2014)	19
Şekil 3.9. AlexNet modeli (Bai 2017)	19
Şekil 3.10. ResNet modeli (Seung et al. 2017)	20
Şekil 3.11. GoogLeNet (Szegedy 2016)	21
Şekil 3.12. Lenet-5 modeli (Golovko 2015)	22
Şekil 3.13. VGG modeli (Ferguson 2017)	22
Şekil 3.14. MS COCO data setine ait bazı görüntüler (Jimut and Sahalab 2019)	23
Şekil 3.15. MNIST veri seti	24
Şekil 3.16. ImageNet veri seti çeşitliliği (Tan 2019)	25
Şekil 3.17. Nesne tanıma için örnek görüntü	26
Şekil 3.18. Sistem başlangıç kodları	28
Şekil 3.19. Toplanan İHA görüntüleri ve etiketlenmesi	28
Şekil 3.20. Eğitim sürecinden bir aşama.....	29
Şekil 3.21. SSD modelin yapısı (Liu et al. 2016)	30
Şekil 3.22. Aktivasyon fonksiyonu (Adalı 2018)	31
Şekil 3.23. Basamak fonksiyon grafiği (Kızrak 2019)	32
Şekil 3.24. Doğrusal fonksiyon grafiği (Kızrak 2019)	32
Şekil 3.25. Sigmoid fonksiyon grafiği (Kızrak 2019)	33
Şekil 3.26. ReLu fonksiyon grafiği (Kızrak 2019)	34
Şekil 3.27. Sızıntı fonksiyon grafiği (Kızrak 2019)	35
Şekil 3.28. Step motor.....	37
Şekil 3.29. TB6600 Step Motor Sürücüsü	38

Şekil 3.30. Arduino Uno ve teknik özellikleri	38
Şekil 3.31. Raspbery ve teknik özellikleri	39
Şekil 3.32. Raspbery Pi kamera ve teknik özellikleri.....	40
Şekil 3.33. Adaptör	40
Şekil 3.34. Dokunmatik ekran	41
Şekil 4.1. Raspbery Pi sistemi	42
Şekil 4.2. Kütüphane eklenmesi ve ekranda uyarı yazdırma döngüsü	43
Şekil 4.3. Raspbery içerisinde bazı algoritma kodları	43
Şekil 4.4. Raspbery içerisinde işlem algoritma kodları	44
Şekil 4.5. Ekranda uyarı döngüsü sonucu görülen logo ve boyutu.....	45
Şekil 4.6. Gerçek zaman akışı sağlayan kodlar.....	45
Şekil 4.7. Pi kamera v2	46
Şekil 4.8. Adaptör (dönüştürücü).....	46
Şekil 4.9. Motor sürücü tb6600 ve arduino ile bağlantısı	47
Şekil 4.10. Arduino Uno	48
Şekil 4.11. Nema17 step(adım) motora sahip kamera	48
Şekil 4.12. Kapasitif dokunmatik ekran ve algortima başlatma kodları	49
Şekil 4.13. Algoritma akış şeması.....	52
Şekil 4.14. İHA tespit cihaz görüntüleri (tüm sistem görüntüsü)	53
Şekil 4.15. (a) açık hava hareketsiz İHA tanıma deneme görüntüsü, (b) açık hava hareket halinde İHA tanıma deneme görüntüsü, (c) düşük ışık ortamında kapalı alanda İHA tanıma deneme görüntüsü	54
Şekil 4.16. (a) hareket halindeki İHA tespiti (b) ekranda tespit sonrası verilen uyarı.....	55

ÇİZELGELER DİZİNİ

Çizelge 3.1. Fonksiyonların test sonuçları (Kızrak 2019)	36
Çizelge 4.1. Test sonucu elde edilen veriler	49
Çizelge 4.2. Derlenmiş değerlendirme kriterleri tablosu (Zeren 2020).....	51



1. GİRİŞ

Ülkelerin geleceği ve varoluşlarını devam ettirmeleri için önemli anahtarlardan biri de hiç şüphesiz gökyüzü hâkimiyetidir. Tarih boyunca yapılan savaşlarda gökyüzünden saldırı planları yapılmıştır. İlk dönemde surların arkasına ok atma ve mancınık kullanılmış olmakla birlikte yakın geçmişte sıcak hava balonları, zeplinler ve son dönemde füzeler, roketler ve uçaklardan atılan bomba ve tahrip teçhizatları kullanılmaktadır. Tüm bu teçhizatlarını kullanılabilmesi için insan faktörü oldukça önemlidir. Özellikle doğrudan insanlı sistemlerde yetişmiş personel risk altında bulunmaktadır. Günümüzde ise insan-cihaz arasındaki bağlantı tamamen değişmiş olup insansız sistemlerin kullanımı artmıştır. Bu sayede, risk altında olan sadece kullanılan araç olmuştur. Bununla beraber yetişmiş personelin sadece bir tane ekipmanı değil binlerce ekipmanı uzaktan ve hiç riske atılmadan yönetmesi olanağı sunulmuştur. Kayıp verilen ekipmanları kısa zamanda ikame edip tekrar sahaya sürülmesi her türlü durumu etkili bir şekilde değiştirmesine neden olmuştur. Bu bağlamda İnsansız Hava Araçları (İHA) kullanımı her geçen gün önemini arttırmaktadır.

İHA; kullanıcı olarak içerisinde bir veya daha fazla pilot olmadan GPS kontrollü uzaktan kumanda ile verilen görevi yapabilen, yerde bulunan pilot tarafından veya otomatik olarak yüklenen uçuş planını gerçekleştirebilen hava araçlarıdır (Kahveci ve Can 2017). Değişik özelliklerde çok farklı uygulama kabiliyetlerine sahip İHA, temelde otomatik ve uzaktan kumandalı olarak sınıflandırılabilir (ICAO, 2011).

İHA' nın özellikle askeri alanda artan kullanımı ve tahrip gücünün artmış olması, ülkelerin savunma amacıyla İHA tespiti ve imhasının yapabilmesini oldukça önemli hale getirmiştir. İHA tespiti için kullanılabilecek yöntemlerden birisi de kamera ile elde edilen görüntüler üzerinden tespittir. Bu amaçla; anlık, yüksek kalitede, çeşidi ve işlevi analiz edilebilir görüntüler alabilmek oldukça hayati bir unsurdur. Kamera ile tespitin yapılabilmesi için, kullanılacak kamera, işlemci, bilgisayar ve yazılıma karar verilmesi gerekmektedir.

Bir görüntünün işlenebilmesi için ilk adım, kullanılan amaca uygun bir kamera yardımıyla görüntünün alınması gerekmektedir. Kamera ile elde edilen görüntünün işlenir

1. GİRİŞ

hale getirilmesi için bir işlemciye bağlanması gerekmektedir. Görüntülerin görünür hale gelmesi için bir ekrana ihtiyaç duyulmaktadır. Kullanılacak ekran uygun ve uyumlu bir şekilde programları yüklenmiş ve kullanıma hazır hale getirilerek kullanılmalıdır. (Demirtaş, 2019).

Görüntülerin anlamlandırılabilmesi için Google tarafından yakın tarih 2015’de geliştirilen, açık kaynaklı TensorFlow kütüphanesi ile evrimsel derin öğrenme destekli yeni çalışmalar yapılmaktadır. Kütüphanede kullanılan dijital ortamların GPU veya CPU sistemleri üzerinden Python, C++, Java gibi birçok yazılım programları ile birlikte yapılabilmektedir. (Wang et al. 20017)

Nesnenin ayırt edici özelliklerinin tamamına tensör denir. Bu tensörler nesneyi ayırt etmek için veri girdisi olarak kullanılır. Nesnenin aranan özellikte olması kontrolünü TensorFlow modeli incelemektedir. Bu modelde; Tensör, nesnenin fiziksel özelliklerini matematiksel ifadelerle dönüştürülmüş halinin şekil, renk ve hacim gibi çok boyutlu değişkenlerin ölçülmesinde ve analiz edilmesinde kullanılan yöntemdir. Eğitim, veri kümesinde nesneye ait fotoğraflar sisteme kayıt edilmekte ve ayırt edici özellikleri işlenip öğrenilmektedir. (Pattanayak, 2017)

Nesne tanıma için veri girdi sayılarının fazla olması ve manuel kontrol edilmesine imkân olmadığından dolayı sistemin bir algoritma ile kontrol edilmesi durumunu ortaya çıkarmıştır. Bu da sistemlerin gelişmesine yol açarak birden fazla sistem ve fonksiyonlar oluşturmuş ve oluşturulmaya da devam etmesine yol açmıştır.

Bu çalışmanın amaçlarından biri de gelişmeleri yakından incelemek ve gelişmelere katkılar sunmak için araştırmalar sonucu konu seçilmiştir.

Yakın geçmişte derin öğrenme yöntemleri ile yapılan nesne tanıma ve tespit sistemlerinde doğruluk oranı % 50’in altında iken, çalışmalarla son 10 yılda bu oran % 90’ lara çıkarılmıştır. (Hinton et al. 2016).

TensorFlow modeli ile nesne tanımlamada nesne özellikleri belirginleştirerek bir etiketlenme olayı gerçekleştirilir. Görüntülerin etiketlerinden gerçekleştirilen veri kümesi

sonucunda oluşturulan dosyalardan, çıkarım grafiği oluşturularak nesne tanıma algoritmasına birden fazla girdi meydana getirilir. (Hinton et al. 2012)

Derin öğrenme tekniğinin ve gereksiniminin temelini yapay sinir ağları oluşturmaktadır. Oluşturulan bu sistem, 1980 yıllarında yetersiz donanım ve teknolojik imkânlarından dolayı geliştirilememiştir. Ancak ilerleyen yıllarda teknolojinin gelişmesi ile birlikte günümüzde kullanım alanı oldukça artmaktadır (Hacıoğlu, 2014). Derin öğrenme yapıları çok katmanlı olup çok çeşitli parametreler içerirler. Bu karmaşık yapılar içerisinde en çok kullanılanları; yapay zekâ, otomatik kodlayıcılar, kısıtlı Boltzmann makineleri ve evrişimsel sinir ağlarıdır. (Özcan, 2014).

Evrişimsel sinir ağları ile görüntü sınıflandırma yapılan nesne tanıma için oluşturulan derin öğrenme tekniği ile oldukça başarılı sonuçlar elde edilebilmektedir. Elde edilen sonuçların doğruluk oranını arttırmak için nesneye ait fotoğraf sayısını arttırmak gerekmektedir. Böylece giriş katmanında oluşturulan grafik gerçeğe yaklaşır. Ancak bu durumda etiket fazlalığından dolayı derin öğrenme sistem eğitim süresi artmaktadır. (Cangil ve Çınar, 2016).

Derin öğrenme sistemleri ile nesne tespitinde nesnelerin baskın özelliklerini çıkarma yetisi sayesinde belli nesne üzerinde nesnenin ne olduğunun tanımlanması, ağ yapısı kurgusunun oluşturulması ve veri haritası çıkarımı mümkün olmuştur. Çalışmayı arka planda oluşturan derin öğrenme sistemlerinde, elde edilen görüntülerin boyutlarının karşılaştırılması yapılmakta ve sonrasında derin öğrenme ağından geçirilerek sonuca ulaşılmaktadır. Evrişimsel sinir ağları, kullanım yerlerine göre bazı farklılıklar göstermesine rağmen temelde aynı işi yapmaktadırlar.

Nesne tanıma ve tespitinde kullanılan kütüphaneler arasında Single Shot Multibox Detactor (SSD) modeli tek seferde nesne tanıma işini yapmaktadır. SSD modeli basit çalışma yaptığından dolayı donanımsal olarak raspberry pi4 gibi düşük kapasiteli cihazlarda dâhil hızlı bir şekilde nesne tanıma işlemi yapabilmektedir. (Daş vd 2019).

Bu tez çalışmasında; Savunma sanayi, sağlık ve güvenlik hizmetlerinin yanında birçok sosyal ve kültürel etkinliklerde kullanım alanına sahip İHA araçları, karşı güvenlik ve tehditlere karşı tespit ve tanımlanması oldukça önemli hale geldiği için yapılmıştır.

Bu bağlamda, normal bir bilgisayarın çalışma prensibine ve donanımına oldukça yakın özellikler göstermesinin yanı sıra taşınabilir, oldukça düşük maliyetli ve çok az enerji tüketebilen kullanışlı bir yapıya sahip olan Raspberry Pi 4 kullanımı tercih edilmiştir. Kameranın Raspberry Pi 4' e bağlantıları yapıldıktan sonra elde edilen görüntülerin görüntülenebilmesi için bir ekran bağlanmıştır. Elde edilen görüntünün işlenerek nesne tespiti yapılabilmesi için, tek seferde ve gerçek zamanlı nesne tanımayı gerçekleştirebilen SSD modeli kullanılmıştır. Bunun yanında aurduino programı ve python programları kullanılmıştır. İhtiyaç duyulan donanımsal ekipmanlar Raspberry Pi 4 sistemi ve bu sistemle uyumlu Pi kamera, dokunmatik ekran, güç sağlamak için motor ve enerji aksamı olarak sıralanabilir. Belirtilen donanımlar ile istenilen deneysel çalışma sistemi anlamlı bir şekilde kurulmuştur.

Bahsedilen donanımların entegrasyonunun ardından, gerçek zamanlı deneyler yapılarak İHA tespiti yapılmıştır. Yapılan deneyler sonucunda, yaklaşık % 94 doğrulukla İHA tespiti yapılabilmiştir.

1.1 Tezin Amacı

Teknolojik gelişmeler, insan istek ve ihtiyaçlarına göre hızla gelişmektedir. Hayatı kolaylaştırmak, zamandan tasarruf etmek gibi hedefleri olan teknolojinin bu anlamda sabit kalması düşünülemez. Teknoloji yalnızca ihtiyaçlara cevap vermek adına değil, kaliteli hizmet sunabilmek için de gelişmektedir. Bu isteklere cevap verme adına teknoloji, yazılım ve cihazlar sürekli gelişmektedir.

Gökyüzünün anlaşılma ve gökyüzünde bulunma isteği insanlık tarihi kadar eskidir. Bu isteği gerçekleştirme adına havanın özellikleri kullanılarak birçok sistem geliştirilmiş ve insanlık için önemli adımlar atılmıştır. Bu adımlardan biri de İHA' lardır. Bu sistemler insanlığı daha hızlı gözlemlleme ve doğru analitik düşünme sağlayarak en önemli zenginliği olan zamandan kazandırmıştır.

İHA teknolojisi, ergonomik yapıya ve kolay kullanıma sahip araçlar olması ile bu cihazların askeri kullanımın yanı sıra, tarım, orman, madencilik ve hobi amaçlı sivil kullanımları da bulunmaktadır. Kullanım alanları genişledikçe, kullanım alanlarına göre şekillenen donanım ve yazılımlar da gelişmektedir.

Görüntüleme ve olaylara bütünüyle bakma olanağı sağlayan İHA ve görüntüleme sistemleri ayrılmaz bir bütün haline gelmektedir. Görüntüleme tek başına yeterli olmamaya başlayınca görüntü işleme yöntemleri ile kurum veya kişilere göre özelleştirme gereği görülmüştür. Bu özelleştirme işlemleri çeşitli tekniklerin doğmasına ve kendini sürekli yenilemesine yöneltilmektedir. Bu yenilenme gereğinin farkında olan Google, açık kütüphaneler yahut ara yüzleri geliştirerek, kullanılacak yazılımları doğru ve verimli kullanmayı hedeflemiştir. Böylece erişim ağını genişletip kullanım kolaylıkları sağlamıştır. Bu çalışmada da Google alt yapısı olan SSD model kullanılmıştır.

Hizmet alanlarının yanı sıra savunma sanayi ve istihbarat alanlarında oldukça gelişen İHA'lar, beraberinde büyük tehditler de oluşturmaktadır. Eski teknoloji ile oluşturulmuş savunma ve istihbarat sistemleri, yeni İHA sistemleri karşısında adeta eli kolu bağlı kalmaktadır. Bu tehditlere karşı koymak için İHA tespiti hayati derecede önemlidir. Bu tez çalışmasında İHA tespiti ve kayıt altına alınması üzerinde duruldu ve % 94 oranında doğruluk ile başarı sağlandı.

1.2 Tezin İçeriği

Bu tez çalışmasında, görüntülerin kayıt altına alınması ve nesne tanımlaması yapabilmek için öncelikle donanımsal materyaller step motor, step motor sürücüsü, Pi kamera, adaptör ve kapasitif dokunmatik ekran uygun şartlarda bir araya getirildi. Yapılan İHA tespit cihazının donanımlarının uygun çalışması açısından arduino ve python yazılımları kullanılmıştır. Arduino motor kontrolcüsü olarak python ise görüntü işleme olayının tümünü gerçekleştirmektedir.

Fiziksel olarak oluşturulan İHA'nın tespiti için kamera tarafından çekilen görüntüleri işleyebilmek için Google alt yapısını kullanılarak SSD derin öğrenme modeli ile nesne tanımlama yapıldı.

1. GİRİŞ

Bu çalışma için görüntü işleme teknik ve uygulamaları ile alakalı kaynaklar incelendi. Kaynaklarda bu sistem için sinir ağıları, yapay zekâ, katmanlar ve nesne tanıma adına yapılan çeşitli çalışmalar irdelendi.

Sistemde öğrenme olayı olduğundan yapay zekâ kısmından bahsedildi. Buna bağlı olarak evrişimli sinir ağıları katmanları incelendi. Alexnet, Resnet, Googlenet, Lenet-5, görüntü işleme yöntemlerinden bahsedilmiştir. Derin öğrenme ve görüntü işleme tekniklerinin avantaj ve dezavantajlarından bahsedilmiştir. Sistemin görüntü işleme alt yapısı olan SSDmobilnet ve bu uygulamanın aktivasyon fonksiyonlarından bahsedilmiştir. Hibrit fonksiyonlar kullanılması durumunda ne gibi avantaj ya da dezavantaj elde edileceğinden bahsedilmiştir.

Bu çalışmanın bel kemiği olan nesne tanıma hakkında bilgi verilmiştir. Giriş görüntülerinin bir harita gibi kısımlara ayırıp matrisler halinde öğrenilmesi ve daha sonra canlı görüntüleme de tek tek alınan görüntüleri önceden oluşturduğu matrislerle karşılaştırması durumları incelenmiştir.

Sistemin teorik olarak tamamlanmasından sonra portatifini geliştirmek için donanımsal yapı ele alınmıştır. Step motor, step motor sürücüsü, Arduino, Raspberry pi 4, v2 kamera, kapasitif dokunmatik ekran hakkında bilgi verilmiş her bir bölümün tek ve kendi içerisinde bir bütün halinde çalışması için donanımların nasıl ve çalışma bölge aralıklarının neye göre seçildiğinden bahsedilmiştir.

Çalışmanın yapılma amacından bahsedildikten sonra elde edilen sonuçlar ve performansından bahsedilmiştir. Ayrıca ileriye dönük öneriler sunulmuştur.

2. KAYNAK ÖZETLERİ

Nesne algılama ve tanıma, dijital ortamda görme olayının gerçekleşmesi ve analiz edilerek varlık veya yokluk olgusunun belirlenmesi anlamına gelmektedir. Dijital sistemde herhangi bir nesne tanımlanırken, öncelikle nesnenin özellikleri tanımlanır. Sistemin içerisine nesnenin belirgin özelliklerinden oluşan veri kümesi yüklenir. Bu veri kümesi nesne için öznitelik grupları oluşturur. Bu gruplar nesne sınıflandırmasını sağlar. Sınıflandırma yapıldıktan sonra yapay zekâ yardımıyla nesne tanıma işlemi sonuçlanır.

Günümüz teknolojik gelişmeleri incelendiğinde, özellikle İHA çalışmalarının son yıllarda oldukça ilgi çekici olduğu görülmektedir. İHA teknolojisi ağırlıklı olarak savunma sanayisinde kullanılmasına rağmen sivil hayatta arama kurtarma çalışmaları, orman yangınları tespiti, tarım ürünlerinin büyüme oranı ve jeofizik alanında madenlerin belirlenmesine kadar geniş yelpazede kullanılmaktadır.

İnsansız araçlar günümüzde oldukça önem kazanmıştır. Hava kara ve deniz araçlarının insansız hale getirilmesi konusunda yoğun çalışmalar devam etmektedir. İnsansız araçların öneminin artması insan hayatının korunması ve kolaylaşması için gerekli olduğu kabul edilmekte ve bu yönde uğraşları artırmaktadır. İnsansız hava araçlarının önemi de bu noktada oldukça fazladır. Hava araçları insan hayatında en önemli alanlardan biridir.

Savunma sanayi açısından İHA sistemlerinin kullanılması oldukça fazla alanda üstünlük sağlamaktadır. İHA sistemlerini birçok radar cihazları tespit edememektedir. Tespit açısından hayati öneme sahip olan bu sistemleri, elektromanyetik veya çeşitli özellikte ki radarların tespitte güçlük çekmesi durumundan ötürü görüntüleme ile tespit yoluna gidilmiştir. Görüntüleme işlemiyle tespit edilmesi ise kameranin görüş açısı ve kalitesine bağlı olduğundan bu sistemlerde oldukça iyi kameralarla ve yakınlaştırma işlemleri ile kolaylıkla tespit yapılabilmektedir.

Tüm bu tespit işlemleri nesne tanıma yönünden sabit alıcılarda ise şehir güvenliğinde yüz tanıma, araç plaka tanıma, kurum ve kuruluşlara girişlerde yabancı

cisim tanımlama gibi hemen her alanda kullanılmaktadır. Bu bağlamda yapılan çeşitli çalışmalar aşağıda sunulmuştur.

Neogy vd (1992) tarafından yapılan çalışmada IBM 7535 kodlu robotu ile bir kamera yardımıyla alınan görüntü işleme algoritmaları gürültü filtreleme, kenar algılama ve eşikleme yöntemleriyle iyi bir hızda etkin bir nesne tanıma sağlanmıştır. Buna paralel olarak Chen vd (1996) yapılan çalışmada bulanık yapay sinir ağı ile nesne tanıma önemli ölçüde doğrulukla elde edildiği bildirilmiştir.

Russ (1999) çalışmasında, nesne görüntüsünde oluşturulan matrislerin ortalama değerinde ki standart sapması ve görüntüdeki parsellerin grilik mertebelerini dağılımına ilişkilendirmiştir. Nesne renginin özneliğinde temel ölçütler olduğu belirtilmiştir. Ayrıca görüntü histogramı da görüntüdeki piksellerin grilik mertebelerine ilişkin bağıl sıklık değerini oluşturduğu ifade edilmiştir. Yani gri ton seviyesinin görüntüde bulunma sıklığını aynı zamanda pikseldeki dağılım hakkında bilgi verdiği dikkat çekilmiştir. Bu kaynak nesne özneliklerini belirlemede yardımcı olmuştur.

Belongie vd (2002) şablon eşleştirme ile şekil ve nesne tanıma için yeni bir çalışma sunmuşlardır. Bu çalışmada 300 adet ticari olarak kanıtlanmış marka logolarının bulunduğu veri tabanındaki logolar, birbirileri ile karşılaştırılarak aralarındaki farklılık veya benzerlik oranlarının tespiti üzerine çalışmalar yapmışlardır. Her bir logonun şekil, renk boyut gibi keskin özelliklerini belirleyerek birbirinden farklılıkları ortaya konulmuştur.

Türkoğlu (2003) görüntü işleme ile nesneleri tespit etmenin önemli olduğu kadar birbirlerinden ayrılmasının da önemli olduğunu öne sürmüştür. Nesnelerin görüntü işleme ile ayırt edebilme yeteneği, nesnenin kendine özgü barındırdığı görsel özellikleri yani nesnenin şekli, rengi, boyutları ve histogramı gibi özelliklerinin görüntü işleme sistemleri ile işlenip ayırt edilmesiyle olduğu bildirilmiştir. Bu işlemin gerçekleşmesi için Bulanık c-ortalama, Shannon entropi, Fractal analiz gibi yöntemler kullanılmıştır. Kullanılan bu yöntemler ile önce veriler bölümlere ayrılarak, veriden istenmeyen kısımlar kolaylıkla ayrılır.

Dalal vd (2005) yaptıkları araştırmalarda elde edilen görüntü piksellerin gradyanlarının büyüklüğünü ve yönünü tanımlamaya çalışmışlardır. Histogram Of Oriented Gradients (HOG) olarak adlandırdıkları bu yöntemde pikseller çok küçük bölmelere bölünür ve her bir bölmedeki gradyanların yönleri ve büyüklükleri görüntü üzerinde yatay ve dikey gezdirilen Sobel süzgeci yardımıyla hesaplanarak sonuçlar elde edilmiştir. Bu yöntemde nesnenin öz nitelikleri daha keskin hale getirilir. Bu kaynak öznelik özelliklerinin keskin bir şekilde ayırt edilmesi hakkında bilgi vermiştir.

Ahonen vd (2006) tarafından yapılan çalışmalarında, elde edilen görüntüler parçalara ayırmış ve her bir parçadan ikili örüntüler (Local Binary Patterns, LBP) çıkarılmıştır. Bu çıkarımlar $n \times n$ boyutlarındaki bir matrisin bir görüntü parçasında üzerine geldiği piksellerin, matrisin merkezindeki piksel ile eşitlenmesi sonucunda elde edilmiş ve diğer yöntemlerle karşılaştırıldığında daha iyi sonuçlar alındığı bildirilmiştir. Nesneyi ayırt edecek kısımları bu şekilde parçalayarak daha fazla veriyi daha az yerde karşılaştırdığı için sonuç olarak daha iyi olduğu düşünülmüştür.

Heusch (2007) Ojala ve arkadaşları tarafından yerel ikili örüntü (LBP) ilk olarak doku analizi için tanıtılmış ve LBP, elde edilen görüntülerde ki bütün piksellerin tek tek komşuları ile arasında ikili kodlamaya dayalı ayırt etme kabiliyeti iyi örüntü tanıma yöntemi olduğu sonucuna varılmıştır. Bu örüntü sayesinde pikseller tek tek karşılaştırılır ve nesne tanınması gerçekleşir.

Wu vd (2008) k en yakın komşu sınıflandırması (k-NN) olarak adlandırılan, her pikselin k boyutundaki komşuluğuna göre benzer özelliklere sahip verileri temel alan en basit sınıflandırma algoritması yardımıyla nesne tanıma uygulaması gerçekleştirmiştir. Seçilen bu sınıflandırma sisteminde k ifadesine bağlı olarak değişikliklere neden olmakta ve sınıflandırma sorunlarında oldukça basit ve iyi performansla uygulanmakta olduğunu bildirmişlerdir. Nesne benzerliklerini komşu sınıflarla karşılaştırması nesne tanıma sisteminde hızlı ve doğruluk oranı yüksek bir sistem oluşturmuştur.

Karakaya vd (2009) bildirilerinde HOG algoritması uygulanarak nesne tespit ve tanıma üzerine çalışılmıştır. Nesnenin piksellerinde gradyanlarının büyüklük ve yönü üzerine çalışılan sistemlerdir. Bu sistemlerde Altera Cyclone II FPGA içeren Terasic DE2

bordu gibi yöntemler uygulanmıştır. Bu çalışmaya özgü yaptıkları 125 adet nesnenin üzerinde tanıma yönteminde %100 bir başarı elde etmiş oldukları bildirilmiştir.

Yiğit ve Temizel (2010) çalışmalarında Gaussian Karışım Modeli (GMM) yardımıyla nesne tanıma gerçekleştirmişlerdir. Bu sistemde gizli haberleşme olgusunun varlığı irdelenmektedir. Nesnenin başka konumlardaki görüntülerini önceden öğrendiği görüntülerle karşılaştırma yaparak karar vermesi ile sağlar. Tespit edilen nesnenin şablonu bir diğer görüntüleme yani termal kameradan alınan görüntü ile birleştirerek tespit edilen nesnenin canlı veya cansız olduğuna karar vermişlerdir. Bu yöntemle, var olan cansız nesnenin ne olduğunun tespiti yapılarak, şüpheli paket uyarısı vermesi sağlanmıştır.

Zheng vd (2010) tarafından yapılan çalışmada, çift katmanlı aşağı ve yukarı yol birleştirilmeli yapay sinir ağı (FBTP- NN) ve üç katmanlı ileri beslemeli yapay sinir ağı (FF-NN) ile yapılan nesne tanıma uygulamalarında %90 - %98 başarı sağlandığı bildirilmiştir. Görüntü işleme de yapay sinir ağlarının kullanılmanın ne kadar doğru olduğunu irdelemiş ve yöntem açısından bilgi edinilmiştir.

Hartl vd (2010) tarafından yayınlanan çalışmada, günümüzde kullanılan akıllı telefonlar yardımıyla nesneleri bölümleme ve tanımlama algoritması olarak, nesneleri tespit ve tanımlama için PC görüntüsü (computer vision CV) temelli teknikler geliştirilmiştir. Yapılan bu çalışmada yazılım olarak C ve C++ kullanılmış donanımsal olarak 416 MHz sabit CPU ve 64 - 256 MB RAM'a sahip flash bellek ve 2 MP otomatik odaklanma özelliğine sahip kamera kullanılmıştır. Sonuç olarak oluşturulan yazılım ve donanım ile nesne tanımlama algoritması anlamında % 89 oranında başarı sağlamıştır.

Altunkurt ve Kahrıman (2011), çalışmalarında, hareket halinde önceden belirlenmiş nesnenin, kamera önünden geçerken görüntünün kamera ortasında kalması için kamerayı adım motor ile döndürme hareketi sağlamışlardır. Çalışmalarında nesnenin farklı ışık şiddetlerine göre renk aralıklarını çıkararak, maksimum-minimum RGB değerlerine göre arka plan oluşturulmuş ve bu sayede nesnenin konumu tespit edilmiştir.

Karakoç (2011) tezinde görüntü içinde görüntü arama sorunu üzerine çalışmıştır. Bu amaçla, görüntüyü aramak için genetik algoritmalar ve tespit işlemi için ise yapay sinir ağları gibi çeşitli yöntemler kullanmıştır. Aranan görüntü diğer görüntüler içerisinden seçilip çıkarılması işlemi yapay sinir ağları yardımıyla yapılır. Veriler karşılaştırılıp aranan veri kümesine en yakın görüntünün elde edilmesi hız ve doğruluk açısından oldukça önemlidir. Sonuç olarak, aynı doğrultuda arama yöntemlerinin yalnız bir iş parçacıklı aramalara göre daha başarılı olduğu görülmüştür. Arama yöntemlerinde yapay sinir ağı kullanmak ise manuel arama yöntemlerine göre daha başarılı sonuçlar ürettiğini görülmüştür.

Hanbay ve Üzen (2017) nesne tespiti için yapılan çalışmalar, temel işlemlerden başlanıp karmaşık yapılar içeren sistemlere doğru yapılmıştır. Kullanılan yöntemler arasındaki fark optik akış, arka plan modeli çıkartma gibi özelliklere dayanarak gruplandırılabilirler. Elde edilen görüntülerdeki nesne tespiti için görsel nesne niteliği yani renk, şekil gibi fiziksel özellikler ve hareket bilgileri kullanılmaktadır. Bu bilgiler çeşitli yöntemlere göre tek tek ya da birlikte işlenerek nesne tespitinin yapılmakta olduğu bildirilmiştir.

Solak ve Altınışik (2017) çalışmalarında, ortamda tespit edilen nesnelerin, kendilerine özgü özelliklerinin belirlenmesi ve birbirinden ayrılmasına yönelik çalışmalarında, üç aşamalı bir sınıflandırma sistemi önermişlerdir. Sınıflandırma sisteminde ilk aşamada kameradan alınan görüntü bir ön işlemden geçmeli, ardından nesnelerden elde edilen özellikler veri tabanına aktarılmalı ve son adımda sınıflandırma gerçekleştirilmesi gerektiğini bildirmişlerdir.

Toprak (2019) herhangi bir araştırmanın etkili sonuçlar verebilmesi için elde edilen görüntülerin verdiği bilgiler eşit miktarda olmalıdır. Bundan dolayı bütün görüntüler belirli boyutlarda elde edilmeli veya boyutlandırılmalıdırlar. Kontrast dengesi, önemsiz bölgelerin kırılması gibi basit işlemler yapılarak boyutlandırma yapılabilmektedir. Ayrıca tavsiye edilen sistemlerin uygulama esnasında doğruluklarını ve sonuç verme performanslarını analiz etmek için bazı hata tespit yöntemlerinin kullanılması gerektiğini bildirmiştir.

3. MATERYAL ve YÖNTEM

Bu bölümde yapay zekâ, evrişimsel sinir ağları ve modelleri tanıtılarak çalışmanın temeli oluşturulmuştur. Daha sonra Alexnet, Resnet, lenet-5 ve googlenet modellerinin tanımlanması yapılmıştır. Elde edilecek görüntülerin işlenmesi için özellikle SSD mobilnet ve tensorflow kütüphaneleri kullanılarak çalışmanın yazılımsal bölümü tamamlanmıştır. Çalışmanın diğer bölümü olan donanımsal yapı step motor, Raspberry pi4, Pi kamera ve güç kaynakları tanıtılmıştır. Son olarak donanımsal ve yazılımsal argümanlar kullanılarak görüntünün alındığı dokunmatik ekran tanıtılmıştır.

3.1 Sistemlerde Yapay Zekâ

Beyin canlıların yaşam faaliyetleri düzenleyen bir organdır. Zekâ ise öğrenme ve uygulama alanında karşımıza çıkan olgudur. İnsanı farklı kılan öğrenme, öğrenilenden faydalanabilme, yeni ve farklı olaylara uyum sağlayabilme, sorunlara çözüm üretebilme ve bilmediği durumları merak edip sonuca ulaşabilme yeteneği olarak adlandırılan zekâyâ sahip olmasıdır. İşte bu zekâ olgusunun sadece insanda değil dijital ortamda da kullanılması için yapılan çalışmalara yapay zekâ adı verilmiştir.

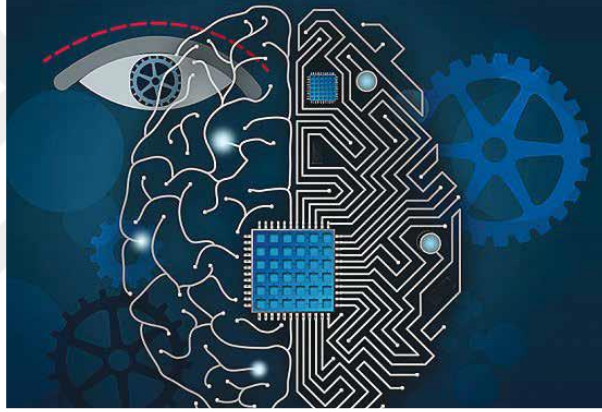


Şekil 3.1. Yapay zekâ uyarlaması (Adalı vd 2017)

Bilgisayar sistemlerinin günümüz dünyasında sadece dışarıdan verilen komutları yerine getirmesi değil, insan zekâsına özgü özellikleri taklit etmesi de beklenmektedir. Buna göre bilgisayarın, zekâ özellikleri sergilemesi olayına yapay zekâ denilmektedir. Yapay zekânın doğuşu 1956 yılında yapılan Dartmouth konferansı olarak kabul edilmiş

ve bu konferansa katılanlar yapay zekânın öncüleri olarak kabul edilmiştir. Şekil 3.1’de temsili olarak insan zekâsı ile dijital olgunun birleşmesi gösterilmektedir. Ayrıca yine aynı yılda ilk defa yapay zekâ terimi John McCarthy tarafından kullanılmıştır (Russel and Norving 2009).

Günümüzde yapay zekanın entegre edildiği ortamlar da kendisinden beklenen en çok on reaksiyon; tabii dil oluşturabilme, yapılan konuşmaları algılayabilme, makine ve teçhizat öğrenimi gerçekleştirebilme, sanal araçlar sağlama, donanım optimizasyonu sağlayabilme, karar verebilme, derin öğrenme alanları, biyohareketleri algılama, robotik süreç yönetimi ve herhangi bir metin analizi yapabilmesidir. Daha önceden öğrenilmiş durumları algoritmaları ile süzüp istenilen sonuçları göstermelidir.



Şekil 3.2. Yapay zekânın insan beyni gibi uyarlanması (Adalı vd 2017)

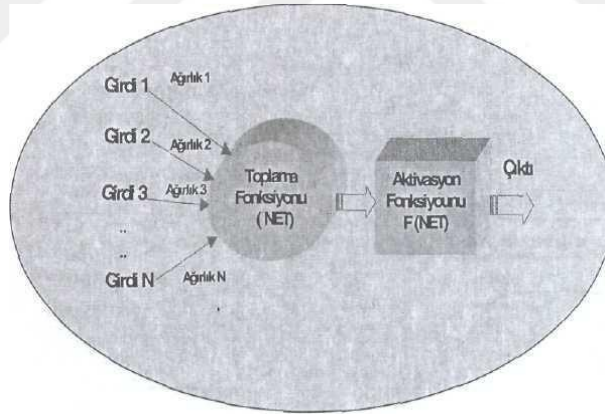
Evrişimsel sinir ağları, makine öğrenmesi ve yapay zekâyâ dayanan veriler ışığında bilgi üretilmesine olanak sağlayan bir sistemdir. Özetle derin öğrenme, makine öğrenmesinin makine öğrenmesi ise yapay zekânın alt dalıdır. Şekil 3.2 ‘de temsili olarak gösterilen insan beyni-dijital benzetim, elektronik ortamda elde edilmek istenen bilgiyi veya bir problemi çözmek için programlan bir algoritma kullanılmalıdır. Yapay zekânın öğrenimi de bu şekilde gerçekleşerek, giriş verisini çıkış verisine çevirerek bir yönerge oluşturur ve sonuç çıktısı verilmektedir. Bir diğeri derin öğrenme, elde edilen veri seti ile sonuçları tahmin eden birden çok katmandan oluşan bir makine öğrenme yöntemidir. Son olarak yapay sinir ağları, insandaki sinir sistemi ve beynin çalışma prensibinde karar verme işleyişini matematiksel olarak modelleme ve böylelikle bir makinanın insan gibi düşünme ve karar verebilmesi hedeflenmektedir (Alpaydın 2011).

Günümüzde, yapay zekâ tabanlı yöntemler; başta savunma sanayi olmak üzere havacılık ve uzay sanayisi, otomotiv sanayisi, iletişim alanı, sağlık endüstrisinde, spor ve eğlence gibi günlük yaşamımızın hemen her alanında sürekli artan bir şekilde kullanılmaktadır.

3.2. Evrimsel Sinir Ağları ve Modelleri

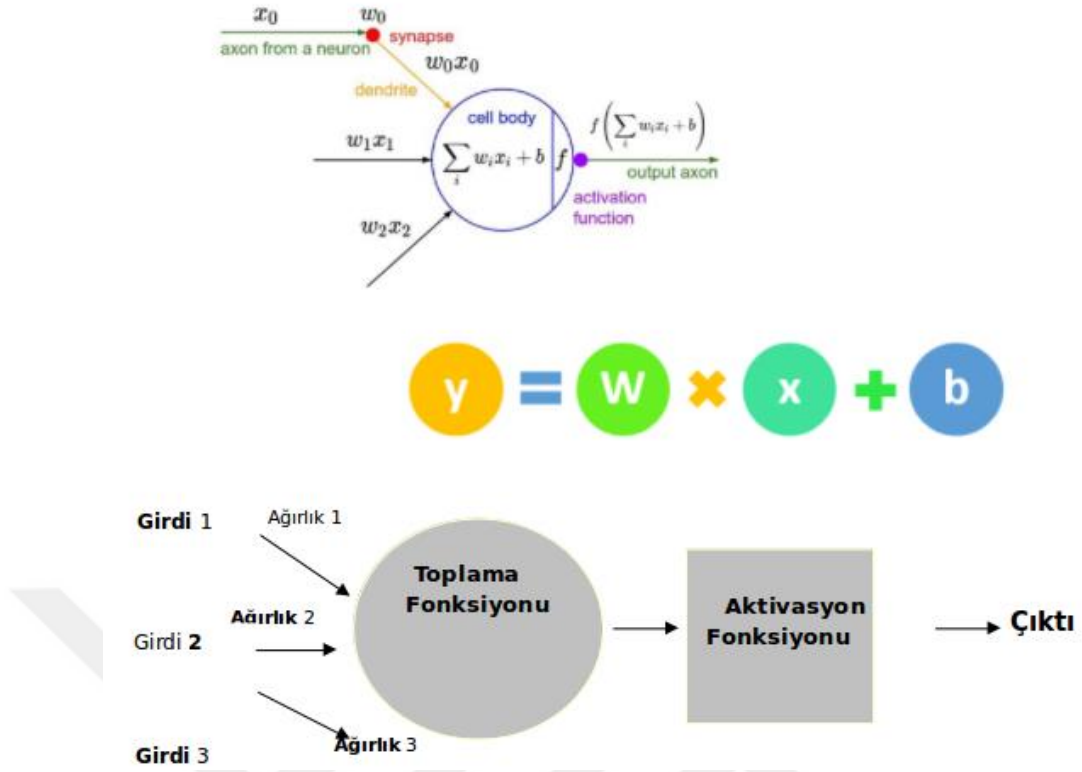
3.2.1 Evrimsel sinir ağlarını oluşturan katmanlar

Yapay zekâ oluşturmak için biyolojik sistemlerde bulunan sinir ağlarının sinir hücreleri olduğu gibi yapay zekânın yapay sinir ağlarının yapay sinir hücreleri bulunmaktadır. Yapay sinir hücreleri bu alanda süreç elemanları şeklinde adlandırılmaktadır. Sinir hücrelerinde bulunan her süreç ögesinin 5 esas ögesi bulunmaktadır. Şekil 3.3’de gösterildiği gibi bu esas ögeler; Girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılarıdır (Ataseven 2013).



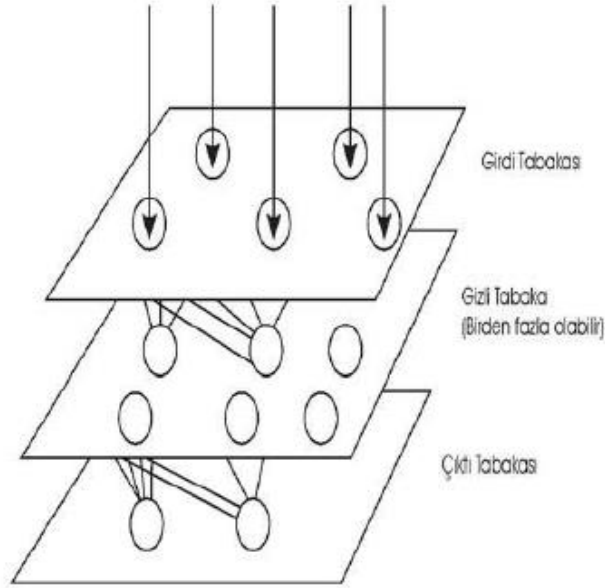
Şekil 3.3.Yapay sinir hücresinin yapısı (Ataseven 2015)

İnsan beynini modelleyerek oluşturulan yapay sinir ağları (YSA) beyin işleyişinin karar mekanizmalarının Şekil 3.4’de gösterildiği gibi matematiksel bir şekilde dijital ortama uyarlanması olarak tanımlanabilir. Yapay sinir ağları, bilgisayar veya benzeri makinaların insan beyninin çalışma prensiplerini örnek alarak, insan zekâsının yapabileceklerini yapmaya çalışması aşamalarıdır. Böylece bilgisayar ve makinelerin insan gibi düşünmesi ve karar alması hedeflenmektedir (Kubat 2015).



Şekil 3.4.Yapay sinir ağlarının matematiksel modeli (Adalı 2018)

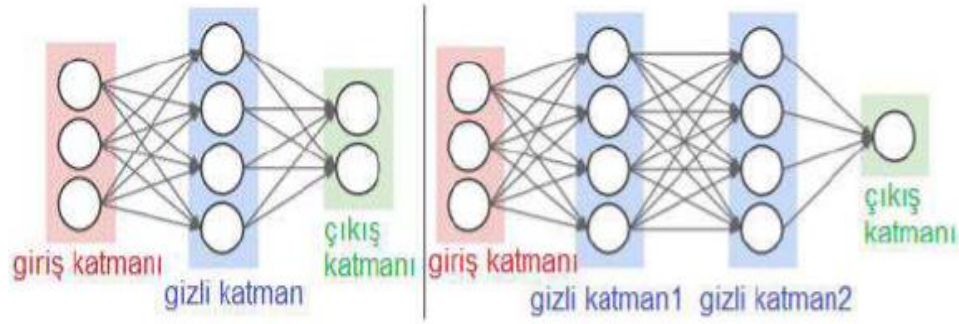
Yapay zekâyı oluşturmak için modellenen yapay sinir ağları, yapay sinir hücrelerinin katmanlar şeklinde birbirine bağlanması ile meydana gelen veri tabanlı sistemlerdir. Yapay sinir hücreleri basitçe katmanlar halinde gruplandırılırlar ve oluşturulan bu tabakalar birbirleriyle ilişkilendirilirler (Koç vd 2004). Genelde tüm yapay sinir ağları Şekil 3.5’de gösterildiği gibi birbirlerine benzerlik gösterirler. Görüldüğü üzere yapıda bazı nöron girdilerini, bazı nöron çıktılarına iletmek için dış ortamla ilişkilendirilirler. Bazı nöronlar ise gizli tabakalar oluşturarak kendi içlerinde izole şekilde haberleşir (Ataseven 2013).



Şekil 3.5. Yapay sinir ağlarının genel yapısı (Yurtoğlu 2005)

Yapay sinir ağları uygulamalarında, genellikle girdi tabakası, gizli tabaka ve çıktı tabakası olmak üzere en az 3 tabakaya ihtiyaç duyulmaktadır. Girdi tabakası dış ortamdan girdileri alan ve bir sonraki tabakaya ileten nöronları içermekte, çıktı tabakasında işlenmiş verileri dışarı ileten nöronlar bulunmaktadır. Şekil 3.6’da katmanlı sinir ağlarının şematiği gösterilmiştir. Girdi ve çıktı tabakaları arasında birçok gizli tabaka ve bu tabakaların birçok nöronu bulunabilmektedir. Gizli tabaka içerisinde bulunan nöronlar sadece tabaka içerisinde bulunun diğer nöronlarla bağlantılı sinyalleri alırlar.

Yapay sinir ağının yapısında sinir hücrelerinin sayısında farklılıklar olmasına rağmen, bu sinir ağının oluşturulmasında geçerli olan herhangi bir kural yoktur. Genelde de giriş katmanı ve çıkış katmanı sayısı girdilerin sayısına eşit olurken asıl sorun gizli katman sayınsındadır. Gizli katman sayısının az olması karışık sorunların çözülmesinde yetersiz kalırken sayının çok olması da kararsızlıklara neden olmaktadır. Katmanlarda bulunan nöronların sayısı ise deneme yanılma yöntemiyle belirlenmektedir (Detienne 2003).



Şekil 3.6. Bir ve birden fazla katmanlı yapay sinir ağı modeli (Lecun 1998)

Yapay zekânın altında tıpkı beyin gibi sinir ağları bulunmaktadır. Bu ağlara makine dilinde evrişimli sinir ağları denmektedir.

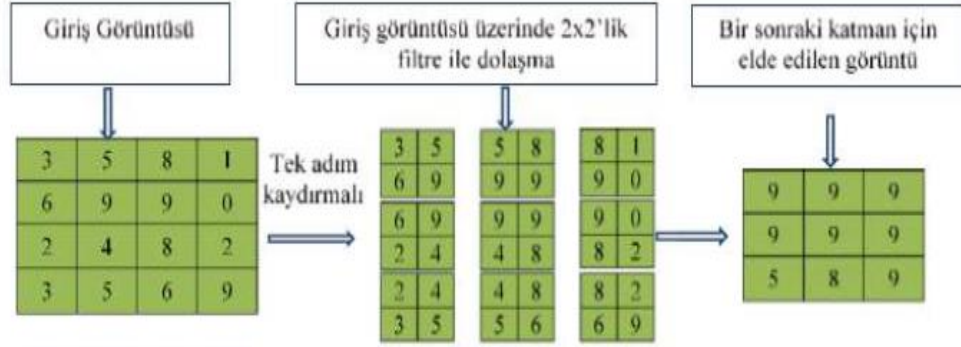
Evrişim sisteminin ana amacı giriş verileri olan ses, görüntü veya metinde verilen bilgilerin veya özelliklerin sonuçlandırılmasıdır. Giriş olarak verilen değerler matrisi olarak düşünülürse, hedef olarak matrisi belirli filtreler ile tarayıp istenilen giriş nesnesi için belirleyici özellikler çıkarmaktır. Bu işlem için evrişimsel işlemi oluşturan katmanları;

Giriş Katmanı; Giriş katmanı evrişimsel sinir ağlarının (ESA) ilk katmanını oluşturmaktadır. İşlenmemiş ham halde gönderilen verinin boyutu (görüntü için çözünürlük), tasarlanacak modelin performansı açısından oldukça önemlidir (Long et al. 2015).

Konvolüsyon Katmanı; Konvolüsyon katmanı ESA’da dönüşüm katmanı olarak tanımlanmaktadır. Dönüşümün esas prensibi belirlenen bir filtrenin görüntü üzerinde tarama yaparak yeni bir matris meydana getirmesidir. Belirlenen filtre tarafından girdi olarak gelen görüntüye konvolüsyon işlemi uygulanarak özellik haritası oluşturularak bir çıkış verisi elde edilir (Long et al. 2015).

Doğrusal Birim Katmanı; Aktifleştirme katmanı olarak da bilinen doğrusal birim katmanı, konvolüsyonel katmanlardan sonra gelen ve ESA nöronlarının çıktıları için en fazla kullanılan katmandır. Bu katman, önceki katmanda doğrusal olmayan durumu doğrusal duruma çevirmek için kullanılır (He et al. 2016).

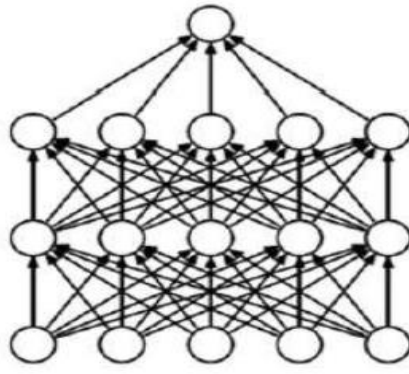
Havuzlama Katmanı; Havuzlama katmanı Şekil 3.7’de gösterildiği gibi doğrusal birim katmanından sonra kullanılmaktadır. Havuzlama katmanının ana amacı, kendinden önce gelen katmanın işlem yükünü azaltmak, işlemi hızlandırmak ve daha fazla doğru oranda gerçekleştirmeyi sağlamaktır (Boureau et al. 2010).



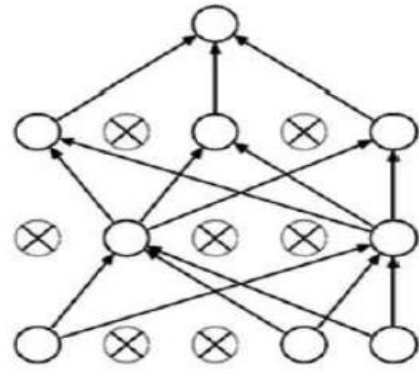
Şekil 3.7. Bir adım kaymalı 5x5’lik giriş görüntüsüne 2x2’lik filtre uygulanarak havuzlama işlemi (Boureau 2010)

Tam Bağlantılı Katman; Bahsedilen katmanlardan sonra gelen tam bağlantılı katman (FC), her bir girişin tüm nöronlara bağlı ESA mimarisinin sonunda yer alır ve sınıf skorları gibi hedefleri optimize etmek için kullanılır (Srivastava et al. 2014).

Dropout Katmanı; Şekil 3.8’de gösterilen standart ESA ağı kullanım aşamasında çok kez tekrar yaptığı için bu sinir ağı ezberleme yapar. Yani giriş de ve çıkışta alınacak cevap bilinmekte ve bunu tekrarlamaktadır. Ezberleme olayı evrimsel sinir ağlarında olmaması gereken bir durumdur. İşte bu ezberlemenin önüne geçebilmek için kullanılan katman dropout katmanıdır. Bu katmanla ağıdaki bazı noktalar dışarıda bırakarak ağların yeri rastgele oluşmuştur. Bu rastgele sistemi ile ezberlemenin önüne geçilmiştir. (Srivastava 2014).



(a) Standart ESA ağı



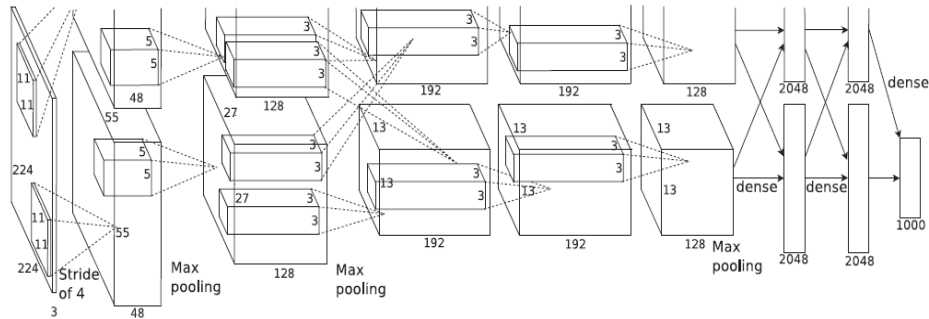
(b) Dropout Katmanı kullanılmış ESA ağı

Şekil 3.8. (a) Standart ESA ağı (b) ESA ağına drop uygulanması (Srivastaya 2014)

Sınıflandırma Katmanı; Sınıflandırma katmanı tam bağlantı katmanından sonra gelen katmandır. Sınıflandırma işleminde genelde SoftMax kullanılır. Sınıflandırma yapılacak nesne sayısı kadar katman çıkış değeri olmalıdır (Krizhevsky et al. 2012).

3.2.2 AlexNet modeli

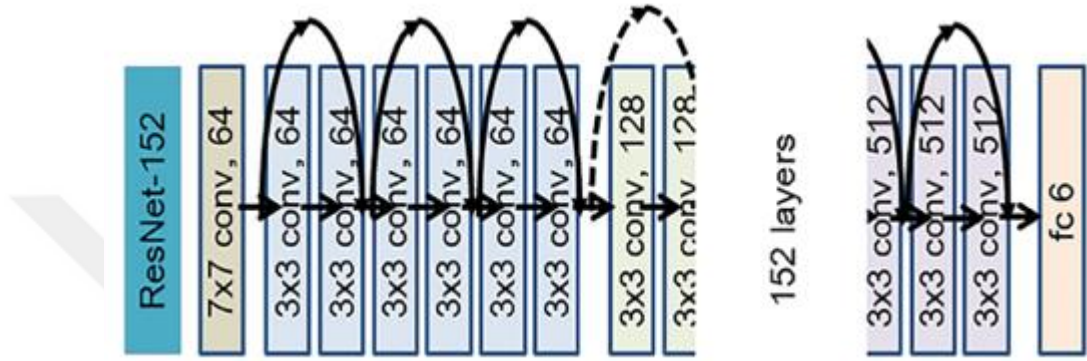
AlexNet aktivasyon işlevi, katmanlarında çok fazla kümelendirme kullanılmaktadır. Bu kümelendirme işlemini başarılı şekilde yapmasından dolayı görüntü işleme sistemlerinde kullanılmaya başlanmıştır. Bir yarışma da Ortalama 60 milyon değişkenin kullanıldığı AlexNet, Şekil 3.9’da görüldüğü üzere, paralel çift bilgisayar üzerinde nesne tanıma doğruluk oranı % 74.3’ten % 83.6’ya çıkarılan ilk modeldir (Bai et al. 2017).



Şekil 3.9. AlexNet modeli (Bai 2017)

3.2.3. ResNet Modeli

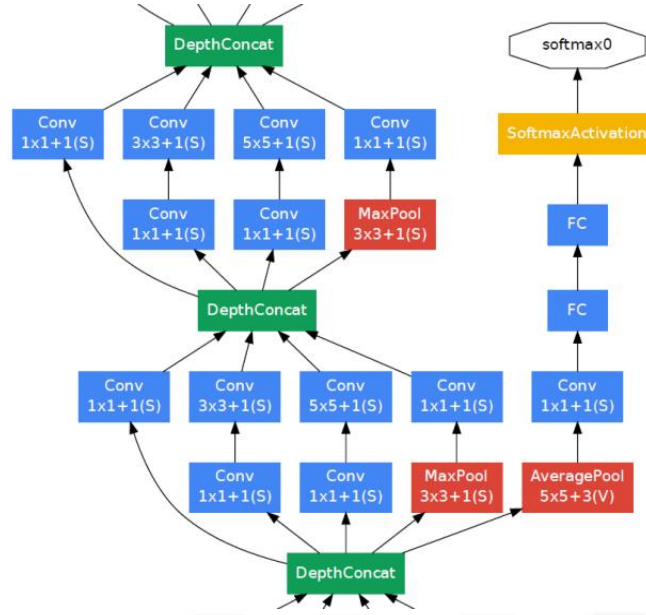
ResNet modeli yaklaşık 152 katmandan oluşmaktadır. Şekil 3.10’da görülen bu model, diğer modeller içerisinde en derin tasarlanan modeldir. ResNet modelinin diğer modellerden ayıran en önemli farkı, artık değerlerin daha sonraki katmanlara besleyen blokların model üzerine eklenmesiyle meydana getirilmesidir (He et al. 2015).



Şekil 3.10. ResNet modeli (Seung et al. 2017)

3.2.4. GoogLeNet modeli

Şekil 3.11’de gösterilen GoogLeNet modeli, diğer modellere göre daha karmaşık bir yapıya sahip olmasına rağmen, değişken sayısının az olması uygulama yükünü düşürmektedir. Bu model 3 tane çok katmanlı algılayıcı oluşumu ile ESA ve ortalama kümelenendirme fonksiyonlarından meydana gelmektedir.

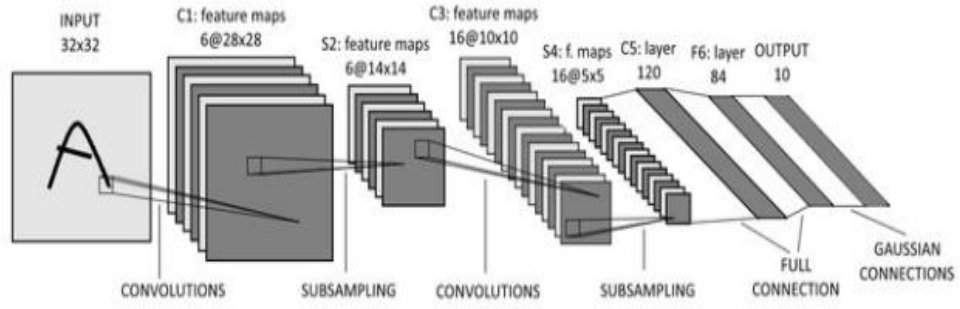


Şekil 3.11. GoogLeNet (Szegedy 2016)

GoogLeNet modelinde sistemin ezberleme durumunu düşürmek için paralel bağlı parçalar bulundurulmaktadır. Bu modele göre veri çoğaltma ve veri azaltma uygulandıktan sonra nesne tanıma başarısının veri ortaklamaları için arttığı tespit edilmiştir. Ayrıca bu model 5 milyon değişken ile 60 milyon değişkene sahip AlexNet modeline göre yaklaşık 12 kat daha az uygulama yüküne tabidir (Szegedy et al. 2016).

3.2.5 Lenet-5

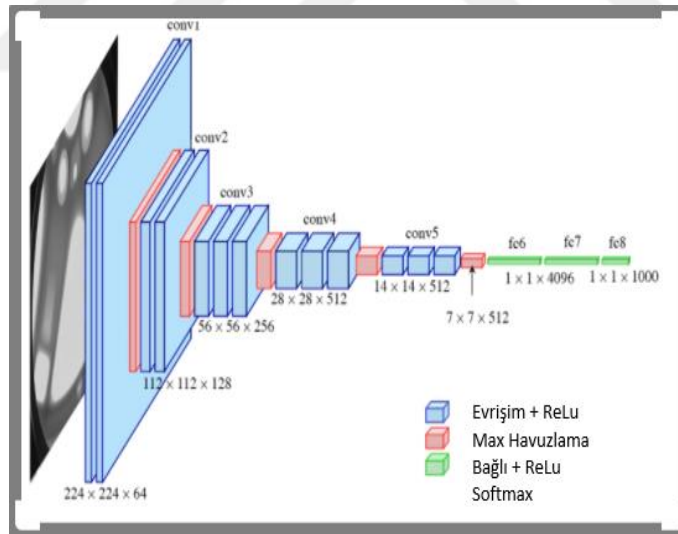
Evrişimsel sinir ağları modelleri arasında ilk başarı sonuçlar veren Lenet-5 modeli Şekil 3.12’de gösterilmiştir. Lenet-5 modelini diğer modellerden ayıran önemli fark, boyut düşürmesi adımlarında azami kümelenme yerine ortalama kümelenme işleminin kullanılmasıdır. AlexNet ve GoogLeNet modellerine göre çok düşük yaklaşık 60 bin değişken hesaplanmaktadır (Lecun et al.1998).



Şekil 3.12. Lenet-5 modeli (Golovko 2015)

3.2.6 VGG

VGG modeli diğer modellere göre basit olmasına rağmen Şekil 3.13’de, yaklaşık 138 milyon değişken ile hesap yapmaktadır. Diğer modellerden en önemli farkı ikili ya da üçlü evrişim katmanlarını ortaklama katmanlarının izlemesidir (Simonyan and Zisserman 2014).



Şekil 3.13. VGG modeli (Ferguson 2017)

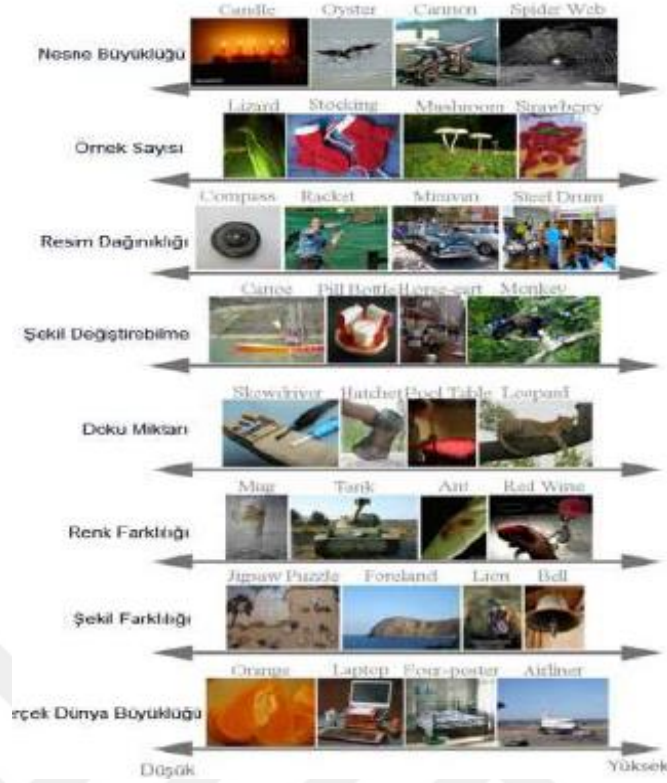


Şekil 3.15. MNIST veri seti

Bu veri seti 60000 tane eğitim verisi ve 10000 tane test verisi ile Şekil 3.15’de gösterildiği gibi 28x28 boyutlarında ve siyah arka plan üzerine el ile yazılmış rakamlardan meydana gelmektedir (Lim et al. 2016).

3.3.3 ImageNet veri seti

Şekil 3.16’da gösterilen ImageNet seti görüntü sınıflandırmada, bir nesnenin konumunu belirlemede ve nesne tanımda kullanılmaktadır. ImageNet veri setinde 50000 doğrulama verisinin yanı sıra 100000 test verisinden oluşan settir.



Şekil 3.16. ImageNet veri seti çeşitliliği (Tan 2019)

3.4 Veri İşleme Modelleri

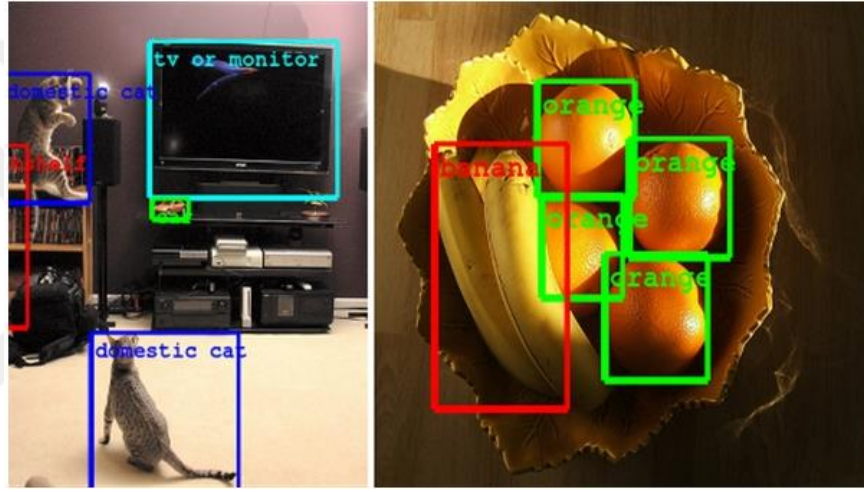
Nesne tanıma işlemi belli bir sistem çerçevesinde gerçekleşir. Nesnenin bir çok konumda görüntü kümesi ve bu görüntü kümesinin makine diline çevrilmiş hali görüntü işlemenin olmazsa olmazıdır. Sistem bu durumda görüntü içerisinde ne aradığını bilecek ve sürekli yeni görüntüler ile karşılaştırma yaparak karar verecektir. Tüm bu işlemler, nesne tanıma label map, tensörflow ve ssdmobilnet kısımlarından oluşmaktadır ve aşağıda gösterilmiştir.

3.4.1 Nesne tanıma

Son yıllarda nesne tanıma üzerine yapılan birçok çalışma yapılmış ve oldukça ilerleme sağlanmıştır. Nesne tanımlaması yapılırken belirlenen kategoriler esas alınarak elde edilen görüntüler üzerinde tarama yapılmaktadır. Şekil 3.17’de bir görüntüde nesne tanınması yapılırken bir sınıflandırma yapılmaktadır. Bu görüntü üzerinde görüntü kümesine göre nesneler sınıflandırılmaktadır. Ayrıca bir veya birden fazla aynı nesne var

ise nesnenin konumundan ve nesneye ait özelliklerinden görüntü taraması yapılarak nesne sınıfına göre belirlenmiş olmaktadır.

Yapılan bu çalışmada, aşağıda detaylıca anlatılacak donanımlar yardımıyla hazırlanan İHA'dan fotoğraflar çekilerek resimler veri olarak toplandı. Toplanan veriler Label Image programıyla ayrı ayrı etiketlenerek görüntü işlemeye uygun hale getirildi. Görüntü işleme için SSD mobilnet görüntü işleme algoritması kullanarak modeller eğitildi. Gerekli kodlar python ve arduino programına hazırlandıktan sonra sonuçlar irdelendi.



Şekil 3.17. Nesne tanıma için örnek görüntü

Tanımlanacak nesnelerin bölge önerisi, bir görüntüde olan değişik boyutlardaki nesneleri tanıırken farklı boyutlarda pencere oluşturulur. Her defasında görüntü üzerinde pencere kaydırma yöntemiyle her bir bölge farklı renk tonuna göre işaretlenerek tespit edilmektedir. Şekil 3.17 de portakallar yeşil etiket muzlar ise kırmızı etiketle aynı görüntü işleme bölgesinde ayrı ayrı sınıflandırılmıştır (Girshick et al. 2014).

3.4.2 Label map

Bir nesnenin tanımlanabilmesi için öncelikle sistemin bu nesne hakkında bilgi sahibi olması gerekir. Bu bilgiyi de sisteme ancak nesne hakkında ayırt edici özellikleri belirleyerek makine diliyle sisteme öğretmektir. Nesne tanıma için eğitim modeline giriş olarak verilecek veri dosyası, bir ID tanımlaması yapılarak oluşturulur. veri dosyasına

verilen bu tanımlama göre nesnenin sınıfı belirlenmektedir. Bu sınıflandırma sayesinde sistem nesnenin ne olduğuna karar vermektedir (Tan 2019).

3.4.3 TensorFlow model eğitimi

Sinir ağlarından oluşan bu alanda TensorFlow, açık kaynak kodlu Google tarafından geliştirilen derin öğrenme (deep learning) kütüphanesidir. TensorFlow'un esnek çalışma şartları sayesinde alan sınırlaması olmadan hesaplamaları bir veya birden çok CPU, GPU kullanarak derin öğrenme çalışmaları yapılabilir. Ayrıca TensorFlow araştırmacılara farklı SSD ve R-CNN modellerini de sunmaktadır (Lin et al. 2014).

Model eğitimi;

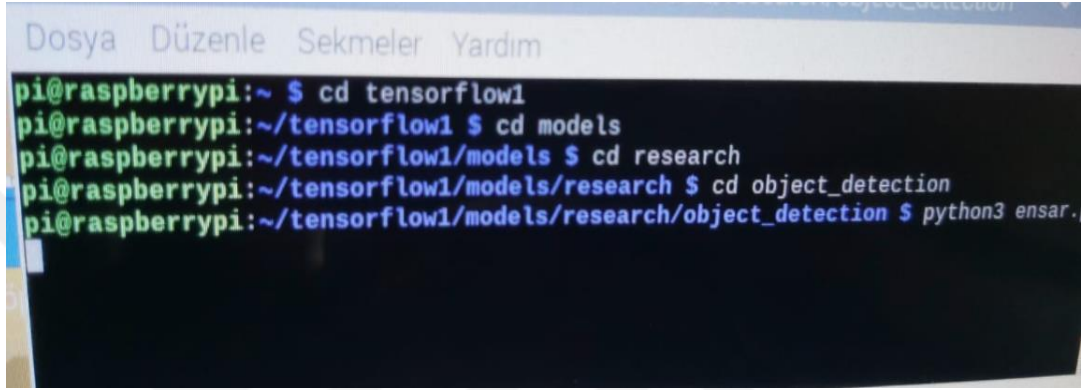
Bu çalışmada, model eğitimi raspberry pi4 içerisinde tensorflow sayesinde gerçekleştirilmiş model eğitimi kullanılmıştır. Tensorflow, eğitim aşamasında İHA modeline ait ilk çalışmalarda 600 adet görüntü yüklenmiş fakat bu eğitim başarı açısından eksik kalmıştır. Bu durum, kalemleri çapraz tutup kamera açısına girildiğinde kalemleri bir İHA olarak algıladığı görülmüştür. Bu durumda eğitimi yenileyip 1200'den fazla görüntü ile yaklaşık 36 saat eğitimden sonra başarı yüzdesi oldukça arttırılmıştır.

Eğitim aşamaları;

Grafiksel işlemler için özellikle geliştirilen GPU, CPU ya göre daha hızlı olduğu için GPU üzerinden eğitim yaptırılması daha sağlıklı olmuştur. Bilgisayar özellikleri olarak yüksek ekran kartına sahip RTX2060 32 GB ekran kartı sahip olması ile büyük avantaj sağlanılmıştır. Test etme kodları aşağıdaki ifadeyle hello tensorflow çıktısını vermektedir. Bu durumda tensorflow devreye girmektedir. Şekil 3.18'de başlangıç kodları verilmiştir.

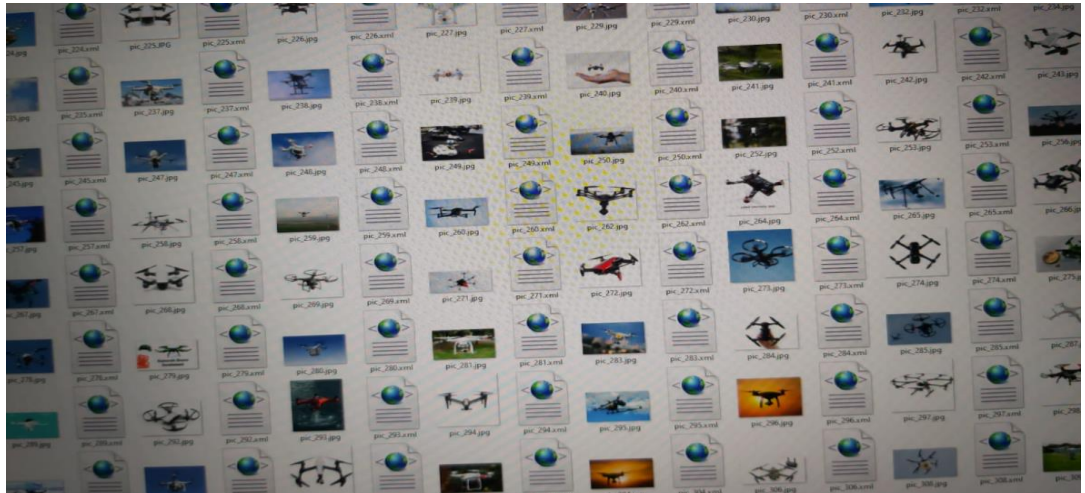
(Tensorflow- gpu)

```
>>> import tensorflow-gpu as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

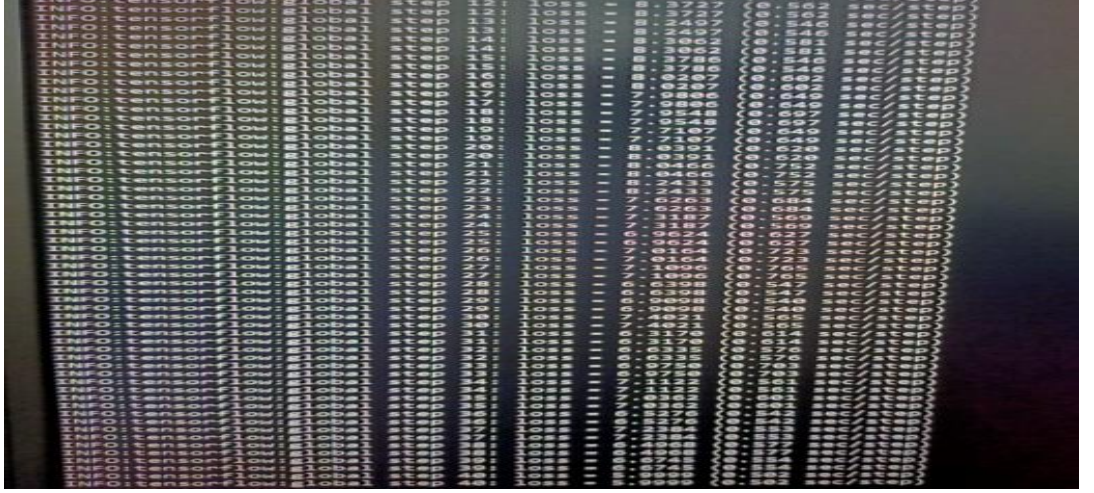


Şekil 3.18. Sistem başlangıç kodları

Bu çalışma da tensorflow model eğitimi için ilk olarak İHA resimleri Şekil 3.19’da gösterildiği gibi veri olarak toplandı. Bu verileri makine diline çevrilmek için labeling programında etiketleme yapıldı. Daha sonra bu etiketlenen fotoğrafları görüntü işlemeye uygun hale getirmek için tüm verileri bir csv dosyasında toparlanıp görüntü işlemeye hazır hale getirildi.



Şekil 3.19. Toplanan İHA görüntüleri ve etiketlenmesi

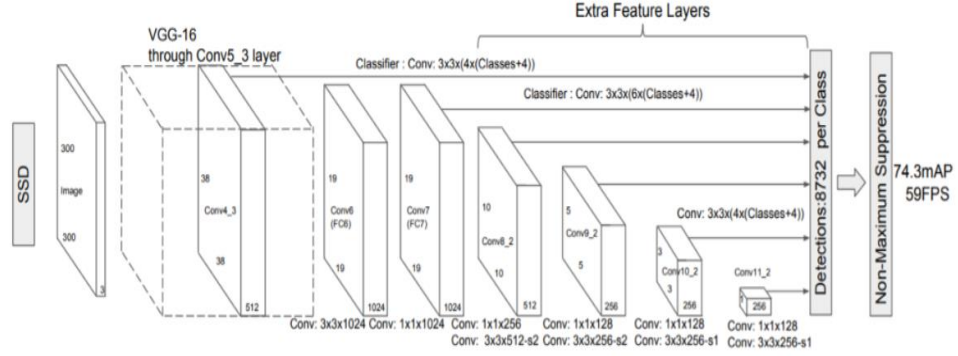


Şekil 3.20. Eğitim sürecinden bir aşama

Yapay zekâ ve sinir ağları çalışmalarında Şekil 3.20’de en önemli olaylardan biri sistemin öğrenme kapasitesidir. Sisteme önceden öğretilen görüntüler sayesinde, canlı nesne görüntüleri elde edilmiş ve sistemin bu nesne görüntülerini birbirleri ile karşılaştırması sağlanmıştır. Bu karşılaştırma sürecinde yapay zekâdan beklenen genelleme yapma, karar verme ve tahmin etme mekanizmalarıdır. Şekil 3.19’da görülen eğitim aşaması etiketlenen 1200’ den fazla görüntü, 36 saatten fazla süre boyunca tensorflow eğitimine sokulmuştur.

3.4.4 SSD mobilnet

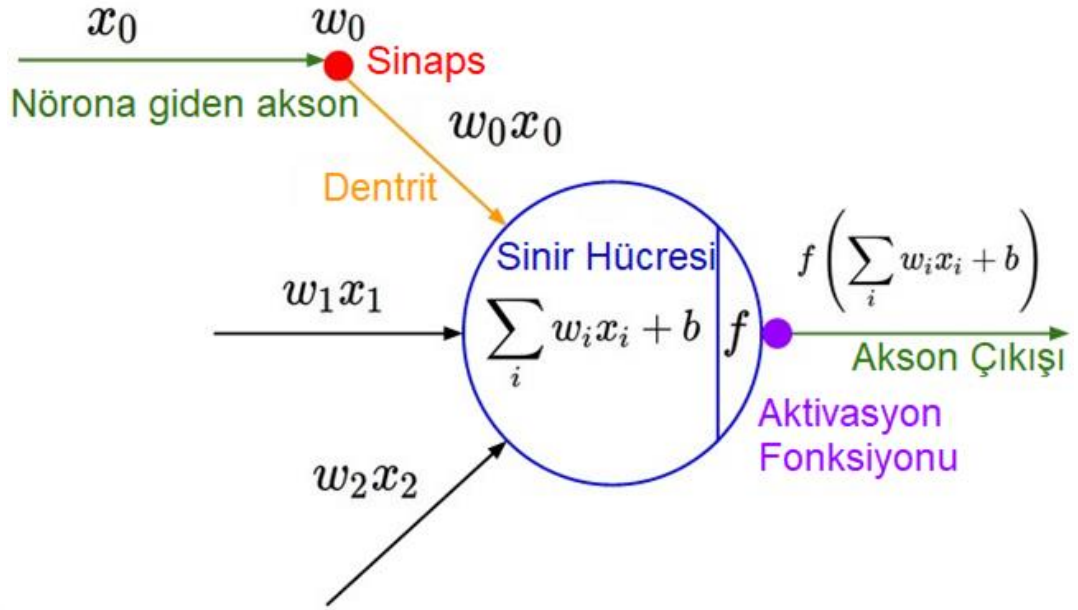
Single Shot MultiBox Detector (SSD) ile nesne tanımlamasını, muhtemel bölgeler belirlenip sınıflandırma yaparak tek seferde nesne tanıma gerçekleştirmektedir. Şematik olarak gösterilen Şekil 3.21’de SSD sistemi, başlangıçta giriş olarak alınan görüntü verileri sinir ağından geçirilerek değişik boyutlarda öz nitelik haritaları meydana getirilmektedir. Bütün özellik haritalarında 3x3’lük filtre vasıtasıyla dikdörtgenler meydana getirilmektedir. Bu dikdörtgenler için nesne sınırlarını ve sınıflandırmalarını belirlerler. Bu dikdörtgenler görüntü içerisindeki küçük ve büyük nesneleri tanımlayabilmektedir (Liu et al 2016).



Şekil 3.21. SSD modelin yapısı (Liu et al. 2016)

Yapay sinir ağları canlıların sinir hücreleriyle benzerlik gösterir. Canlı sinir hücreleri birbirleri ile bağlantılı olup küçük elektriksel akımlar sayesinde haberleşmektedirler. Bu haberleşme hiyerarşi sırasında görme, duyma, öğrenme, ezberleme gibi faaliyetler gerçekleştirilir. Tıpkı bu canlı sinir hücreleri gibi yapay sinir ağlarının da aralarında haberleşme sağlaması gerekmektedir. Bu haberleşme olayı belirli fonksiyonlarla haberleşerek öğrenme, algılama, tepki verme gibi refleksleri gösterebilir.

Gerçek dünya özellikleri doğrusal olmayan yapıdadırlar. Bu yapıları yapay sinir ağlarına öğretmek için aktivasyon fonksiyonlarına ihtiyaç duyulmaktadır. Şekil 3.22’de görüldüğü üzere basitçe herhangi bir yapay sinir ağında x giriş verileri, w ağırlıklar olarak tanımlanırsa ve ağın sonuç kısmına aktarılan değere $f(x)$ yani aktivasyon fonksiyonu uygulanmaktadır. Bu çıkış birleşik sistemlerde elde edilecek yeni bir işlem için başka bir algoritmanın giriş kısmı olabilmektedir.

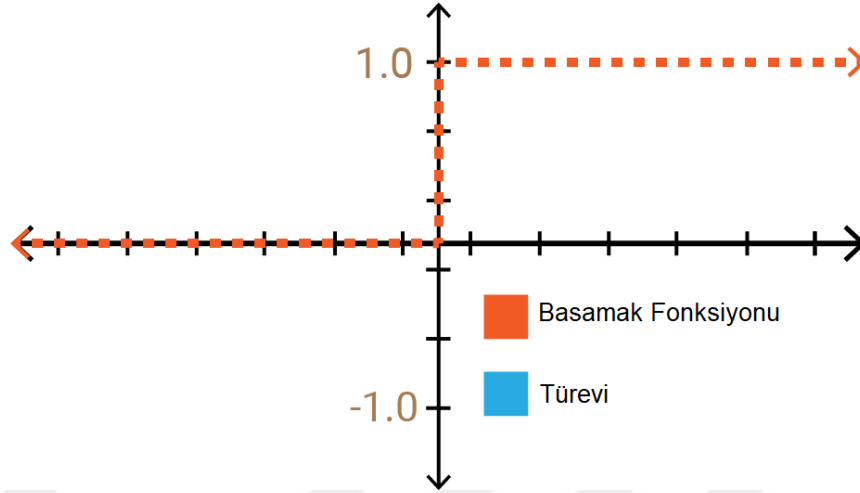


Şekil 3.22. Aktivasyon fonksiyonu (Adalı 2018)

$f(x)$ aktivasyon fonksiyonunun uygulanma sebebi, çıkıştaki sinyalin doğrusal fonksiyon olmasını engellemektir. Doğrusal fonksiyonlar tek dereceli olduğundan öğrenme sınırlı olmaktadır. Sistemlerin gerçek dünya verileri olan ses, görüntü, video veya yazı gibi karmaşık verileri öğrenmesi gerekmektedir. Bu durum da çok katmanlı derin sinir ağları karmaşık verilerden anlamlı özellikleri öğrenebilir.

Tez çalışmasında kullanılan SSD Mobilnet algoritmasının aktivasyon fonksiyonu olan sigmoid fonksiyonu ve bazı fonksiyonların avantaj ve dezavantajlarından aşağıda bahsedilmiştir (Kızrak 2019).

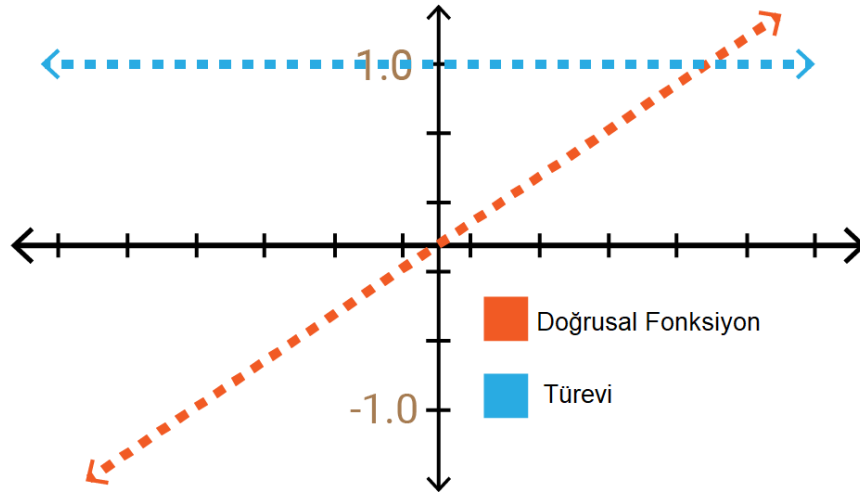
Basamak (step) fonksiyonu:



Şekil 3.23. Basamak fonksiyon grafiği (Kızrak 2019)

İki değeri olan bir fonksiyondur. Genelde çıkış katmanlarında kullanılır. Şekil 3.23’de görüldüğü üzere gizli katmanlarda türev değeri olmadığı için kullanılması çok kullanışlı değildir.

Doğrusal (Lineer) Fonksiyon:



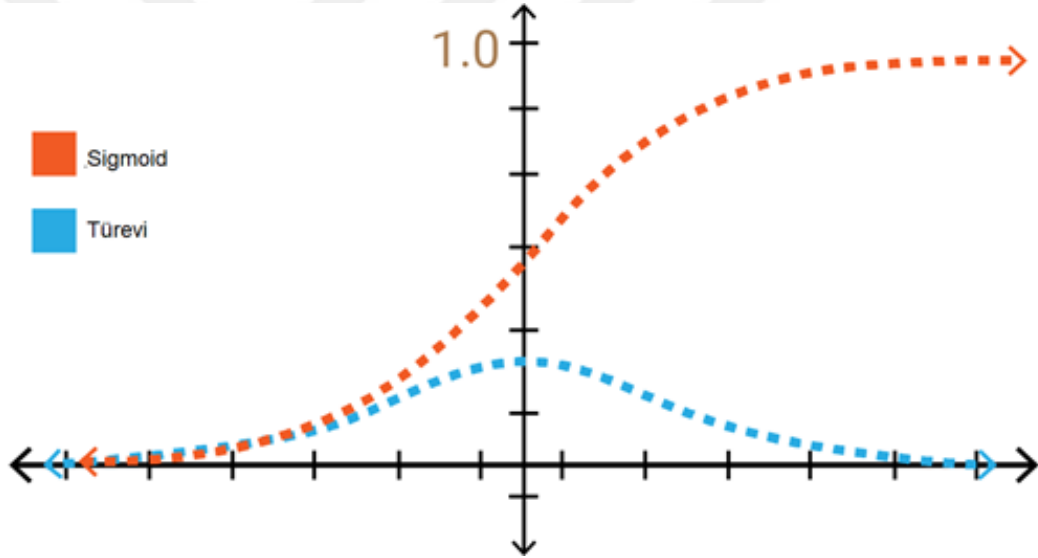
Şekil 3.24. Doğrusal fonksiyon grafiği (Kızrak 2019)

Kullanılma açısından basamak fonksiyonundan daha iyi olan doğrusal fonksiyon Şekil 3.24’de görüldüğü gibi iki değerden daha çok veriye sahiptir. Ancak önemli

sayılacak bir dezavantajı doğrusal fonksiyonun türevinin sabit olmasıdır. Değişkenin yok edilmesiyle sabit bir ifade elde edilmiş olmaktadır. Bu durumda öğrenme olayından söz edilememektedir. Başka bir dezavantajı ise tüm katmanlarda doğrusal fonksiyon kullanıldığında giriş ve çıkış katmanlarında hep aynı doğrusal fonksiyon elde edilmektedir. Bu kullanım çok katmanlı çalışmalarda sabit sonuç doğurmaktadır (Kızrak 2019).

Sigmoid fonksiyonu:

Doğada çoğu şey doğrusal olmadığından dolayı doğayı anlamak için de doğrusal olmayan fonksiyonlara ihtiyaç vardır. Bu fonksiyonlardan biri de sigmoid fonksiyonudur.

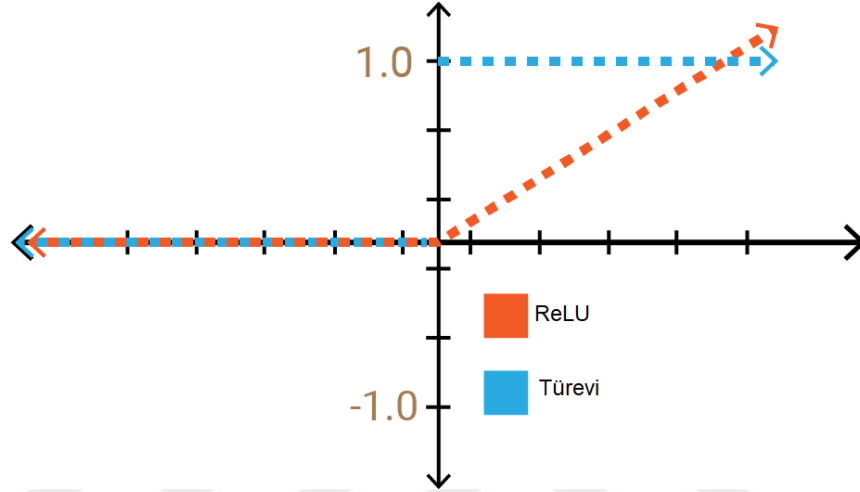


Şekil 3.25. Sigmoid fonksiyon grafiği (Kızrak 2019)

Türevlenebilir olması ve katmanlarda farklı fonksiyonlar kullandığından dolayı Şekil 3.25’de gösterilen sigmoid fonksiyonu derin öğrenme için ideal bir fonksiyondur. Sigmoid fonksiyonunda yapılacak küçük bir değişiklik fonksiyonun türevi olan y de büyük değişimlere sebep olmaktadır. Bu değişiklikler sayesinde fonksiyon ile türevi arasındaki açıklık iyi bir sınıflayıcı anlamı taşımaktadır. Sonuç kısmında ise her zaman 0 ve 1 değerlerinin görülmesi sistemin kararlı olduğunu gösterir. Fakat türev kısmında sonlara doğru 0 değerine yakın görülmekte buda yavaş öğrenme demektir. Ayrıca karmaşık öğrenmeler gereken işlemlerde yine sonlara doğru x deki değişime çok az cevap

vermeye başlayan bu durum çok değişkenli fonksiyonların kısmi türevlerini saklayan gradyan kaybına sebep olmaktadır. Bu olay bilgi kaybı demektir. Bu kayıplar da sigmoid fonksiyonunun dezavantajlarındandır (Kızrak 2019).

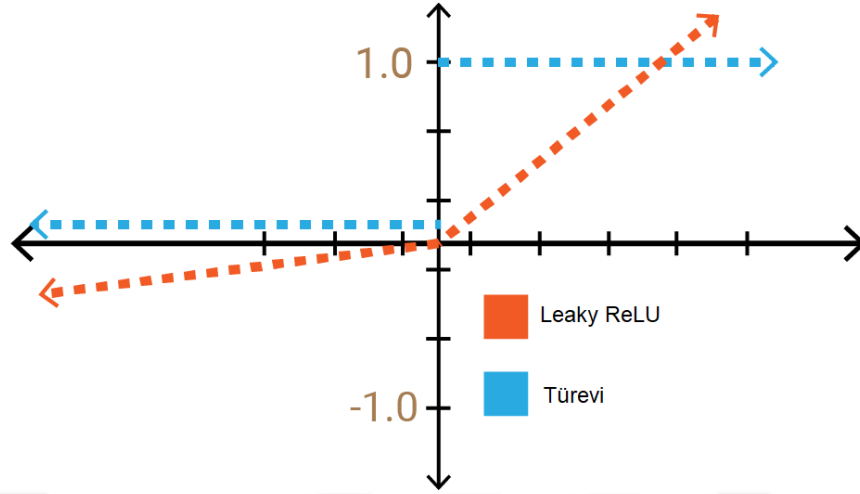
ReLU (Rectified Linear Unit) Fonksiyonu: Sigmoid fonksiyonlarının dezavantajlarını ortadan kaldırmak için geliştirilmiştir.



Şekil 3.26. ReLu fonksiyon grafiği (Kızrak 2019)

Sigmoid fonksiyonlarına alternatif olarak ReLu fonksiyonu Şekil 3.26’da incelenecek olursa; Her ne kadar doğrusal görünse de aslında iyi bir tahmin edici fonksiyondur. ReLu fonksiyonların yakınsamaları ile herhangi bir fonksiyon tahmin edilmesi kolaydır. Görüldüğü gibi negatif tarafta 0 değerinin olması ağıın daha hızlı çalışacağını göstermektedir. Hesap yükünü azaltması diğer fonksiyonlara göre kullanım isteğini artırmış olsa da 0 değer kısmında türevinin olmaması da öğrenme olmadığı anlamına gelmektedir. Buda başka alternatifler oluşturmaya zorlamıştır (Kızrak 2019).

Sızıntı (Leaky) ReLU Fonksiyonu:



Şekil 3.27. Sızıntı fonksiyon grafiği (Kızrak 2019)

Normal ReLU fonksiyonuna nazaran negatif düzlemde iyileştirme yaparak 0 durumunda ki öğrenme olayını durdurmamak için sifıra çok yakın 0,01 değeri ile oluşturulan bu fonksiyon Şekil 3.27’de gösterilmiştir. Sıfır durumuna çok yakın olma durumuna sızıntı denir. Sızdırılan bu ReLU değeri, tanım aralığı -sonsuzda devam etmekte 0 a yakın ama 0 olmamaktadır. Normal ReLU’daki 0 değerinde ölen gradyanların engellenmesi hem negatif hem de pozitif kısımda öğrenme olduğu anlamına gelmektedir. Bu durumda karmaşık yapıların görüntü işleme ihtiyacına karşılık vermektedir (Kızrak 2019).

Aktivasyon fonksiyonlarında türev alınmasının sebebi; Veri girdiler ile ilgili hata değerlerini tahmin ve hesaplama adına yapay sinir ağında hatanın geriye yayılım algoritması uygulanmaktadır. Optimizasyonu belirlemek için hata oranını minimuma indirmek gerekmektedir. Uygun optimizasyon algoritmasını seçmek ise hedefe göre özelleşmektedir.

Geriye yayılım (backpropagation) algoritması ile öğrenme işlemini nöronlar sayesinde yapmaktadır. Bu algoritma türev alan bir sistemden oluşur. Birinci dereceden x ’e bağlı bir fonksiyonun, x ’e göre türevi alındığında c gibi bir sabit sonuca erişilir. Bu durumda fonksiyonun x ile bir ilişkinin kalmadığını gösterir. Türevin sabit çıkması sistemde öğrenme olmadığı anlamına geldiğinden dolayı aktivasyon fonksiyonlarında

doğrusal fonksiyonlar kullanmak hata olabilir. Bu durumda sigmoid fonksiyonu ve ReLU fonksiyonları gibi doğrusal olmayan türevi sabit çıkmayan fonksiyonlar sayesinde geri yayılım yapıp nöronlar sayesinde öğrenme gerçekleştirilir.

Öğrenme olayında oldukça fazla fonksiyonlar geliştirilmektedir. Bunun sebebi kullanıldığı alanlar arttıkça doğan ihtiyaçlardır.

Sistemin yöntemi ssd mobilnet arka planda ReLu fonksiyonu ile çalışmakta ve verimi yüksek olması amaçlanmıştır. Aşağıda bazı fonksiyonların test sonuçları Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Fonksiyonların test sonuçları (Kızrak 2019)

AKTİVASYON FONKSİYON	TEST BAŞARIMI	TEST YİTİMİ
Sigmoid Fonksiyonu	0,9866	0,0567
Hiberbolik Tanjant Fonksiyonu	0,9905	0,0474
ReLU Fonksiyonu	0,9929	0,0401
Leaky (Sızıntı) ReLU	0,9925	0,0403
Swish Fonksiyonu	0,9922	0,0412

3.5 Donanımsal Yapılar

Donanımsal olarak çalışmanın ana teması görüntü alma, aldığı görüntüleri daha önce sınıflandırma yapılan nesnelerden karşılaştırma yapmaktır. Karşılaştırmanın ardından sistemde kullanılan algoritmalar ile nesnenin ne olduğuna karar verilmesidir. Görüntü toplamak için raspberry pi4 ile uyumlu versiyon 2 kamerası, bunu 360 derecede belli açıda sürekli yapabilmesi için step motor ve bu motoru sürebilmek için birde motor sürücüsü kullanılmıştır. Motor sürücüsünü kontrol edip istenilen hıza ayarlamak için arduino kullanılmıştır. Sistemi kapatmak ve tasarımını radara benzetmek adına 3d yazıcı kullanılmıştır.

3.5.1 Step motor

Şekil 3.28’de görülen step motor, girişe uygulanan elektrik enerjisini fiziksel enerjiye çevirerek dönme hareketi ile sonuçlandıran elektromekanik aygıtlardır. Bu dönme hareketi isimlerinden anlaşıldığı gibi adım adım hareket etmektedir. Temel olarak üç çeşit step motor bulunmaktadır. Bunlar sabit mıknatıslı step motorlar, değişken relüktanslı step motorlar ve hybrid step motorlardır (Zeren 2020). Step motorlar, dijital ortam işaretlerine uyumlu olması, kontrol edilebilmesi, düşük hassasiyetlerde bile yüksek performanslı olarak çalışabilmesi, geniş hız aralığında çalışabilmesi gibi avantajlarından dolayı tez çalışmasında kullanıldı. Bu avantajlara ek olarak 360° derecelik görüş alanı ve kameranın hızından dolayı sabit ve kontrol edilebilir olması ayrıca bir avantaj sağladığı için çalışmada etkin bir şekilde kullanıldı.

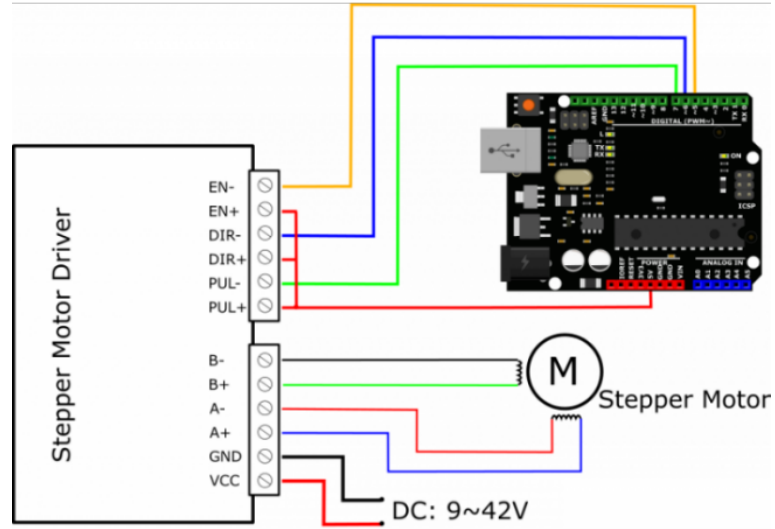


Adım açısı: 1.8°
Çalışma gerilimi: 3.3V DC
Çalışma akımı: 1.5A
Tutunma torku: 400mN.m
Faz direnci: 2.2 $\Omega \pm 10\%$ (20°)
Endüktans: 3.8mH $\pm 20\%$
Yalıtım direnci: 100M Ω Min 500VDC

Şekil 3.28. Step motor

3.5.2 Step motor sürücü TB6600

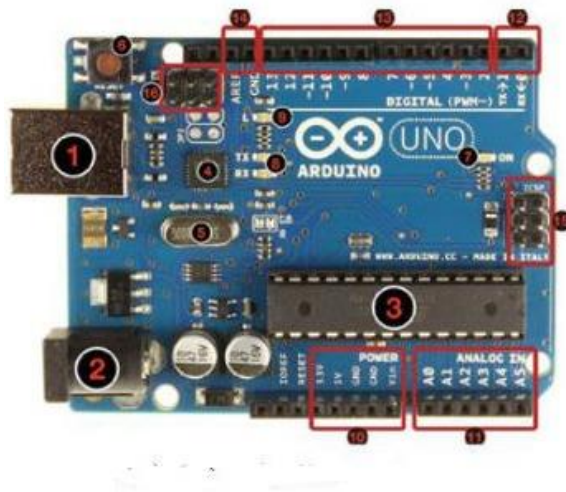
Motor dönüş hızı hassasiyet gibi ayarlamaları için kontrolcü görevini motor sürücüler yüklenir. Şekil 3.29’da görülen TB660 step motor sürücüsü iki fazlı kolay kullanılabilen bir sürücüdür. Mikro denetleyici ile dijital sinyal gönderebilen cihazlar ile kontrol edilebilmektedir (Zeren 2020).



Şekil 3.29. TB6600 Step Motor Sürücüsü

3.5.3 Arduino uno

Şekil 3.30’da gösterilen mikro işlemci, elektronik uygulamalarında kolayca ulaşılabilen açık kaynaklı bir cihazdır. Motor sürücüsünü doğru çalıştırmak için, ara işlemci görevinde olmak üzere arduino motor sürme, sensör veya kumanda devrelerinde kullanılabilir.



- 1 : USB jakı
- 2: Power jakı (7-12 V DC)
- 3: Mikrodenetleyici ATmega328
- 4: Haberleşme çipi
- 5: 16 MHz kristal
- 6: Reset butonu
- 7: Power ledleri
- 8: TX / RX ledleri
- 9: Ledler
- 10: Power pinleri
- 11: Analog girişler
- 12: TX / RX pinleri
- 13: Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)
- 14: Grounf ve AREF pinleri
- 15: ATmega328 için ICSP
- 16: USB arayüzü için ICSP

Şekil 3.30. Arduino Uno ve teknik özellikleri

3.5.4 Raspberry pi4

İngiltere de üretilmiş ve geliştirilmiş raspberry, mini bilgisayar olarak da bilinen bu cihaz gerçekte de bilgisayardan farkı yok denecek kadar azdır. Şekil 3.31’de gösterdiği gibi monitör, mouse, klavye gibi bağlantı yerleri bulunan raspberry, bir bilgisayarla yapılabilecek işleri daha az alanda gerçekleştirebilir. Taşıma kolaylığı ve işlem yeteneğinden dolayı bu tezin ana ekipmanları arasında yerini almıştır.

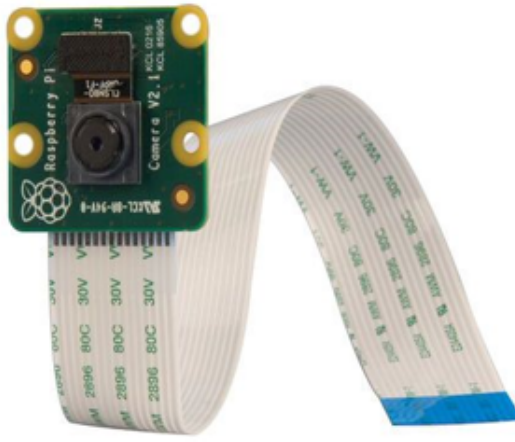


1.5GHz <u>quad-core</u> 64-bit ARM Cortex-A72 işlemci
1 GB, 2GB, 4GB LPDDR4 SDRAM seçenekli
<u>Gigabit ethernet girişi.</u>
Bluetooth 5.0.
2 adet USB 3.0 ve 2 adet USB 2.0 port
<u>Çift monitör desteği.</u>
<u>VideoCore VI grafikleri.</u>
40 <u>pinli</u> GPIO bağlantı noktaları

Şekil 3.31. Raspbery ve teknik özellikleri

3.5.5 Pi kamera V2

Görüntü işleme olayının en önemli elemanlarından biri de kameradır. Sabit odaklı bir objektife sahip, Raspberry Pi için özel olarak tasarlanmış V2 kamerası Şekil 3.32’de gösterildiği gibi yüksek kaliteli 8 megapiksel çözünürlüklü Sony IMX219 görüntü sensörüne sahiptir. 3280 x 2464 piksellik statik görüntüler üretebilir ve ayrıca 1080p30, 720p60 ve 640x480p90 videoları desteklemektedir. Kart üst yüzeyindeki küçük prizlerden biriyle Pi’ye bağlanmakta ve özellikle kameralara bağlantısı için tasarlanmış olan CSI arabirimini kullanmaktadır.



Sabit odaklı lens
8 megapiksel doğal çözünürlüklü sensör - 3280 x 2464 piksel fotoğraf çekebilir
1080p30, 720p60 ve 640x480p90 video destekliyor
Boyut: 5mm x 23mm x 9mm
Ağırlık: Sadece 3gram
Raspberry Pi işletim sistemi Raspbian'ın son versiyonu ile uyumlu olarak çalışmaktadır
Kısa ribbon kablo ile Raspberry Pi'ye doğrudan bağlanabilir

Şekil 3.32. Raspberry Pi kamera ve teknik özellikleri

3.5.6 Motor için adaptör

Şekil 3.33'de gösterildiği gibi küçük voltajlarda çalışan cihazların ihtiyaçlarını karşılayacak voltajlara göre şebeke gerilimini ayarlayan ara elemandır. Türkiye de 220 V olan şebeke değeri step motora uygun hale getirmek için adaptör diğer adı ile transformatör kullanıldı.



Şekil 3.33. Adaptör

3.5.7 Kapasitif dokunmatik 7 inç ekran

Dokunmatik ekran klavye, fare yahut monitör görevini tek başına yerine getirebilen cihazdır. Raspberry Pi ile uyumludur. Ancak Raspberry Pi ile direk bağlantı

3. MATERYAL ve YÖNTEM

sağlanamaz. Bağlantının sağlanabilmesi için HDMI/DVI 40-Pin TTL dönüştürücü gerekmektedir. Şekil 3.34’de gösterilen ekran 1024x600 piksel çözünürlüğe sahiptir 3V - 3.6V aralığında gerilim ile çalışır. Ekran boyutu 154.21 (G) × 85.92 (H) şeklindedir.



Şekil 3.34. Dokunmatik ekran

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Google alt yapı programlarının kamera ve Raspery Pi 4 elemanlarıyla doğru ve uyumlu çalışması için SSD görüntü işleme tekniği kullanıldı. Bu teknik Tesla gibi büyük firmaların insansız araçlarda kullandığı bir teknik olması nedeniyle güvenilirlik açısından önemlidir.

Raspberry Pi 4 google alt yapısı ile uyumlu çalışan ve oldukça kompleks yapılar için tasarlanan bir mini bilgisayar çalışmanın ana unsurlarındandır. Şekil 4.1’de gösterilmiştir. Raspberry Pi 4 sistemi görüntülerin işlenmesi, karar mekanizması, ekranda bildirmesi, hassasiyet ayarları gibi ana algoritmanın işlendiği sistemleri içerisinde bulundurmaktadır.

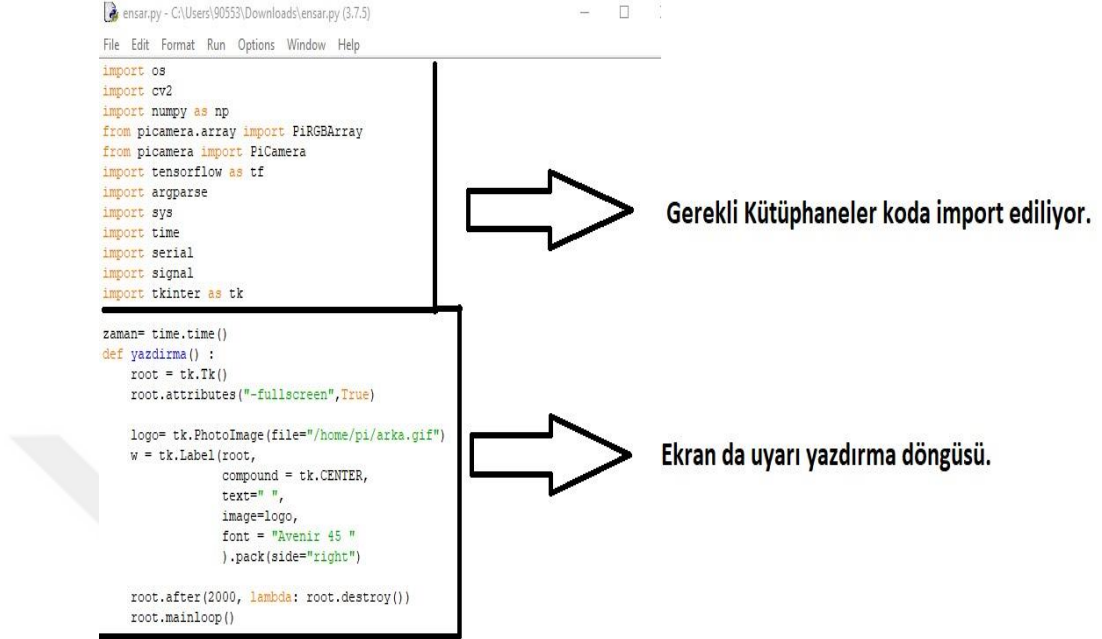


Şekil 4.1. Raspberry Pi sistemi

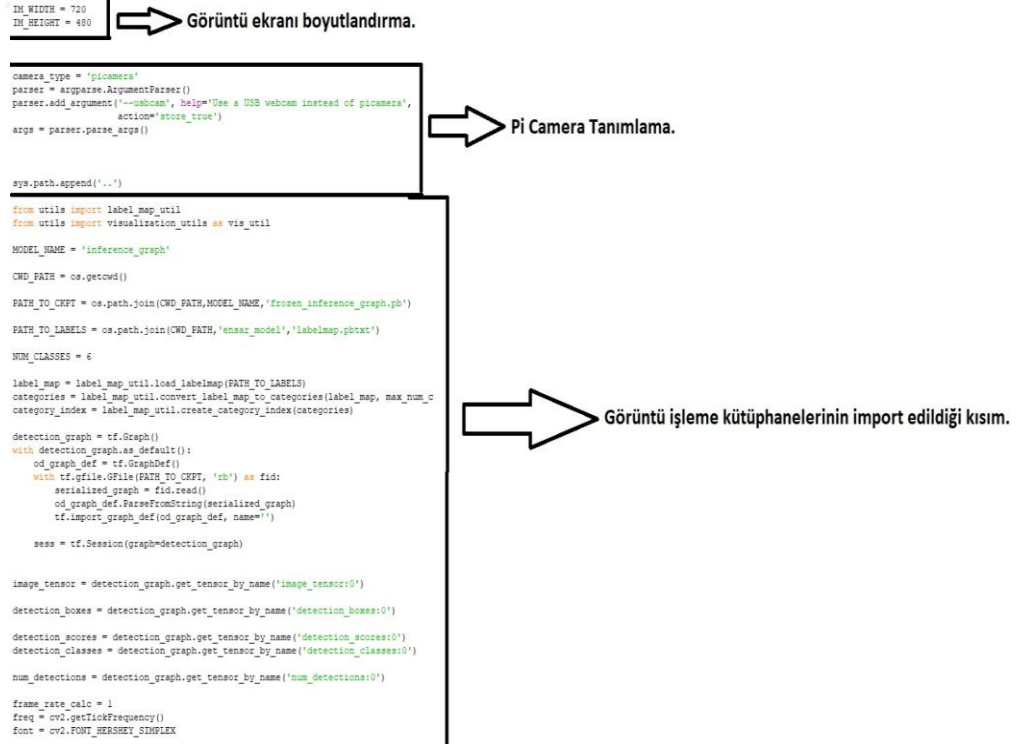
Raspberry kullanımında önce bağlanacak olan elemanların kütüphaneleri yüklenmelidir. Bu yükleme işlemi raspberry’nin her elamanla uyumlu ve kendi dilinden yani makine dilinden veri akışı sağlaması için yapılmaktadır. Örneğin raspberry ile kamera arasında gerçekleşen data alış-verişini, raspberry sisteminin kendi içerisinde anlamlandırabilmesi için Şekil 4.2, Şekil 4.3 ve Şekil 4.4’te gösterilen kamera kodu kütüphaneleri eklendi. Aynı nedenlerden ötürü diğer kod ve elemanlar eklenmiş ve algoritma oluşturuldu. Ayrıca ekranda gösterilen uyarı logosu, boyutu bekleme süresi ve

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

şekli kodlarla ayarlanabilmektedir. Şekil 4.5'te kurulan sistemle uyarı logosu ekran görüntüsü gösterildi.



Şekil 4.2. Kütüphane eklenmesi ve ekranda uyarı yazdırma döngüsü



Şekil 4.3. Raspberry içerisinde bazı algoritma kodları

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

```
iha_sayici = 0
pause = 0
pause_counter = 0

def iha_radar(frame):

    zaman= time.time()
    global pause, pause_counter, iha_sayici, iha

    frame_expanded = np.expand_dims(frame, axis=0)

    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={image_tensor: frame_expanded})

    vis_util.visualize_boxes_and_labels_on_image_array(
        frame,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=12,
        min_score_thresh=0.85)

    if ((int(classes[0][0]) == 1) and (scores[0][0]) >= 0.55) and (pause == 0):
        a = int(((boxes[0][0][1]+boxes[0][0][3])/2)*IM_WIDTH)
        b = int(((boxes[0][0][0]+boxes[0][0][2])/2)*IM_HEIGHT)
        cv2.circle(frame, (a,b), 5, (255,255,255), -1)
        iha_sayici = iha_sayici + 1

    if iha_sayici >= 2:
        camera.capture('/home/pi/Desktop/ihra/' + str(zaman) + '.jpg')
        yazdirma()

        detected_kagit = True
        iha_sayici = 0
        pause = 1

    if pause == 1:
        pause_counter = pause_counter + 1

        if pause_counter > 1:
            pause = 0
            pause_counter = 0

    return frame
```

Yüzdeye göre etiket yapma kodu ve üstteki kodlarla etiket kalınlığı rengi gibi ayarlar.

1. Kısımda benzetilen id ile karşılaştırma yapılıyor. 2. kısımda skora göre algılaması sağlanmaktadır.

⇒ Görüntünün algılandığı kısım

⇒ 2 kez kontrol yaptırıp fotoğrafını çektilir kaydettirdiğimiz algoritma.

Şekil 4.4. Raspberry içerisinde işlem algoritma kodları

Sistemin çalışır halde olması ve dokunmatik ekranda görülen uyarı yazısı Şekil 4.5'te verilmiştir.



Şekil 4.5. Ekranda uyarı döngüsü sonucu görülen logo ve boyutu

Bu çalışmada gerçek zamanlı veri akışı sağlamak önemli yere sahiptir. Bu akışı doğru yapabilmek için eklenen kodlar Şekil 4.6’da verilmiştir.

```
if camera_type == "picamera":
    camera = PiCamera()
    camera.resolution = (IM_WIDTH, IM_HEIGHT)
    camera.framerate = 30
    rawCapture = PiRGBArray(camera, size=(IM_WIDTH, IM_HEIGHT))
    rawCapture.truncate(0)

    for frame1 in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        t1 = cv2.getTickCount()

        frame = np.copy(frame1.array)
        frame.setflags(write=1)
        frame_expanded = np.expand_dims(frame, axis=0)
        frame = lha_radar(frame)

        cv2.putText(frame, "FPS: {0:.2f}".format(frame_rate_calc), (30, 50), font, 1, (255, 255, 0), 2, cv2.LINE_AA)

        cv2.imshow('İHA RADAR KUSAR FOSATEPE', frame)

        t2 = cv2.getTickCount()
        time1 = (t2-t1)/freq
        frame_rate_calc = 1/time1

        if cv2.waitKey(1) == ord('q'):
            break

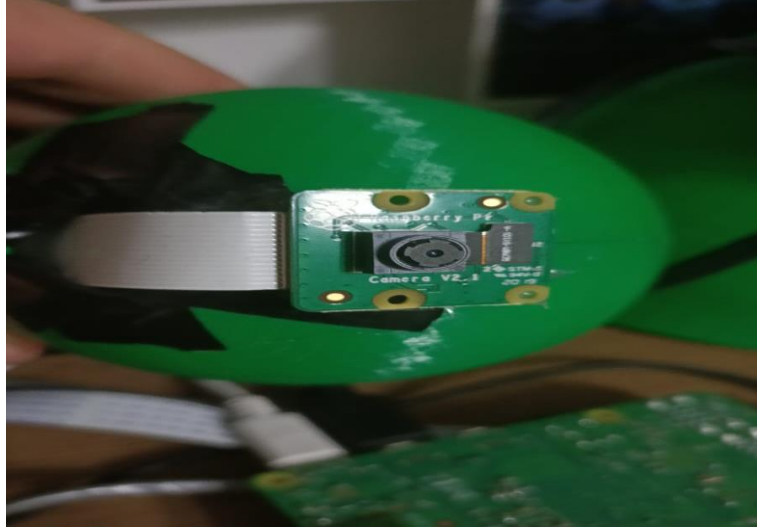
        rawCapture.truncate(0)

    camera.close()
```

➡ Görüntünün canlı akışını sağlayan ve pencereyi açan kod kısmı.

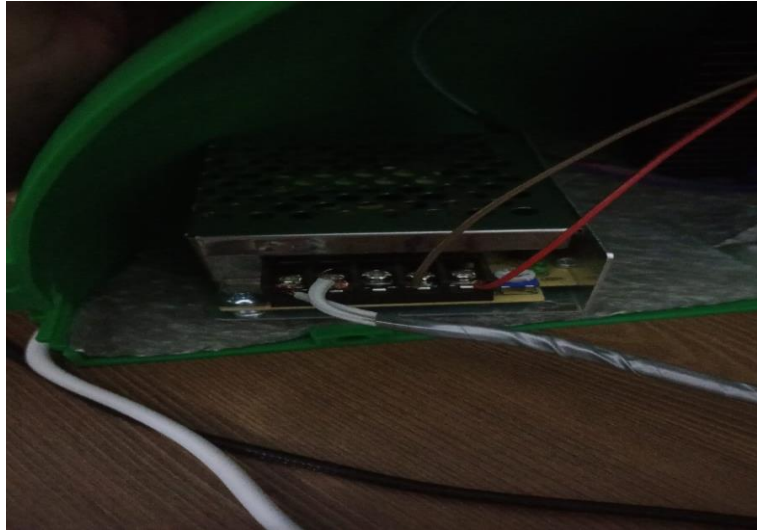
Şekil 4.6. Gerçek zaman akışı sağlayan kodlar

Raspberry ile uyumlu fps hızı 2.79 olan v2 kamera kullanıldı. Pi kamera da denilen bu kamera Şekil 4.7’de gösterilmiş olup, hem canlı akış hem de görüntüleme kısmında giriş verisinin kalitesi açısından önemli bir yere sahiptir.



Şekil 4.7. Pi kamera v2

Şekil 4.8’de gösterilen adaptör, şebeke gerilimi olan 220V’da çalışmayan cihazlar için kullanılmaktadır. Adaptör şebeke gerilimini motor çalışma gerilimine göre ayarlayan bir dönüştürücüdür. Bataryası olan şarj edilebilir cihazların çoğunda adaptör kullanılmaktadır. Bu çalışma da her ne kadar batarya kullanılsa da şebeke gerilimini anlık olarak step motorun çalışma gerilimine dönüştürmek için adaptör kullanılmıştır.



Şekil 4.8. Adaptör (dönüştürücü)

Motor sürücüsü, motorların devrini, frekansını değiştirerek özelleştiren elektriksel ekipmanlardır. Devir kontrolcüsü olması yanında koruma ve arduino programı devir kodları ile kontrol özellikleri eklendi. Böylece tek bir yöne doğru değil tam tersi yöne de

dönmektedir. Motor miline bağlı olan 3d yazıcıda basılmış kamera yatağı üzerindeki v2 kamera görüntüyü kaliteli bir şekilde algoritma içerisine alabilmesi için en uygun hız ile kodlandı. Şekil 4.9’da gösterilen motor sürücüleri bağlı oldukları kod çözücü veya mikro kontrolcülerden kare dalga almaktadırlar. Gelen sinyaller işlenerek motorun adım (dönüş) atması sağlanır.



Şekil 4.9. Motor sürücü tb6600 ve arduino ile bağlantısı

Arduino çeşitli sensörlerden veri bilgisi alabilen elektronik cihazdır. Şekil 4.10’da arduino ile kurulmuş sistem görülmektedir. Sistem içi bilgiler çeşitli şekil ve yöntemler kullanarak özelleştirebilir. Motor gibi cihazların uyarım sinyallerini ayarlayarak sürücülerinin kontrolünü sağlar. Motor sürücüsü ile beraber çalışıp gönderdiği sinyallerle nasıl çalışacağını karar mekanizması arduino ile yapılır. Motor hızı istenilen devirde ayarlanarak kamera görüntü kalitesi ayarlanmasını sağlar.



Şekil 4.10. Arduino Uno

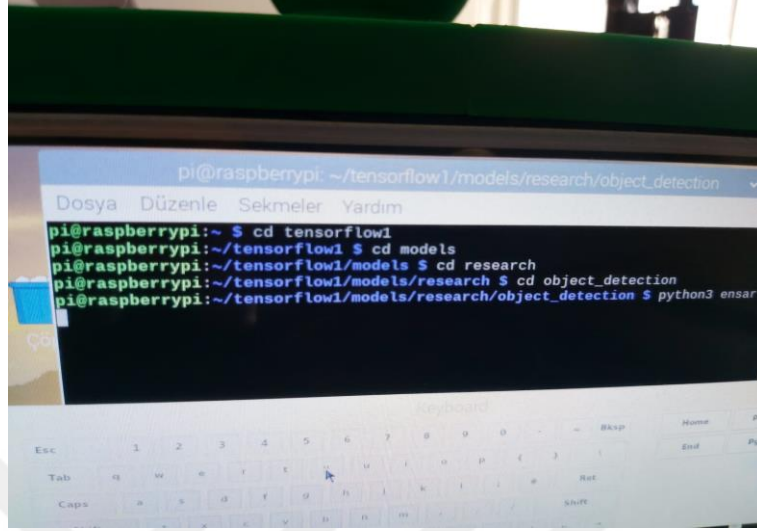
Step (adım) motor; elektronik anahtarlar sayesinde sargılara enerji verilir ve rotor, üzerinde akım olan sargının karşısına gelerek durur. Milin hızı sargılara sırasıyla gönderilen sinyallerle belirlenir. Bu hız step motorda değişkendir ve verilen enerji ile doğru orantılıdır. Şekil 4.11’de gösterildiği gibi 360° derece dönme özelliği ile tüm çevreyi eş zamanlı olarak tarama yapmaktadır.



Şekil 4.11. Nema17 step(adım) motora sahip kamera

Şekil 4.12’de gösterildiği gibi, 7 inç kapasitif dokunmatik ekran; Kapasitif ekran dokunmatik olduğu için klavye, mouse gibi ek elemanların görevini de üstlenmektedir. Ergonomi açısından avantaj sağlayan dokunmatik ekran canlı akışın sürekli görüldüğü,

nesne tanındıktan sonra dikkat logosunun çıktığı ve nesne fotolarının kaydedilip tekrar gözlemlendiği donanımdır.



Şekil 4.12. Kapasitif dokunmatik ekran ve algortima başlatma kodları

Araştırma çalışma ya da projelerin ayrılmaz parçası verimdir. Çalışmanın hayata geçirilmesi maliyeti veya ergonomik olup olmaması adına verim hesabı yapılmalıdır. Bu çalışma da verim İHA tespiti üzerine yapıldı. 100 kez havalandırılan İHA kamera görüş açısına mesafe değişikliği yapılarak görüş alanına girdi. Bu denemelerin Çizelge 4.1’de görüldüğü gibi nesnenin 94 kez doğru pozitif, 4 kez yanlış pozitif, 1 kez yanlış negatif ve 1 kez de doğru negatif tespiti gerçekleştirmiştir. Kapalı alanda gerçekleştirilen bu denemelerin ortam şartlarından etkilendiği unutulmamalıdır. Eşitlik 1’de matematiksel olarak doğruluk hesabı yapılmıştır. Açık alanda ise kapalı ortam koşullarından arındırıldığı için % 99 başarı elde edildi.

$$\text{Doğruluk} = \frac{\text{Doğru Algılama Sayısı}}{\text{Toplam Algılama Sayısı}} \times 100 \quad (1)$$

Çizelge 4.1. Test sonucu elde edilen veriler

Nesne Adı	Doğru Pozitif Oranı	Yanlış Pozitif Oranı	Doğru Negatif Oranı	Yanlış Negatif Oranı	Doğruluk
İHA	94	4	1	1	% 94

Doğruluk Oranı: Veri setinde her bir örneğin sınıf etiketlerinin sınıflandırılmasında algoritmaların performansının değerlendirilmesinde doğruluğu ile hesaplanır.

Doğru pozitif oranı: Sınıflayıcıya göre pozitif olarak etiketlenenlerin ne kadarının gerçekte pozitif olduğunu gösterir.

Yanlış pozitif oranı: Sınıflayıcıya göre pozitif olarak etiketlenenlerin ne kadarının gerçekte negatif olduğunu gösterir.

Doğru negatif oranı: Sınıflayıcıya göre negatif olarak etiketlenenlerin ne kadarının gerçekte negatif olduğunu gösterir.

Yanlış negatif oranı: Sınıflayıcıya göre negatif olarak etiketlenenlerin ne kadarının gerçekte pozitif olduğunu gösterir.

Doğru pozitif oranı ve yanlış pozitif oranının toplamı 1'e eşittir. Ayrıca doğru negatif oranı ile yanlış negatif oranının toplamı da 1'e eşittir.

Doğru tahmin oranı: Pozitif olarak etiketlenen sınıfların ne kadarının gerçekte pozitif olduğunu göstermektedir. Değerlenen kriterler Çizelge 4.2'de verilmiştir.

Görüntü işleme teknolojisi elektronik ve haberleşme alanlarının ortaklaşa çalışması ile ortaya çıkmış bir teknolojidir. Bu çalışma da elektronik kısımlar önemli rol oynamakla beraber birlikte uyumlu çalışması da önemlidir. Motor dönüş hızı kameranın verimliliğini ayarlarken kamera da görüntü işleme modeli olan SSD Mobilnet gömülü Raspberry Pi 4 e iyi bir giriş verisi sağlamaktadır.

Çizelge 4.2. Derlenmiş değerlendirme kriterleri tablosu (Zeren 2020)

KRİTERLER	KISALTMALAR	FORMÜLLER
DOĞRULUK ORANI (ACCURACY)	ACC	$(TP+TN)/(N+P)$
DUYARLILIK (DOĞRU POZİTİF ORANI) (SENSITIVITY)	TPR	$TP/(TP+FN)$
ÖZGÜLLÜK (DOĞRU NEGATİF ORANI) (SPECIFITY)	TNR	$TN/(TN+FP)$
YANLIŞ POZİTİF ORANI (FALSE POSITIVE RATE)	FPR	$FP/(FP+TN)$
YANLIŞ NEGATİF ORANI (FALSE NEGATIVE RATE)	FNR	$FN/(FN+TP)$
DOĞRU TAHMİN ORANI (POSITIVE PRED VALUE - PRECISION)	PPV	$PPVTP/(TP+FP)$
F PUANI (F SCORE)	F	$2*((PPV*TPR)/(PPV+TPR))$
HATA ORANI (ERROR RATE)	ERR	$(FP+FN)/(N+P)$

Burada;

TP: Pozitif Örneğin Doğru Sınıflandırılması İşlemi olarak tanımlanır.

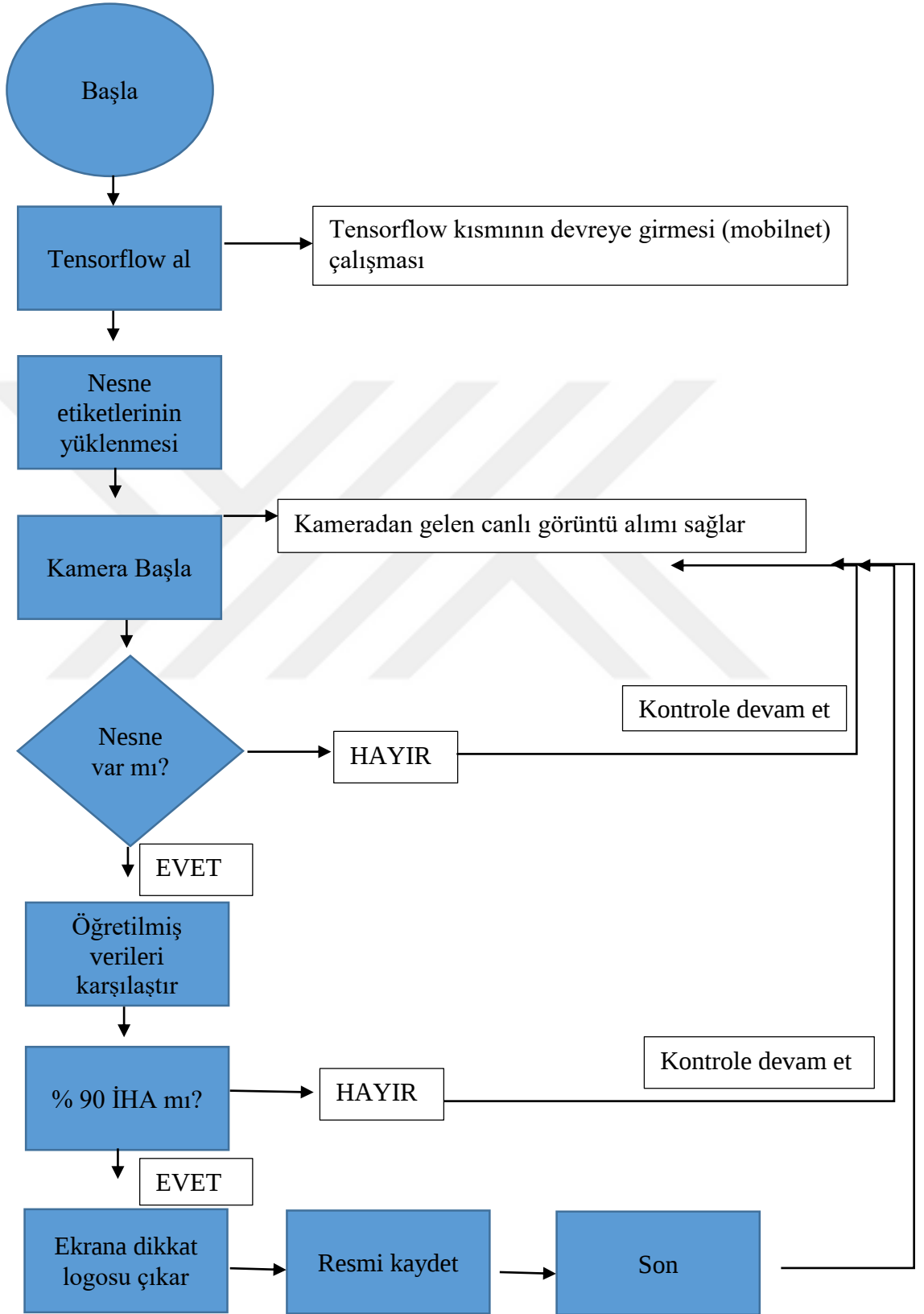
TN: Negatif Örneğin Doğru Sınıflandırılması İşlemi olarak tanımlanır.

FP: Negatif Örneğin Yanlış Sınıflandırılması İşlemi olarak tanımlanır.

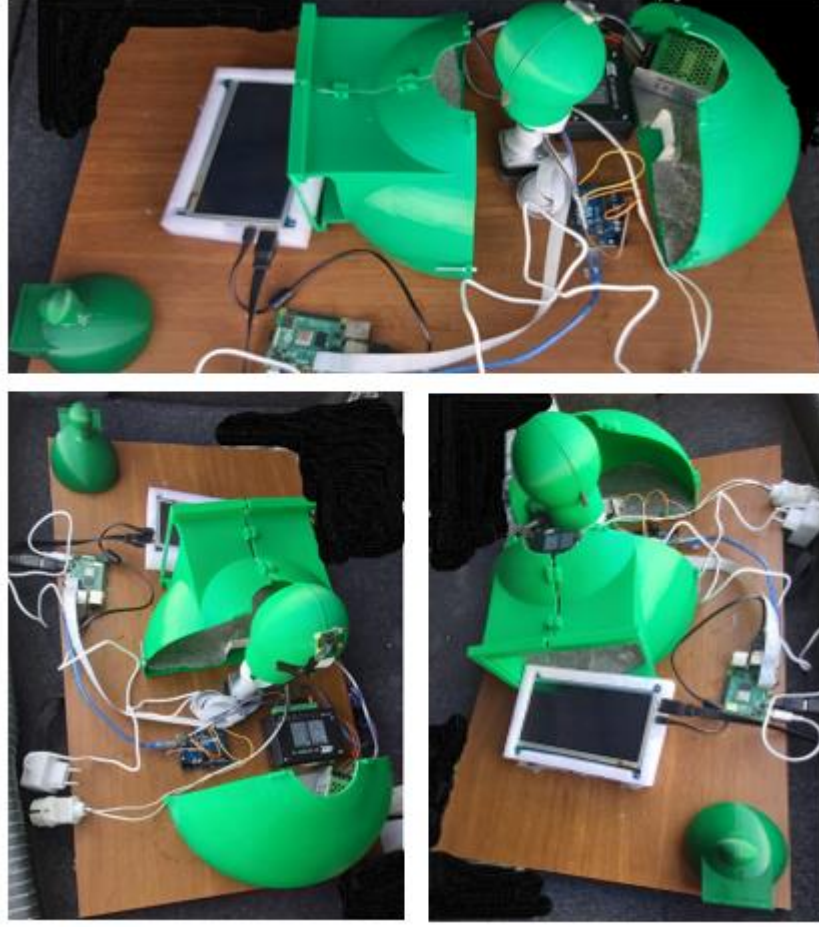
FN: Pozitif Örneğin Yanlış Sınıflandırılması İşlemi olarak tanımlanır.

Şekil 4.13'te görüldüğü üzere sistemin genel olarak algoritması kameradan gelen canlı görüntüyü alması ile başlar. Bunu yaklaşık olarak 1 saniye de bir gerçekleştirir. Nesne tanıma olayında önceden öğrenilmiş nesne görüntülerinin sisteme girilmesi gerekir. Bu durumda nesne görüntülerinin tensör ve sayısal haritası çıkarılarak etiketlenir. Etiketler sisteme yüklenir ve canlı görüntüler bu etiketli görüntülerle kıyaslanır. Aranılan nesne, kameradan gelen görüntü algoritması içerisinde sürekli kontrol edilir. Bir nesne ile karşılaşılması durumunda hemen kıyaslama başlar ve %90 benzerlik gösterirse kayıt

altına alıp ekranda dikkat logosu görülür. Aynı nesneyi birden fazla konumda kayıt altına alıp görüş açısından çıkana kadar dikkat logosunu göstermeye devam eder.



Şekil 4.13. Algoritma akış şeması



Şekil 4.14. İHA tespit cihaz görüntüleri (tüm sistem görüntüsü)

Gerekli yazılım sistemlerinin kontrolü ve sistem çalışmasını sağlayan raspberry pi 4 sistemi, V2 kamerası ve dokunmatik ekran gibi donanımsal parçalarla bütün hale getirilen sistem Şekil 4.14’te gösterilmiştir.

Kapalı alanda sistemin çalışması gerçekleştirildi. Şekil 4.15 (a)’da hareket halinde olan İHA ekranda görülmektedir. Arka planda gerçekleştirilen işlemler sonrası algoritma akış şemasında gösterildiği gibi, bu cismin İHA olduğuna sistem tarafından karar verilmiştir. İHA olduğuna karar verildikten sonra Şekil 4.15 (b)’de ekranda dikkat uyarısı belirlemektedir. Sonuç olarak sistem başarılı bir şekilde çalıştırıldı.



(a)

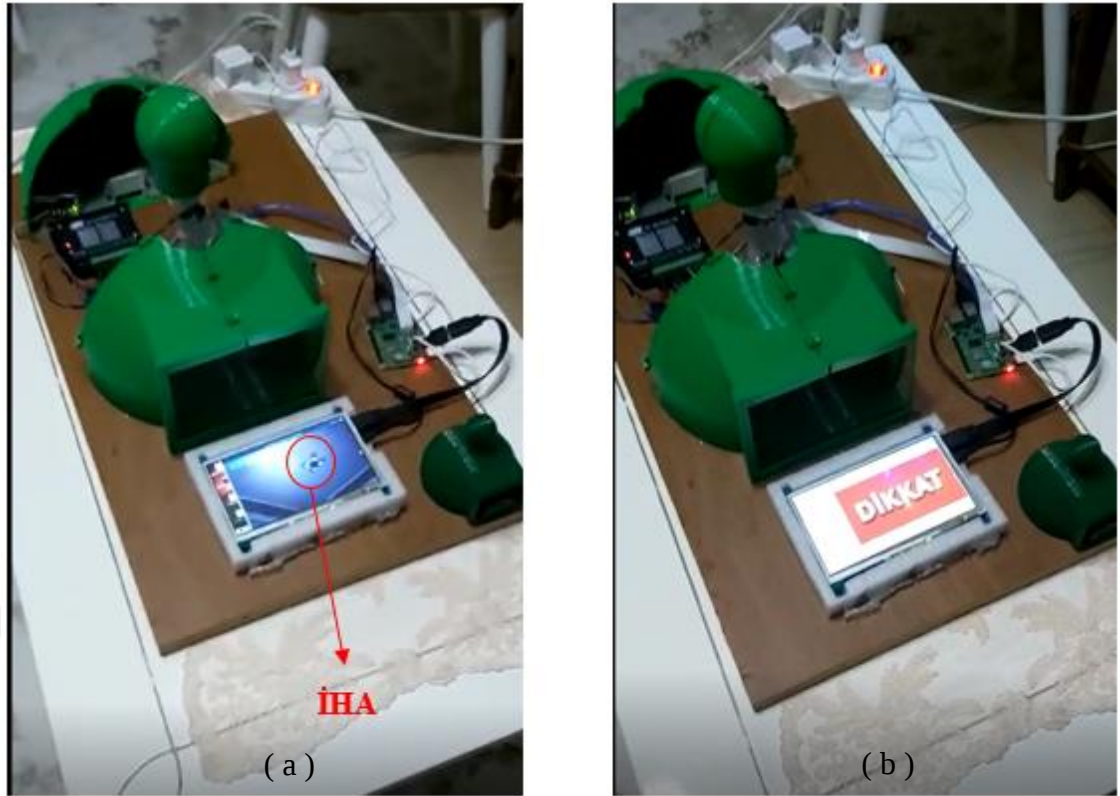


(b)



(c)

Şekil 4.15. (a) açık hava hareketsiz İHA tanıma deneme görüntüsü, (b) açık hava hareket halinde İHA tanıma deneme görüntüsü, (c) düşük ışık ortamında kapalı alanda İHA tanıma deneme görüntüsü



Şekil 4.16. (a) hareket halindeki İHA tespiti (b) ekranda tespit sonrası verilen uyarı

Bu çalışmada, uygun sistem donanımları ve yazılımları ile hazırlanmış parçaların birleştirilmesi ile İHA tespiti Şekil 4.15'te ve Şekil 4.16'da görüldüğü gibi açık havada, kapalı ortamda ve düşük ışık ortamında başarılı bir şekilde gerçekleştirilmiştir. Kurulan bu sistem sadece tespit değil aynı zamanda müdahale donanımları eklenerek yapılacak çalışmalarla geliştirilebilir.

5. SONUÇ ve ÖNERİLER

İHA araçları, savunma sanayi, sağlık ve güvenlik hizmetlerinin yanı sıra tarım ve çeşitli etkinliklerde kullanım alanlarına sahiptir. Kullanıldıkları alanlarda oldukça etkili olan İHA araçları, kullanım sırasında oluşturacağı tehditler dikkate alınmalıdır. Bu sebeple bu araçların tespit edilmesi üzerine çalışmalar oldukça önemlidir. Bu çalışmada İHA'ların tespiti, tanımlanması ve kayıt altına alınması önemli hale geldiği için yapılmıştır.

Bu çalışmada, normal bir bilgisayarın çalışma prensibine sahip taşınabilir, düşük maliyetli ve az enerji tüketebilen kullanışlı bir yapıya sahip olan Raspberry Pi 4 seçilmiştir. V2 kamerasının Raspberry Pi 4' e bağlantıları yapıldıktan sonra elde edilen görüntülerin takibi için bir kapasitif dokunmatik ekran gerekli donanımlarla birlikte bağlandı.

Görüntü verileri işlenerek nesne tespiti yapılabilmesi için, birer birer ve reel zamanlı nesne tanımayı gerçekleştirebilen SSD modeli sistemde kullanıldı. İhtiyaç olan donanımsal ekipmanlar ki bunlar raspbery sistemi ve bu sistemle uyumlu çalışabilen v2 pi kamera, dokunmatik ekran, güç sağlamak için motor ve enerji aksamı ile istenilen deneysel çalışma sistemi anlamlı bir şekilde oluşturuldu. Ayrıca google görüntü işleme alt yapısı kullanılmasıyla tüm ekipmanların kütüphaneleri ile eklenmiş çalışma bölgeleri birbiri ile uyumlu hale getirilerek, ekipmanlar arasında ki haberleşme kolayca sağlandı.

Kamera 360° dönen sistemiyle görüş açısını sürekli tarayarak canlı akışı sağlanarak her 2 saniye de bir kontrol sağlandı. Canlı akışa sürekli devam eden kamera ve algoritma sistemi, aynı nesneyi farklı konum ve açılarda görüş açısına kaç kere girerse girsin her durumda sinyal verip kaydı yapıldı. Bu işlem yaklaşık 2.5 saniye de gerçekleşti. Kayıt yapması gerçek zamanlı olup günün hangi saatinde olduğu ayrıntısını da böylece kaçırmamış oldu.

Karar hassasiyeti % 90 olarak manuel ayarlanan algoritma, herhangi bir öğrenilmiş görüntü ile % 90 ayarında eşleşme sağlarsa bunun İHA olduğuna karar verir ve o anda kapasitif ekranda dikkat logosu algortiması devreye girerek görüntü kayıt

edildi. Sonuç olarak model çalışma verimi % 94 olarak hesaplandı. Bu verim kapalı alanda gerçekleştirildi. Açık alanda ise % 99 oranında başarı görüldü.

Görüntü kalitesi algoritma girdisi için çok önemlidir. Bu nedenle kamera ne kadar iyi olursa sistemde o kadar verimli olmaktadır. Bu durumda, kullanılan yöntemin tek başına değil de diğer yöntemlerle karşılaştırmalı olarak çalışılması önerildi. Bir yöntemden çıkan sonucu başka yöntemlerde girdi olarak kullanarak yüksek verimli çalışmalar elde edilebileceği tahmin edildi.

Bu çalışma, özellikle savunma sanayi açısından önemi, bul ve müdahale et esasına dayanarak yapılmıştır. Müdahale edebilmek için tehdidin önce tespit edilmesi gerekmektedir. Bu çalışma tespit kısmı ile alakadar olmakta, fakat müdahale algoritması üzerine çalışılması halinde geliştirilebilir olduğu görüldü.

İlerleyen dönemde, müdahale ekipman ve algoritmasının uyumlu şekilde eklenmesi ile günümüz teknolojisinde istihbarat dahil olmak üzere bir çok alanda kullanılan İHA'nın kolayca ele geçirilmesi veya yok edilmesi üzerine çalışmalar yapılması hedeflenmiştir. Ayrıca, kullanım alanına veya taşıyıcı araçlara entegre durumuna uygun tasarım yapılması da planlanmaktadır.

KAYNAKLAR

- Adalı, E., İnce, G., Uyar, T., Sarıel, S., Köse, H., Temel, Z., Demircan, K., Girişken, Y., Burnaz, Ş., 2017. İnsanlaşan makineler ve yapay zeka. İstanbul Teknik Üniversitesi Vakfı Yayını Sayı 75.
- Ahonen, T., Hadid, A. and Pietikainen, M. 2006. Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence. IEEE Transactions on, 28(12), p.2037-2041.
- Alpaydın, E. 2011. Yapay Öğrenme. Türkiye Boğaziçi Üniversitesi Yayınevi
- Altunkurt, Ö., Kahriman, M. 2011. Gerçek Zamanlı Olarak, Anfis ile Renk Tabanlı Nesne Tespit ve Motorlu Sistem ile Takip Edilmesi. SDU Teknik Bilimler Dergisi, 1, 1, 1-5.
- Ataseven, B. 2013. Yapay Sinir Ağları ile Öngörü Modellemesi. Öneri Dergisi. 10, 39, 101-115.
- Bayatı, A. 2019. Evrimsel sinir ağlarını kullanarak drone tarafından elde edilen görüntülerde nesne tanıma. Selçuk Üniversitesi Fen bilimleri Enstitüsü Yüksek lisans Tezi.
- Bai, D., Wang, C., Zhang, B., Yi, X., Yang, X. 2017. Sequence searching with CNN features for robust and fast visual place recognition. Computers & Graphics. Special Issue on CAD/ Graphics.
- Belongie, S., Jitendra, M., Puzicha, J. 2002. Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4).
- Boureau, Y.L., Ponce, J., Lecun, Y. 2010. A Theoretical Analysis of Feature Pooling in Visual Recognition. Proceedings of the 27th International Conference.
- Cengil, E., Çınar, A. 2016. A New Approach For Image Classification: Convolutional Neural Network. European Journal of Technic, vol 6, number 2 .
- Dalal, N. & B. Triggs. 2005. Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. ss. 886-893.
- Demirtaş, A. 2019. Döner Kanatlı İnsansız Hava Araçları İle Görsel Verilere Dayalı Erken Yangın Algılama Sistemi. Ankara Üniversitesi Fen Bilimleri Yüksek Lisans Tezi, Ankara.
- Detienne, K. B.; Detienne D. H. & Joshi, S. A. 2003. Neural Networks As Statistical Tools For Business Researchers. Organizational Research Methods, 6(2), 236-265.

- Elfwinga, S., Uchibea, E., Doya, K. 2014. Expected Energy-Based Restricted Boltzmann Machine for Classification. Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna-son, Okinawa 904-0495, Japan
- Ferguson, M., Ak, R., Yung, L., Law, K. 2017. Automatic localization of casting defects with convolutional neural networks. IEEE International Conference on Big Data. Boston.
- Girshick, R., Donahue, J., Darrell, T and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. The IEEE Conference on 62 Computer Vision and Pattern Recognition (CVPR), pp. 580-587.
- Girshick, R. 2015. Fast r-cnn, Proceedings of the IEEE international conference on computer vision, pp. 1440-1448.
- Golovko, V., 2015. Towards Automatic Epileptic Seizure Detection in EEGS based on Neural Networks and Largest Lyapunov Exponent. International Journal of Computing. 14 (1). 36-47.
- Hacıoğlu, C. 2014. Derinlikli Öğrenme Kullanılarak Konuşmadan Uykululuk/Uykusuzluk Tespiti, Master thesis, İstanbul Technical University, İstanbul.
- Hanbay, K., Üzen, H. 2017. Nesne Tespit ve Takip Metotları; Kapsamlı Bir Derleme. Tr. Doğa ve Fen Dergisi.6. 2.
- Hartl, A., Arth, C. ve Schmalstieg, D. 2010. Instant segmentation and feature extraction for recognition of simple objects on mobile phones. IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 17-24.
- He, K., Zhang, X., Ren, S., and Sun, J. 2016. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.
- Heusch, G., Y. Rodriguez & S. Marcel. 2007. On the recent use of local binary patterns for face authentication. International Journal on Image and Video Processing Special Issue on Facial Image Processing. ss. 06-34.
- G. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P. and Sainath, T.N. 2012. Deep neural networks for acoustic modeling in speech recognition. The Shared views of four research groups, pp. 82- 97.
- ICAO, 2011, Unmanned Aircraft Systems (UAS), Cir. 328, AN/190, 2011.
- Jimut, B.P., Shalabh, A. 2019. Real Time Object Detection Can be Embedded on Low Powered Devices. International Journal of Computer Sciences and Engineering. 7, 2, 2347-2693.

- Kahveci, M. ve Can, N. 2017. İnsansız hava araçları; Tarihçesi, Dünyada ve Türkiye'deki Yasal Durumu. Selçuk Üniversitesi Mühendislik Bilim ve Teknik Dergisi c.5, s.4, ss. 511-535.
- Karakaya, F., Altun, H., Çavuşoğlu, M. A. 2009. FPGA tabanlı gömülü sistemde gerçek zamanlı nesne tanıma uygulamaları için HOG algoritmasının uygulanması. IEEE 17 Sinyal İşleme ve İletişim Uygulamaları Konferansı, Antalya.
- Karakoç, M. 2011. Görüntü İşleme Teknikleri ve Yapay Zekâ Yöntemleri Kullanarak Görüntü İçinde Görüntü Arama. Pamukkale Üniversitesi FBE Bilgisayar Mühendisliği A.B.D. Yüksek Lisans Tezi, Denizli.
- Kızrak, A. 2019. Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması. <https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a>.
- Kubat, M. 2015. Artificial Neural Networks. in An Introduction to Machine Learning, p.91–111.
- Koç, M. L.; Balas C. E. & Arslan, A. 2004. Taş Dolgu Dalgakıranların Yapay Sinir Ağları İle Ön Tasarımı. İMO Teknik Dergi, 15(4), 3351-3375.
- Krizhevsky, A., Sutskever, I. and . Hinton, G.E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, cilt 1, pp. 1097-1105.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1988. Gradient-based learning applied to document recognition. Proceeding of the IEEE, v 86, no. 11, pp. 2278-2324.
- Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L and Dollár, P. 2014. Microsoft COCO: Common Objects in Context. European conference on computer vision, pp. 740-755.
- Lim, S.H., S. R. Young, S.R and Patton, R.M. 2016. An analysis of image storage systems for scalable training of deep neural networks. cilt 5, no. 7, p. 11.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y and Berg, A.C. 2016. SSD: Single Shot MultiBox Detector. arXiv:1512.02325.
- Long, J., Shelhamer, E., Darrell, T. 2015. Fully convolutional networks for semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Neogy, A., Balakrishnan, S. N. ve Dagli C. H. 1992. Moving object recognition and guidance of robots using neural Networks. Aerospace and electronics conference, NAECON, proceedings of the iee 1992 national, 3, 883-886.
- Özcan, H. 2014. Çok Düşük Çözünürlüklü Yüz İmgelerinde Derin Öğrenme Uygulamaları. Yüksek Lisans Tezi, Deniz Harp Okulu, Bilgisayar Müh. Bölümü.

- Pattanayak, S. (2017). Introduction to Deep-Learning Concepts and Tensorflow. Pro Deep Learning with Tensorflow,89-152. doi:10.1007/978-1-4842-3096-1_2
- Russel. S. ve Norving. P. 2009. Artificial intelligence. A Modern Approach Prentice Hall. p. 17.
- Russ, J.C. 1999. The Image Processing Hand Book, Third Edition, CRC Press
- Seung, H. B. 2019. Object Detection Based on Region Decomposition and Assembly. Incheon National University, 119 Academy-ro, Yeonsu-gu, Incheon, 22012, Korea
- Simonyan, K. and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. ArXiv Prepr. ArXiv, pp. 1409-1556.
- Solak, S., Altınışik, U. 2017. Görüntü İşleme Teknikleri ve Kümeleme Yöntemleri kullanılarak Fındık Meyvesinin Tespit ve Sınıflandırılması. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi.
- Srivastava, N., Hinton, G., Krizhevsky, A., I. Sutskever, I., and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, pp. 1930-1958.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich. 2015. A. Going Deeper with Convolutions. Computer Vision and Pattern Recognition.
- Toprak, Ö. 2019. Görüntü İşleme Teknikleri ile Yaş Tahmini. Maltepe Üniversitesi Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi. İstanbul.
- Tan, Z. 2019. Derin Öğrenme Yardımıyla Araç Sınıflandırma. Fırat Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.
- Türkoğlu, İ. 2003. Örüntü tanıma sistemleri ders notları
- Wang, Z., Wang, Z., Zhang, H., & Guo, X. (2017). A Novel Fire Detection Approach Based on CNN-SVM Using Tensorflow. Intelligent Computing MethodologiesnLecture Notes in Computer Science,682-693. doi:10.1007/978-3-319-63315-2_60
- Wu, X., Kumar V., Quinlan J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., Steinberg, D. 2008. Top Ten Algorithms In Data Mining. Springer.
- Yiğit, A., Temizel, A. 2010. Termal ve Görünür Bant İmgeleri Kullanılarak Terk Edilen Nesne Tespiti. IEEE 18. Sinyal İşleme ve İletişim Uygulamaları Kurultayı. Diyarbakır.
- Yurtoğlu, H. 2005. Yapay Sinir Ağları Metodolojisi ile Önörü Modellemesi: Bazı Makroekonomik Değişkenler için Türkiye Örneği. Uzmanlık Tezi, Devlet Planlama Teşkilatı.

Zeren, M.T. 2020. İnsansız Hava Araçları ve Uydu görüntülerinden Elde Edilen Veri Seti ile Havaalanlarının Tespitinin Yapılmasında SSD ve Faster R-CNN Algoritmalarının Karşılaştırılması

Zheng, Y., Meng, Y. ve Jin, Y. 2010. Fusing Bottom-up and Top-down Pathways in Neural Networks for Visual Object Recognition. Neural Networks (IJCNN). International Joint Conference, 1-8.

