

**GÖRME ENGELLİ BİREYLER İÇİN GERÇEK ZAMANLI GÖMÜLÜ
NESNE TANIMA SİSTEMİNİN TASARIMI VE UYGULANMASI**

YÜKSEK LİSANS TEZİ

SONAY DUMAN

**MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**MERSİN
AĞUSTOS- 2019**

**GÖRME ENGELLİ BİREYLER İÇİN GERÇEK ZAMANLI GÖMÜLÜ
NESNE TANIMA SİSTEMİNİN TASARIMI VE UYGULANMASI**

YÜKSEK LİSANS TEZİ

SONAY DUMAN

**MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

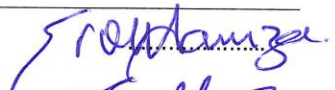


**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**Danışman
Dr.Öğr. Üyesi Abdullah ELEWİ
Prof.Dr. Zeki YETGİN**

**MERSİN
AĞUSTOS- 2019**

ONAY

Sonay DUMAN tarafından Dr.Öğr.Üyesi Abdullah ELEWİ ve Prof.Dr. Zeki YETGİN danışmanlığında hazırlanan "Görme Engelli Bireyler İçin Gerçek Zamanlı Gömülü Nesne Tanıma Sisteminin Tasarımı Ve Uygulanması" başlıklı çalışma aşağıda imzaları bulunan jüri üyeleri tarafından 7 Ağustos 2019 tarihinde yapılan Tez Savunma Sınavı sonucunda oy birliği ile Yüksek Lisans Yeterlik tezi olarak kabul edilmiştir.

Görevi	Ünvanı, Adı ve Soyadı	İmza
Başkan	Prof.Dr. Hamza EROL	
Üye	Dr.Öğr.Üyesi Abdullah ELEWİ	
Üye	Dr.Öğr.Üyesi Mehmet Ali AKTAŞ	

Yukarıdaki Jüri kararı Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29/8/2019 tarih ve 2019.24/1326 sayılı kararıyla onaylanmıştır.



Bu tezde kullanılan özgün bilgiler, şekil, tablo ve fotoğraflardan kaynak göstermeden alıntı yapmak 5846 sayılı Fikir ve Sanat Eserleri Kanunu hükümlerine tabidir.

ETİK BEYAN

Mersin Üniversitesi Lisansüstü Eğitim-Öğretim Yönetmeliğinde belirtilen kurallara uygun olarak hazırladığım bu tez çalışmasında,

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlâk kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak kullandığımı,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Bu tezin herhangi bir bölümünü Mersin Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı,
- Tezin tüm telif haklarını Mersin Üniversitesi'ne devrettiğimi

beyan ederim.

ETHICAL DECLARATION

This thesis is prepared in accordance with the rules specified in Mersin University Graduate Education Regulation and I declare to comply with the following conditions:

- I have obtained all the information and the documents of the thesis in accordance with the academic rules.
- I presented all the visual, auditory and written informations and results in accordance with scientific ethics.
- I refer in accordance with the norms of scientific works about the case of exploitation of others' works.
- I used all of the referred works as the references.
- I did not do any tampering in the used data.
- I did not present any part of this thesis as an another thesis at Mersin University or another university.
- I transfer all copyrights of this thesis to the Mersin University.

07/08/2019



Soray DUMAN

ÖZET

GÖRME ENGELLİ BİREYLER İÇİN GERÇEK ZAMANLI GÖMÜLÜ NESNE TANIMA SİSTEMİNİN TASARIMI VE UYGULANMASI

Görme engelli bireyler, günlük yaşamlarında birçok zorlukla karşılaşmaktadır. Genel olarak beyaz baston kullanan bireyler, etraflarındaki nesnelere algılamak, dışarıda engellere takılmadan yürümek veya alışveriş yapmak gibi sıradan aktiviteleri gerçekleştirirken zorlanmakta ve yardıma ihtiyaç duymaktadır. Bu sebeplerden dolayı, günümüzde görme engelli bireylere yardımcı olabilecek etkili çözümler üretmek adına, bilişim alanında çok sayıda çalışma yapılmaktadır. Bu çalışmada, bilgisayar görüşü alanında görme engelli bireyler için konvolüsyonel yapay sinir ağları kullanılarak geliştirilmiş, bireylerin etrafındaki nesnelere ve insanları algılamasına ve mesafelerini tam olarak tahmin etmelerine yardımcı olmak için taşınabilir bir sistem tasarlanarak uygulanmıştır. Tasarlanan sistemde, öncelikle mesafe tespitini tek(monocular) kamera ile gerçekleştirmek 1 ile 10 metre arasında görüntüler alınarak bir veri seti oluşturulmuş ve seçilen özelliklere göre veriler 7 farklı regresyon modeli ile eğitilip test edilmiştir. Sistemin prototipi, Raspberry Pi üzerine monte edilmiş tek bir kamera ile You Only Look Once adı verilen, konvolüsyonel yapay sinir ağı tabanlı gerçek zamanlı bir nesne algılama tekniği kullanılarak oluşturulmuştur. Mesafe tespiti modülünü prototip üzerinde stabil ve güvenilir bir hale getirmek amacıyla, Raspberry Pi kamerası ile 2 ile 10 metre arasında görüntüler alınarak ikinci bir veri seti oluşturulmuştur. Geliştirilen prototip, nesne tanıma yaparak tespit edilen nesnelere (insan) mesafesini tahmin edebilmekte ve bu verileri görme engelli bireylere sesli olarak iletmektedir. Sonuçlar, sistemin nesnelere tespit edebildiğini ve tespit edilen insanların mesafelerini en yüksek % 98,8 doğrulukla Rastgele Orman modeli ile tahmin edebildiğini göstermektedir.

Anahtar Kelimeler: YOLO, CNN, Raspberry Pi, Mesafe Tespiti, Görme Engelli, Bilgisayar Görüşü

Danışman: Dr. Öğr. Üyesi Abdullah ELEWİ, Mersin Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Mersin.

2.Danışman: Prof. Dr. Zeki YETGİN, Mersin Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Mersin

ABSTRACT

DESIGN AND IMPLEMENTATION OF REAL-TIME EMBEDDED OBJECT DETECTION SYSTEM FOR VISUALLY IMPAIRED INDIVIDUALS

Visually impaired individuals face many difficulties in their daily lives. Generally, people who use white canes need help in performing ordinary activities such as detecting objects around them, walking outside without obstructions or shopping. For these reasons, there are many studies in the field of informatics in order to produce effective solutions that can help visually impaired individuals. In this study, convolutional neural networks have been used for helping visually impaired individuals, and a portable system has been designed and implemented to help these individuals in perceive objects and people around them and estimating their distance. In the designed system, a data set was created by taking images between 1-10 meters in order to realize the distance determination with a monocular camera and the data was trained and tested with 7 different regression models according to the selected features. The prototype of the system was built using a single camera mounted on Raspberry Pi board, using a convolutional neural network based real-time object detection technique called You Only Look Once. In order to make the distance detection module stable and reliable on the prototype, a second data set was created by taking images between 2 and 10 meters with Raspberry Pi camera. The prototype is able to estimate the distance of the detected objects and transmits these data to visually impaired individuals. The results show that the system is able to detect objects and estimate the distances of the detected people with 98,8 % accuracy using the Random Forest model.

Keywords: YOLO, CNN, Raspberry Pi, Distance Estimation, Visually Impaired, Computer Vision

Advisor: Asst. Prof. Abdullah ELEWİ, Mersin University, Computer Engineering, Mersin.

Co-Advisor: Prof. Zeki YETGİN, Mersin University, Computer Engineering, Mersin.

TEŐEKKÜR

Bu tez alıřmasının tasarlanması, araştırılması, yürütülmesinde ve tamamlanmasında desteęini esirgemeyen, engin bilgi ve tecrübelerinden faydalandığım, planlama ve yönlendirmeleri sayesinde tez alıřmamın bilimsel temeller doęrultusunda en iyi şekilde hazırlanmasını saęlayan deęerli hocalarım, Dr. Öğr. Üyesi Abdullah ELEWİ ve Prof. Dr. Zeki YETGİN'e teőekkür ederim.

Üzerimdeki emekleri ve hayatımın her aşamasında olduęu gibi tez alıřmamı hazırlama sürecinde de bana olan sonsuz destekleri için deęerli annem Sevim DUMAN, babam Nurettin DUMAN, ağabeylerim Koray ve Fırat DUMAN' a, aynı zamanda sürecin tamamında desteęini esirgemeyen, deęerli Rojin YAZAR'a tüm kalbimle teőekkür ederim.



İÇİNDEKİLER

	Sayfa
İÇ KAPAK	ii
ONAY	iii
ETİK BEYAN	iv
ÖZET	v
ABSTRACT	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
TABLolar DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
KISALTMALAR ve SİMGELER	xiv
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMALARI	4
2.1. Görme Engelli Bireyler İçin Geliştirilmiş Olan Konvolüsyonel Yapay Sinir Ağ Tabanlı Yardımcı Teknolojiler	4
2.1.1. Kör İnsanlar Görsün (Let Blind People See)	4
2.1.2. Görme Yetisi Olmadan Görmek (Seeing Without Sight)	6
2.1.3. Görme Engelli Bireyler için Derin Öğrenme Tabanlı Bir Yaklaşım (Object Detection Featuring 3D Audio Localization for Microsoft HoloLens)	8
2.1.4. Sanal Rehber Köpek (Virtual Guide Dog)	10
2.2. Uzaklık Tahmini ve Nesne Tespiti için Kullanılan Diğer Yöntemler/Sistemler	12
2.2.1. Sensör Tabanlı Sistemler	12
2.2.2. Video Tabanlı Sistemler	15
2.2.2.1. Tek(Monocular) Kamera Tabanlı Sistemler	15
2.2.2.2. Stereo Kamera Tabanlı Sistemler	19
2.2.2.3. RGB-D Kamera Tabanlı Sistemler	20
3. MATERYAL ve YÖNTEM	21
3.1. Ham Veri Setinin Oluşturulması	22
3.2. YOLO'nun Çalışma Şekli ve NNPack Darknet Kütüphanesi	23
3.2.1. Bilgisayar Görüşü ve Konvolüsyonel Yapay Sinir Ağları	24
3.2.2. YOLO'nun Çalışma Şekli	26
3.2.3. NNPACK Darknet Kütüphanesi	26
3.3. Regresyon	28

	Sayfa
3.3.1. Rassal Orman	29
3.3.2. Karar Ağacı	31
3.3.3. Adaboost Karar Ağacı	32
3.3.4. K-En Yakın Komşu	35
3.3.5. Adaboost K-En Yakın Komşu	37
3.3.6. Topluluk (Ensemble) K-En Yakın Komşu	38
3.3.7. MLP(Çok Katmanlı Algılayıcı) Regresör	39
3.4. Google Text to Speech Modülü ile Sesli Bildirim Oluşturulması	41
4. BULGULAR ve TARTIŞMA	43
4.1. Uzaklık Tespiti için Uygulanan Regresyon Tekniklerinin Teorik Test Sonuçları	43
4.2. Prototipin Yapısı	49
4.2.1. Mesafe Tespiti Sonuçları	55
4.2.2. Gerçek Zamanlı Nesne Tanıma Sonuçları	60
5. SONUÇLAR ve ÖNERİLER	62
KAYNAKLAR	64
EKLER (<i>Var ise</i>)	-
ÖZGEÇMİŞ	71

TABLÖLAR DİZİNİ

	Sayfa
Tablo 2.1. Kullanıcı, İstemci ve Sunucu Etkileşimi	9
Tablo 3.1. Veri Setinden Örnek Bir Veri	23
Tablo 4.1. Raspberry Pi 3 Model B Özellikleri	50



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Veri Akış Şeması	5
Şekil 2.2. Prototip Test Sonucu	6
Şekil 2.3. Sistemin Çalışma Şekli	8
Şekil 2.4. Test Sonuç Görüntüleri	8
Şekil 2.5. Hareket Tanıma Siteminin Çalışma Şekli	10
Şekil 2.6. Sistem Mimarisi ve Prototip Görüntüleri	11
Şekil 2.7. Guidance, SmartCane, Batcane, Necklace sonar	13
Şekil 2.8. CyARM	13
Şekil 2.9. Smart-Robot	14
Şekil 2.10. RoboCart	14
Şekil 2.11. ShopTalk	15
Şekil 2.12. Arianna	16
Şekil 2.13. Monacu ve Ark.	16
Şekil 2.14. Manikandan ve Ark., Springer ve Ark.	18
Şekil 2.15. Prototipin Görünümü	18
Şekil 2.16. Smart Walker	19
Şekil 2.17. Everding ve Ark.	19
Şekil 2.18. Neto ve Ark	20
Şekil 2.19. INSANA	20
Şekil 3.1. Sistemin Akış Şeması	22
Şekil 3.2. Nesne Tanıma için Örnek Görüntü	24
Şekil 3.3. Altı farklı bakış açısından çekilen bir nesne	25
Şekil 3.4. LeNet Mimarisi	25
Şekil 3.5. YOLO'nun Çalışma Şekli	26
Şekil 3.6. X ve Y İlişkisi	29
Şekil 3.7. Rassal Orman Çalışma Şekli	29
Şekil 3.8. Rassal Orman Python Kodu	30
Şekil 3.9. Karar Ağacı Örneği	31
Şekil 3.10. Standart Sapma Hesaplaması	32
Şekil 3.11. Torbalama ve Güçlendirme Çalışma Prensibi	33
Şekil 3.12. Adaboost Karar Ağacı'nın Çalışma Prensibi	33
Şekil 3.13. Adaboost Karar Ağacı Sonucu	34
Şekil 3.14. Adaboost Karar Ağacı Python Kodu	35
Şekil 3.15. KNN'in Yapısı	36

	Sayfa
Şekil 3.16. KNN Python Kodu	37
Şekil 3.17. Adaboost KNN Python Kodu	38
Şekil.3.18. Topluluk (Ensemble) KNN Python Kodu	39
Şekil 3.19. MLP Sinir Ağ Yapısı	40
Şekil 3.20. MLP Python Kodu	41
Şekil 3.21. Konuşma Sentezi Çalışma Şekli	42
Şekil 4.1. Uzaklık ve Vücut Uzunluğu İlişkisi Grafiği	44
Şekil 4.2. Uzaklık ve Vücut Genişliği İlişkisi Grafiği	44
Şekil 4.3. Uzaklık ve Yüz Uzunluğu İlişkisi Grafiği	45
Şekil 4.4. Uzaklık ve Yüz Genişliği İlişkisi Grafiği	45
Şekil 4.5. R ² Değerleri Grafiği	46
Şekil 4.6. MAE Değerleri Grafiği	47
Şekil 4.7. MSE Değerleri Grafiği	47
Şekil 4.8. Mesafe Doğruluk Değerleri Grafiği	48
Şekil 4.9. MMM Değerleri Grafiği	49
Şekil 4.10. Raspberry Pi 3 Model B ve Kamera Modülü	49
Şekil 4.11. Sistemin Modülleri	50
Şekil 4.12. Veri Akış Şeması	51
Şekil 4.13. Python Ortamı Kurulum Komutu	51
Şekil 4.14. Ninja Kurulum Komutları	52
Şekil 4.15. Ninja Kurulum Komutları	52
Şekil 4.16. NNPack Kurulum Komutları	52
Şekil 4.17. NNPack'i Oluşturmak İçin Gerekli Komut	53
Şekil 4.18. Darknet Kurulum Komutları	53
Şekil 4.19. YOLOv3-tiny Kullanma Komutları	53
Şekil 4.20. Yüz Tespiti Çalıştırma Komutu	53
Şekil 4.21. YOLO ile Yüz Tespiti Sonuçları	54
Şekil 4.22. 4 metre Mesafeden Vücut ve Yüz Boyutları	55
Şekil 4.23. Uzaklık ve Vücut Uzunluğu İlişkisi Grafiği	56
Şekil 4.24. Uzaklık ve Vücut Genişliği İlişkisi Grafiği	56
Şekil 4.25. Uzaklık ve Yüz Uzunluğu İlişkisi Grafiği	57
Şekil 4.26. Uzaklık ve Yüz Genişliği İlişkisi Grafiği	57
Şekil 4.27. Mesafe Doğruluk Değerleri Grafiği	58
Şekil 4.28. R ² Değerleri Grafiği	58
Şekil 4.29. MAE Değerleri Grafiği	59

	Sayfa
Şekil 4.30. MSE Değerleri Grafiği	59
Şekil 4.31. MMM Değerleri Grafiği	60
Şekil 4.32. YOLOv3 tiny ile Nesne Tespiti Anı	61



KISALTMALAR ve SİMGELER

Kısaltma/Simge	Tanım
CNN	Convolutional Neural Networks (Konvolüsyonel Yapay Sinir Ağları)
R-CNN	Region Based Convolutional Neural Networks(Bölge Tabanlı Konvolüsyonel Yapay Sinir Ağları)
YOLO	You Only Look Once
GPU	Graphic Processing Unit (Grafik İşleme Birimi)
VOC	Visual Object Classes (Görsel Nesne Sınıfları)
ETA	Electronic Travel Aid (Elektronik Gezi Yardımcısı)
MLP	Multi Layer Perceptron(Çok Katmanlı Algılayıcı)
KNN	K Nearest Neighbor (K En Yakın Komşu)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
MSE	Mean Square Error (Ortalama Kare Hata)
R ²	R-squared (Belirleme Katsayısı)
MMM	Mean of Min over Max (Minimumun Maksimuma Oranının Ortalaması)
gTTS	Google Text-to-Speech (Google Metinden Konuşmaya Dönüştürme)

1. GİRİŞ

Dünya Sağlık Örgütü'nün 2017 yılında yapmış olduğu araştırmaya göre Dünya çapında 253 milyon görme engelli birey yaşamaktadır. Bu bireylerin 36 milyonu görme yetisini tamamen kaybetmiş bireylerden oluşmaktadır. 2020 yılı itibariyle 38,5 milyon olması öngörülen bu sayının, 2050 yılında ise neredeyse üçe katlanarak 115 milyon olması beklenmektedir[1]. Görme engelli bireylerin günlük yaşamlarında kapalı ve açık mekânlarda rahatça yürümeleri, takılabilecekleri engelleri önceden tespit edebilmeleri, alışık olmadıkları bir ortamda önlerine çıkan nesnelere neler olduğunu anlamaları veya aradıkları nesnelere bulmaları oldukça zordur. Özellikle araç ve insan trafiğinin yoğun olduğu, çukur, tümsek, elektrik direği ve tabela gibi engellerin sık yer aldığı cadde ve sokaklarda büyük zorluklarla karşı karşıya kalmaktadırlar. Günümüzde görme engelli bireyler tarafından bahsedilen zorlu durumlara karşı en yaygın şekilde kullanılan yardımcı cihaz beyaz bastondur. Beyaz bastonlar, görme engelli bireylerin yayalar ve sürücüler tarafından fark edilmelerini ve zeminle birebir temas ederek engelleri fark etmelerini sağlamak amacıyla üretilmiştir. İlkel bir teknoloji olmasına rağmen kullanımının kolay ve maliyetinin ucuz olması sebebiyle daha çok tercih edilmektedir[2]. Maalesef sadece baston kullanımı, görme engelli bireylerin günlük yaşamlarında ihtiyaçlarını tamamen bağımsız olarak gerçekleştirebilmelerine yetmemektedir. Örneğin; istedikleri bir nesneyi bulma, alışveriş yapma ve yön bulma gibi durumlarda çoğu zaman başka bir kişinin yardımına ihtiyaç duymaktadırlar. Bahsedilen problemlerden yola çıkarak geliştirilen yeni yardımcı teknolojilerin temeli Bilgisayar Görüşü(Computer Vision) tekniklerine dayanmaktadır. Bilgisayar görüşü, insanların doğuştan var olan görme yetilerini bilgisayara vermeyi amaçlayan bir çalışma alanıdır. İnsanlar, görme yetilerini yitirmedikleri sürece etraflarındaki 3 boyutlu dünyayı anlayabilmek için gözlerini ve beyinlerini kullanırlar. Gördükleri nesnelere ne olduğunu ve nerede olduğunu bu sayede tanımlayabilirler. Benzer tanımlamayı, çeşitli yazılım ve donanımlar kullanarak bilgisayara vermek de mümkündür[3]. Bu sayede görme yetisini kaybetmiş olan bireylere çeşitli yardımcı teknolojiler tasarlanabilmektedir.

Geçmişten günümüzde, bilgisayarlı görme teknolojileri kullanılarak görme engelli bireylere asistanlık edebilecek farklı araçlar geliştirilmiştir. Apple ve Google Play Store'da yer alan mobil uygulama TapTapSee[4], görme engelli kullanıcılar tarafından çekilen bir fotoğrafı tanımlamak için bilgisayarlı görme teknolojileri kullanılarak geliştirilmiştir. Benzer şekilde her iki mobil platformda yer alan Text Detective[5] adlı uygulama mobil cihaz üzerinden çekilen resimdeki metinleri sesli olarak okumaktadır. Günümüzde, bilgisayar görüşü teknolojileriyle birlikte konvolüsyonel yapay sinir ağları ile yapılan çalışmalar da oldukça popüler hale gelmiştir. Görme engelli bireyler için, bilgisayar görüşü temelinde konvolüsyonel yapay sinir ağ

teknikleri kullanılarak üst düzey teknolojiler (state-of-art) geliştirilmeye başlanmıştır. Bununla birlikte, literatürde konvolüsyonel yapay sinir ağları kullanmayan ve farklı yöntemlerle geliştirilen birçok çalışma bulunmaktadır. Bunlara örnek olarak, Microsoft Seeing AI[6], Third Eye[7] ve vOICe[8] verilebilir.

Bu çalışmada görme engelli bireyler için nesne tanıma ve mesafe tespiti yapabilen, bu sonuçları kullanıcılara sesli bildirimlerle aktaran bir sistem tasarlanarak sistemim prototipi geliştirilmiştir. Literatürde yer alan çalışmalar genellikle nesne tanıma amacıyla geliştirilmiş olan sistemlerdir. Nesne tanıma ile birlikte mesafe tespiti yapabilen sistemlerin sayısı daha azdır. Bunun temel sebebi, mesafe tespiti yapabilmek için genellikle kullanılan sensör ve kameraların maliyetlerinin yüksek ve ağırlıkları sebebiyle zor taşınabilir olmalarıdır. Mesafe tespitinde daha az kullanılan bir yöntem olan tek(monocular) kamera ile tespit maliyet ve taşınabilirlik açısından daha uygun olmasına rağmen daha zor algoritmalar gerektirdiği için çok fazla çalışılmamaktadır. Bu doğrultuda, tez kapsamında geliştirilen sistemin bu alana katkı sağlayacağı söylenebilir.

Bu tez çalışmasında, görme engelli bireyler için gerçek zamanlı nesne tanıma ve mesafe tespiti yapabilen yardımcı bir sistem tasarlanarak uygulanmıştır. Tez kapsamında gerçekleştirilen çalışmada aşağıdaki yenilikler ortaya konmuştur.

1. Görme engelli bireylerin açık ve kapalı ortamlarda etraflarındaki nesne ve insanları tanımlayıp mesafe tespiti ile ne uzaklıkta olduklarını belirleyerek kullanıcılara sesli bildirimlerle yönlendirme yapabilen yardımcı bir sistemin tasarlanması
2. Tasarlanan yardımcı sistemin, nesne tanıma, mesafe tespiti ve sesli bildirim modüllerinden oluşan taşınabilir bir prototipinin geliştirilmesi
3. Tek(Monocular) kamera ile elde edilen özellikler kullanılarak mesafe tespitinde özgün bir metodun uygulanması
4. Elde edilen verilerin farklı regresyon algoritmaları ile analizlerini yaparken kullanılabilir olan "Mean of Min over Max(MMM)" parametresinin kullanılması
5. 1 ve 10 metre aralığında 1000 adet kayıttan oluşan, bu kayıtların özellikler bazında analiz edilmesiyle elde edilen verilerle birinci veri setinin ve sistemin bir prototip üzerinde uygulanması için Raspberry Pi kamera modülü kullanılarak 2 ve 10 metre aralığında 450 kayıttan oluşan, bu kayıtların özellikler bazında analiz edilmesiyle elde edilen verilerle ikinci bir setinin oluşturulması

Bu tez çalışması Giriş, Kaynak Araştırmaları, Materyal ve Yöntem, Bulgular ve Tartışma, Sonuç ve Öneriler olmak üzere 5 ana bölümden oluşmaktadır. Çalışma kapsamında, 1. Bölüm olan "Giriş" kısmında tezin konusuna paralel olarak görme engelli bireylerin karşılaştıkları

genel zorluklar, görme engelli bireyler için yardımcı teknolojiler alanında olan gelişmeler ve tez çalışması genel hatlarıyla anlatılmıştır. Tezin ilgili olduğu alanda yapılmış olan çalışmalarla ilgili literatür taraması tezin 2. Bölümü olan “Kaynak Araştırmaları” kısmında ayrıntılı olarak anlatılmıştır. Kaynak araştırmasının ardından, tezin 3. Bölümü olan “Materyal ve Yöntem” kısmında tez çalışmasında kullanılan, gerçek zamanlı nesne tespiti sisteminden, mesafe tespiti modülü için geliştirilen yöntem ve bu yöntemin uygulanma şekline ve son olarak da kullanıcılara sesli bildirimlerin iletim şemasından bahsedilerek çalışma için kullanılan tüm materyal ve yöntemlerle ilgili ayrıntılı bilgiler verilmiştir. 4. Bölüm olan “Bulgular ve Tartışma” kısmında çalışmanın tasarımı, uygulanması ve elde edilen sonuçlar ile ilgili bilgiler verilerek sonuçlar gerekli grafiklerle sunulmuştur. Tezin son bölümünde, çalışmanın sonuçları genel olarak ele alınarak öneriler “Sonuç ve Öneriler” başlığı altında verilmiştir.



2. KAYNAK ARAŞTIRMALARI

Bu bölümde görme engelli bireyler için geliştirilmiş olan konvolüsyonel yapay sinir ağ tabanlı yardımcı teknolojiler ve uzaklık tahmini ve nesne tespiti için kullanılan diğer yöntemler/sistemlerin tasarımları, uygulamaları ve kullanım şekilleri incelenmiştir.

2.1. Görme Engelli Bireyler İçin Geliştirilmiş Olan Konvolüsyonel Yapay Sinir Ağ Tabanlı Yardımcı Teknolojiler

Bu bölümde, literatürde yer alan görme engelli bireylere asistanlık etmesi amacıyla konvolüsyonel yapay sinir ağları kullanılarak geliştirilmiş olan yardımcı teknolojiler incelenmiştir. Konvolüsyonel yapay sinir ağları (Convolutional Neural Network-CNN) ileri yönlü bir yapay sinir ağı türüdür. Algoritmasının temeli, hayvanların görme merkezinden esinlenerek ortaya çıkmıştır. Burada matematiksel konvolüsyon işlemi, hayvanlarda bir nöronun kendi uyarı alanından uyarılara verdiği yanıt olarak düşünülebilir [9], [10], [11]. Bir veya daha fazla konvolüsyonel katman, altörnekleme (subsampling) katman ve standart çok katmanlı bir sinir ağı gibi bir veya daha fazla tam bağlı katman CNN'yi oluşturur [12]. CNN algoritmaları ses ve görüntü işleme alanlarında oldukça yaygın halde kullanılmaya başlanmış ve özellikle görüntü işleme üzerinde en iyi sonuçlar(state-of-art) elde edilmiştir. Son 10 yıllık sürede yaygınlaşan CNN tabanlı yardımcı teknolojilerle başarılı sonuçlar elde edilmeye başlanmıştır. Bu teknolojiler sayesinde, görme engelli bireyler etraflarındaki nesnelere tanımlayabilmekte ve engel teşkil eden nesnelere karşı önlem alabilmektedirler.

2.1.1. Kör İnsanlar Görsün (Let Blind People See)

Let Blinde People See, Standord Üniversitesi'nde üç araştırmacı tarafından gerçekleştirilmiş bir çalışmadır[13]. Çalışmanın amacı, görme engelli bireyler için nesne tanıma yaparak, nesnelere ne olduğunu sesli bir şekilde onlara iletmeyi sağlayacak bir sistem oluşturmaktır. Sistem birden fazla modül içermektedir. Bu modüller;

1. İstemci Modülü: Video kaydı almak için taşınabilir bir kamera(Microsoft Kinect[14] veya GoPro[15])
2. Sunucu Modülü: Kamera tarafından alınan videonun sunucu üzerinde gerçek zamanlı olarak işlenerek, nesne tanıma yapılmasını sağlayan konvolüsyonel yapay sinir ağı tabanlı YOLO algoritması ve nesnelere 3B konumlarını 3B ses şeklinde kullanıcıya bildirilmesini sağlayan Unity Oyun Motoru[16] tabanlı 3B ses oluşturma uygulaması

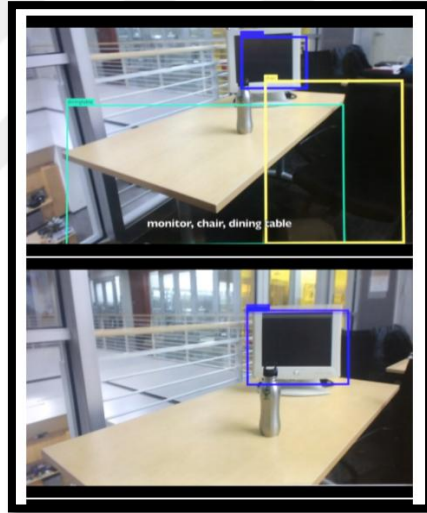
Çalışmada, öncelikle nesne tanıma yapmak amacıyla farklı algoritmalar denenmiştir. İlk olarak R-CNN[17] algoritması denenmiş, fakat R-CNN'in bölge tabanlı yapıda olması ve her bir bölgeyi sınıflandırmak için ayrı ayrı konvolüsyonel ağ uygulaması sistemi yavaşlatarak gerçek zamanlı bir sistemden oldukça uzak kılmıştır. Yavaş çalışması, karmaşık eğitim süreçlerine ve fazla hafızaya ihtiyaç duyması R-CNN'in gerçekleştirilmek istenen sistem için tercih edilmemesine sebep olmuştur. Bunun ardından, Fast R-CNN[18] ve Faster R-CNN[19] algoritmalarını denenmiş, bu iki algoritmanın hız ve güvenilirlik açısından R-CNN'e göre daha iyi olmasına rağmen, karmaşık optimizasyon ve eğitim süreçlerinden ötürü tercih edilmediğini belirtilmiştir. Son adımda YOLO(You Only Look Once)[20] algoritması kullanılarak, YOLO'nun tasarlanan gerçek zamanlı sisteme çok uygun bir yapıda çalıştığı, hız ve nesne tanıma doğruluğu açısından tüm algoritmalar arasından en iyi sonuçlara ulaştığı belirtilmiştir. İlk adım olarak elde edilen videodan YOLO ile nesne tanıma yapan sistemde, ikinci adım ise nesnenin kullanıcıya olan uzaklığının belirlenmesidir. Çalışmada, uzaklık tespiti için iki farklı teknik denenmiştir. Bu tekniklerden ilki, Microsoft Kinect kamera kullanılarak Kinect'in RGB görüntüler üzerinden gerçek zamanlı derinlik haritası çıkarmasından faydalanmak olmuştur. Fakat Kinect'in ağır olması sebebiyle taşınabilirliğe uygun olmaması ve kablosuz bağlantı özelliğine sahip olmaması kullanımına engel teşkil etmiştir. Benzer şekilde farklı kameralar düşünülse de, bu tarz kameralar yüksek işlem kaynağı (Grafik İşleme Birimi-GPU) gerektirmesi sebebiyle tercih edilmemiştir. İkinci teknik olarak, kamerayı kullanmak yerine elde edilen görüntü üzerinden uzaklık tahmini yapmak denenmiştir. Bu teknikte, kameranın bakış açısı ve nesnenin yer aldığı sınırlı kutu belirlendiğinde, sınırlı kutunun merkezindeki pikselin konumuna göre yön tahmini yapılabilecektir. Daha sonra derinlik tahmini yapabilmek için, her sınıfa özgü "varsayılan yükseklik" değeri atanmıştır. Örneğin; insanlar için varsayılan yükseklik değeri 5,5 fit, sandalyeler için de 2,5 fit olarak belirlenmiştir. 20 farklı sınıf için bu değerler kodlanmıştır.

Sonuç olarak, sınırlı kutunun yüksekliği ve varsayılan yükseklik değerine göre derinlik tahmini yapılabilmektedir. İkinci yöntem tercih edildiği için, hafif, kolay taşınabilir ve kablosuz bağlantı özelliğine sahip olan GoPro Hero 3 kamerası kullanılmıştır. Sistemin veri akış şeması Şekil 2.1.'de gösterilmektedir.



Şekil 2.1. Veri Akış Şeması [13]

YOLO algoritması yerel sunucu üzerinde yüksek performanslı grafik işleme birimi ile çalışarak nesne tanıma yapmakta ve bilgiyi direkt olarak Unity3D ses oluşturucusuna göndererek sesli bildirim sağlanmaktadır. Sürekli aynı nesne için bildirim vermemek adına algoritma, tanımlanan nesneyle ilgili ne zaman bildirim vermesi gerektiğine de karar vermektedir. Örneğin, program “sandalye” nesnesini tanımlayıp kullanıcıya sesli olarak bildirdikten sonra 5 saniye geçene kadar tekrar “sandalye” nesnesi için sesli bildirim vermemektedir. Çalışma prototipi gerçekleştirilerek test edilmiştir (Şekil 2.2). Test sonucunda, sistemin görme engelli kullanıcılar üzerinde arttırılmış gerçeklik hissi uyandırdığı ve kullanıcıların bir sandalyeyi algılayarak sandalyeye doğru yönelmelerinin ortalama 15 saniye sürdüğü gözlemlenmiştir. Kullanıcıların sistemde şikâyet ettiği nokta ise sesli bildirimlerde karmaşıklık yaşamaları olmuştur. Bunun için araştırmacılar, yeni yöntemler geliştirerek sesli bildirimlerin, soldan sağa doğru nesnelere sırayla sayacak şekilde değiştirilebileceğini veya kullanıcıların o esnada ihtiyaç duydukları nesnelere adını söyleyerek sisteme o yönde bir tarama yapması için bildirim verebilecekleri bir özellik eklenebileceğini belirtmişlerdir.



Şekil 2.2. Prototip Test Sonucu [13]

2.1.2. Görme Yetisi Olmadan Görmek (Seeing Without Sight)

“Seeing Without Sight” bilgisayar görüşü ve CNN algoritmalarından faydalanılarak Politehnica Üniversitesi’nde üç araştırmacı tarafından gerçekleştirilen bir çalışmadır.[21] Çalışmanın temel amacı, otomatik ve gerçek zamanlı bir nesne tanıma sistemi oluşturarak görme engelli bireylere dış mekânlarda, özellikle hareket halindeki nesnelere takip edebilen yardımcı olacak bir donanım geliştirmektir. Bu sebeple, ani kamera/nesne hareketlerini, dönüşlerini veya farklı karmaşık değişiklikleri tolere edebilecek şekilde tasarlanmıştır. Bir video kamera, nVidia(GTX 1050) grafik karta sahip bir bilgisayar ve kulaklık içeren sistem,

nesne tanıma ve takip yaparken kullanıcıların önünde engel teşkil eden nesnelere var ise uyarı vermektedir. Nesne tanıma algoritması olarak gerçek zamanlı sistem tasarımına uygun olması sebebiyle YOLO modeli ve deneysel çalışmalarda veri seti olarak VOT 2016 [22] tercih edilmiştir. Çalışmanın sahip olduğu özellikler genel olarak şöyle sıralanabilir;

1. Nesnelerin görünüşleri ve belirgin hareketleri arasındaki ilişkiyi çevirim dışı bir şekilde öğrenebilen ilişki tabanlı, özgün bir algoritma
2. Nesne yok olması, ani kamera/nesne hareketlerini tolere edebilmesi
3. Nesne tanıma algoritması olarak YOLO modelini kullanması ve sınıfların içeriğinin görme engelli bireylerin ihtiyaçlarına göre genişletilmesi
4. Kullanıcı için tehlike teşkil eden engel niteliğindeki nesnelerin tanımlanarak akustik bildirimler verilmesi

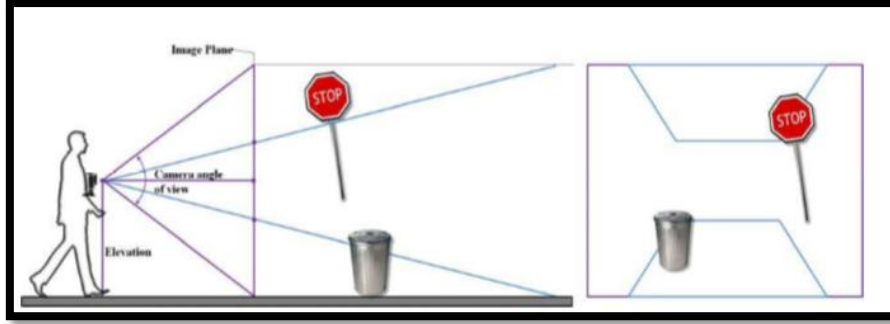
Önerilen sistemin, yukarıda bahsedilen fonksiyonları farklı modüller ile gerçekleştirilmektedir. Genel anlamda sistemin 4 farklı modüle sahip olduğu söylenebilir.

1. Nesne Tanıma Modülü: Bu modülde, gerçek zamanlı görüntü işleme yapabilen YOLO modeli nesne tanıma algoritması olarak kullanılmıştır. Aynı zamanda YOLO ile nesnenin görüntü içerisindeki konumu da belirlenebilmektedir.

2. Nesne Takip Modülü: Önerilen sistemde nesnelere takip etmek amacıyla farklı bir yaklaşım ortaya konmuştur. GOTURN[23] yaklaşımına benzer şekilde, ilişki tabanlı tekniği kullanarak nesnelerin görünüşleri ve hareketleri arasındaki ilişkiyi çevirim dışı olarak öğrenebilen bir yöntem kullanılmıştır. Sinir ağının eğitimi, gerçek dünyada hareket eden nesne örnekleri kullanılarak çevirim dışı bir şekilde gerçekleştirilmektedir. Bu yöntemde özgün nesnelere devamlı şekilde takip edilerek görünüşleri güncellenmekte yani kısaca "Tespit ile Takip" yapılmaktadır.

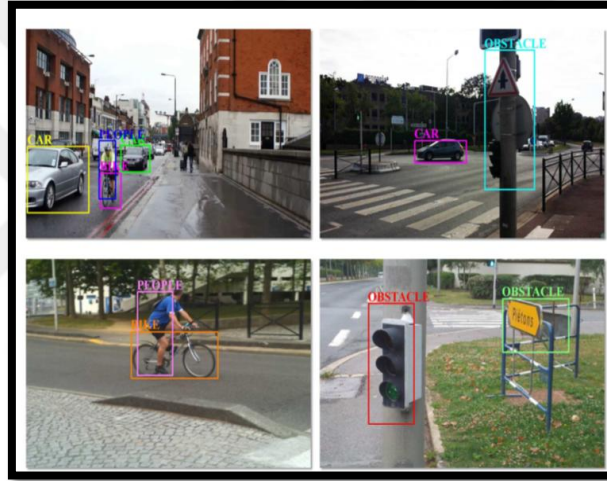
3. Engel Sınıflandırma Modülü: Engellerin sınıflandırılmasını sağlamak için YOLO modeli, yardımcı bir donanıma uygun eğitim sınıfları seçilerek genişletilmiştir. Örneğin; araba, bisiklet ve insanların yer aldığı sınıflar kullanılmaya devam ederken, engel teşkil eden çöp tenekesi, çit, merdiven, direk gibi nesnelerin yer aldığı alt sınıflar oluşturulmuştur.

4. Akustik Geri Bildirim Modülü: Nesnelere takip edilip sınıflandırdıktan sonra kullanıcıya olan yakınlıklarına göre tehlike dereceleri belirlenmektedir. Bunun için kullanıcının yürüme yolu ve baş hizasında ikizkenar yamuk şeklinde iki alan tehlike alanı olarak belirlenmiştir(Şekil 2.3.). Sınıflandırılan nesnelere, tehlike seviyelerine göre analiz edilir. Eğer bir nesne, yamuk şeklindeki alan içerisine düşüyorsa "U" yani önemli, düşmüyorsa "N" yani önemli olmayan nesne olarak kategorize edilir. Bu alanlarda yer alan nesnelere için de akustik uyarı mesajları kullanıcıya bildirilir. Kullanıcıya aynı anda çok fazla bildirim gönderip karışıklık yaratmamak adına, uyarılar belirli aralıklarla gönderilmektedir.



Şekil 2.3. Sistemin Çalışma Şekli [21]

Çalışmanın sonucunda, görme engelli bireyler için nesnelere tespit edip tanımlayan, takip eden ve engel teşkil eden durumlarda uyarı veren yardımcı bir donanım oluşturulmuştur. Şekil 2.4.'da deneysel çalışmanın sonucunda elde edilen sonuçlar gösterilmektedir.



Şekil 2.4. Test Sonuç Görüntüleri [21]

2.1.3. Görme Engelli Bireyler için Derin Öğrenme Tabanlı Bir Yaklaşım (Object Detection Featuring 3D Audio Localization for Microsoft HoloLens)

Alt başlığı "Görme Engelli Bireyler için Derin Öğrenme Tabanlı Bir Yaklaşım" olan çalışma Dortmund Uygulamalı Bilimler ve Sanatlar Üniversitesi'nde üç araştırmacı tarafından gerçekleştirilmiştir. Çalışmanın amacı; görme engelli bireylere nesne tanıma yaparak sesli bildirimler verebilecek giyilebilir bir donanım oluşturmaktır. Prototipi oluşturulan sistemin yazılım tasarımı, istemci-sunucu iletişimine dayanmaktadır[24].

- Sunucu tarafında; grafik işleme birimi olarak NVIDIA Titan X ve nesne tanıma algoritması olarak YOLOv2 modeli kullanılmıştır.
- İstemci tarafında; HoloToolkit [25], Unity 3D ve Visual Studio IDE'ye entegre bir biçimde çalışmaktadır. HoloToolKit, kameraları, ses girisini, 3B ses çıktısını ve sunucu iletişimini

kontrol etmektedir. Hareket ve ses dışında, HoloLens “Clicker” adında ufak boyutta bir buton sayesinde Bluetooth bağlantısıyla da kontrol edilebilmektedir. Sunucu tarafında nesne tespiti yapılırken öncelikle, entegre kamera JPG formatında, 896x504 boyutlarında RGB görüntüyü aldıktan sonra analiz edilmek üzere sunucuya gönderir. Görüntü sunucuya ulaştığında, YOLOv2 ile analiz edilir ve sonuçlar JavaScript Object Notation(JSON) formatına dönüştürülerek tekrar istemciye gönderilir. Sunucu, 2B koordinatları HoloLens’e gönderdikten sonra gelen 2B veriye ve HoloLens tarafından oluşturulan sanal 3B modele göre nesnelerin gerçek 3B koordinatları hesaplanır. Sanal 3B model haritası, HoloLens’in Time-of-Flight(TOF) sistemi ve gizli kameralarıyla oluşturulabilmektedir. Tasarlanan donanımın kullanırken, öncelikle kullanıcının sesli bir komut vermesi veya “Clicker” butonuna tıklaması gerekmektedir. Eğer, istenilen nesnelere görüntü içerisinde bulunuyorsa sistem bu nesnelere sesli bildirimlerle kullanıcıya iletmektedir. Sesli komutların algılanması için, Microsoft’un uluslararası Windows platformunun (UWP) parçası olan Speech Recognition platformu kullanılmıştır. Kullanılan, UWP fonksiyonları Unity3D ile de kullanılabilir. Kullanıcı, istemci ve sunucu etkileşimi tablo 2.1.’de verilmiştir.

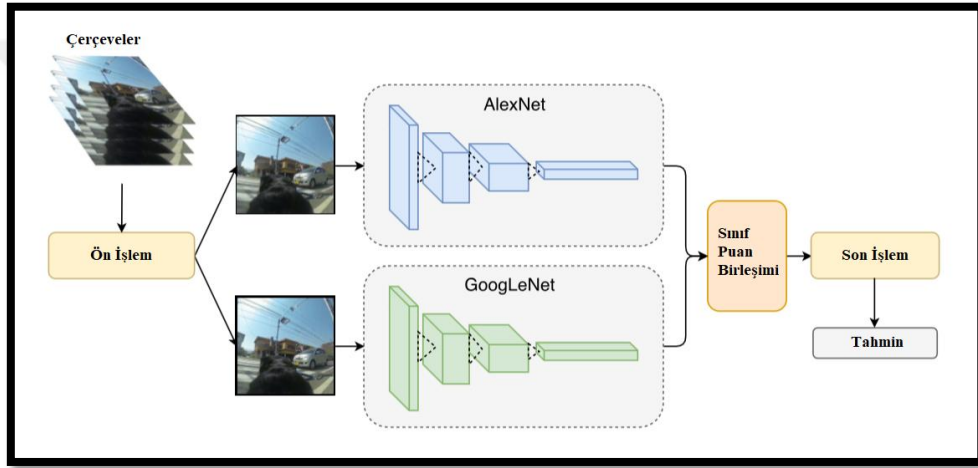
Çalışmanın sonucunda, sesli komutların %83’ü doğru algılanmıştır. YOLOv2 denenerek VOC 2007 veri setinde, %65,5 doğrulukla çalışmıştır. Geri kalan %34,5 hatanın, %19,0’i konumlandırma, %6,75’i benzer nesne sınıf tespiti, %4,75’i arka plan tespit hatası ve %4’ü de yanlış sınıflandırma olarak belirtilmiştir.

Tablo 2.1. Kullanıcı, İstemci ve Sunucu Etkileşimi

Kullanıcı	İstemci	Sunucu
“scan”	Görüntüyü kaydeder. Sunucuya gönderir. Nesne listesini sesli olarak okur.	Görüntüyü alır. Nesne tespiti/tanıma yapar. Tanınan nesnelerin listesini istemciye gönderir.
“objects”	Nesne listesini sesli olarak okur.	
“cup”	Bardağı pembe küple işaretler.	
“distance”	Seçili olan nesneye olan uzaklığı söyler.	
“all”	Tüm nesnelere seçer veya seçili olan nesnelere bırakır.	

2.1.4. Sanal Rehber Köpek (Virtual Guide Dog)

Sanal Rehber Köpek adlı çalışma Pontifical Katolik Üniversitesi'nde beş araştırmacı tarafından gerçekleştirilmiştir. Çalışmanın amacı, bir rehber köpek üzerine yerleştirilen kameradan alınan görüntüler ile oluşturulan veri seti ile konvolüsyonel yapay sinir ağını eğiterek, görme engelli bireylere etrafta gerçekleşen olaylar hakkında geri bildirim sağlamayan bir sistem oluşturmaktır. Farklı CNN mimarilerinin her birinin kendi içerisinde farklı özellikleri olduğunu belirten araştırmacılar, iki farklı CNN mimarisini aynı anda paralel olarak kullanacakları bir yöntem seçmişlerdir. GoogLeNet [26] ve AlexNet [27] ağlarını kullanarak tasarladıkları sistemin çalışma şekli Şekil 2.5.'da gösterilmiştir[28].



Şekil 2.5. Hareket Tanıma Sisteminin Çalışma Şekli [28]

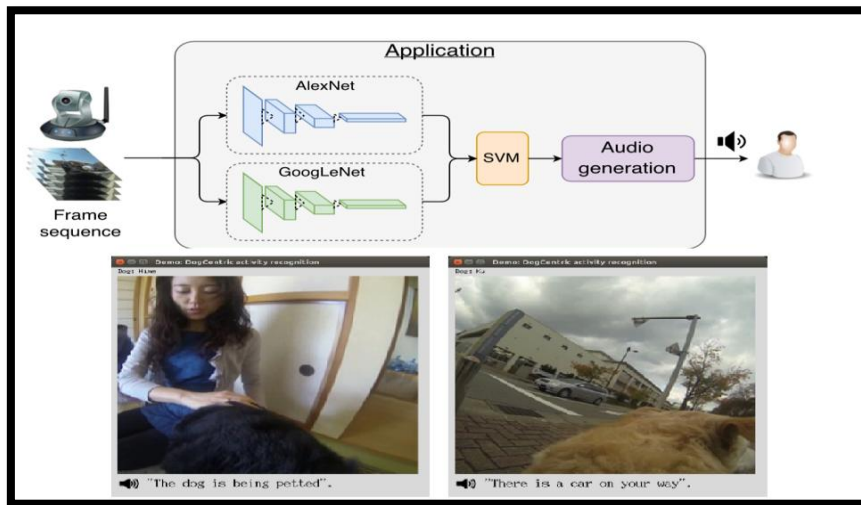
İlk aşaması "Ön İşlem" olarak adlandırılan sistemin bu aşamasında, görüntüler 256x256 şeklinde yeniden boyutlandırılmaktadır. Yeniden boyutlandırma bazı özelliklerin azaltılmasını sağlamakta ve işlem süresini kısaltmaktadır. İkinci aşamada sistem, aynı anda iki farklı CNN ağı paralel olarak çalıştırmaktadır. AlexNet'in sistemde kullanılan versiyonu; 5 konvolüsyonel katman içeren 8 ağırlık katmanı, 3 tam bağlantılı katman(fully-connected layer) ve 3 max-pooling katmanı içermektedir. GoogLeNet de 22 katmanlı derin bir ağ mimarisidir. Sistemde görüntü serileri bu iki ağ içerisindeki birden fazla konvolüsyonel, pooling ve tam bağlı katman içerisinde geçerek softmax katmanında sonlanırlar. Softmax katmanı da her sınıf için olasılıkları oluşturur. Sistemin üçüncü aşaması olan "Sınıf Puan Birleşimi" adlı aşamada her iki CNN ağının çıktılarından ortak bir vektör oluşturulur. Daha sonra "Son İşlem" aşamasında tahmin edilen sınıflardaki gürültü azaltılır ve tahmin edilen sınıflar SVM(Destek Vektör Makineleri) kullanılarak düzeltilir. Sistemi oluştururken "DogCentric Activity" [29] isimli veri seti kullanılmıştır. Bu veri seti, köpeklerin sırtına yerleştirilen kameralar ile dış ortamdan alınan videolardan(saniyede 48 çerçeve) oluşmaktadır. Veri seti, 4 köpeğin 10 farklı hareketini(koşma,

insanlarla etkileşim, arabanın geçmesini bekleme gibi) içeren 209 videoya bölünmüştür. Bu 10 farklı hareket şunları içermektedir;

- “Arabanın geçmesini bekleme” = “Araba” (4.920 çerçeve)
- “Su içme” = “İçmek” (3.300 çerçeve)
- “Bir şeyler yeme”= “Beslenme” (3.795 çerçeve)
- “Köpeğin kafasını sola çevirmesi” = “Sol” (1.950 çerçeve)
- “Köpeğin kafasını sağa çevirmesi” = “Sağ” (1.380 çerçeve)
- “Biri tarafından sevilmesi” = “Okşama” (3.740 çerçeve)
- “Topla oynaması” = “Oyna” (3.545 çerçeve)
- “Vücudunu sallaması” = “Salla” (1.880 çerçeve)
- “ Bir şeyleri koklama” = “Kokla” (4.960 çerçeve)
- “Yürümesi” = “Yürü” (4.175 çerçeve)

Sistem oluşturulurken DogCentric Activity veri setinde gereksiz olarak görülebilecek olan hareketler(“köpek sola bakıyor” gibi) veri setinden çıkarılmıştır. Sistemde yer alan sınıflar aşağıda kalın harflerle belirtilmiş ve kullanıcılara verilecek olan sesli bildirimler yanlarına yazılmıştır.

- **Araba:** Yolunuzda araba var.
- **İçmek:** Köpek bir şey içiyor.
- **Beslenmek:** Köpek bir şey yiyor.
- **Sevilmek:** Köpek biri tarafından okşanıyor.
- **Koklamak:** Köpek bir şeyi kokluyor. Sistem mimarisi ve prototip görüntüleri Şekil 2.6.'da verilmiştir. Sonuç olarak sistemin, gerekli sınıflandırmaları yaparak görme engelli kullanıcılara sesli bildirim vermesi sağlanmıştır.



Şekil 2.6. Sistem Mimarisi ve Prototip Görüntüleri[28]

2.2. Uzaklık Tahmini ve Nesne Tespiti için Kullanılan Diğer Yöntemler/Sistemler

Bu bölümde görme engelli bireylere asistanlık etmek amacıyla geliştirilen genel olarak “Electronic Travel Aid (ETA)” yani “Elektronik Gezi Yardımcısı” olarak adlandırılan sistemler incelenmiştir. Sensör ve video tabanlı olmak üzere geliştirilen bu sistemlerde nesne tespiti ve uzaklık tahmini yapabilmek için kullanılan yöntemler açıklanmıştır.

2.2.1. Sensör Tabanlı Sistemler

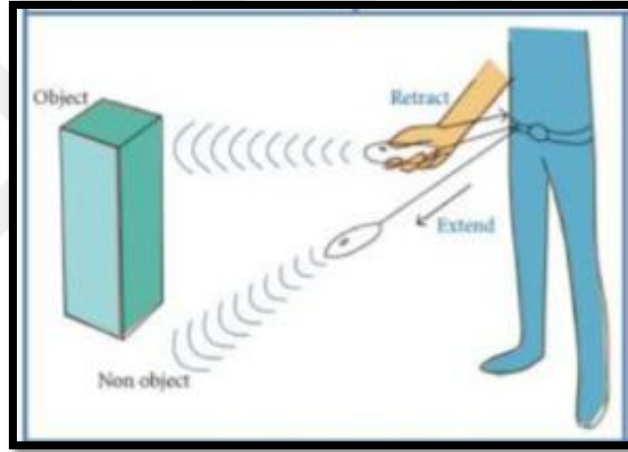
Sensör ağıyla oluşturulan ETA’lar çevreden toplanan bilgileri işitsel veya dokunsal sinyallere dönüştürerek görme engelli bireylere aktarırlar. Sensör tabanlı yardımcı cihazları; ultrasound[30,31], infrared[32], küresel konum belirleme sistemi(GPS) [33,34], radyo frekans tanımlayıcı(RFID) [33, 35, 36] kullanarak konum belirleme ve nesne tespiti yapabilmektedir. sensör tabanlı sistemlerden bir diğeri Mowat sensörleriyle oluşturulan sistemlerdir. Mowat sensörü görme engelli birey etrafındaki çeşitli engellere olan uzaklığını tahmin edebilmektedir. Bu sistemlerin, görme engelli bireyler tarafından rahatlıkla kullanılabilmesi için taşınabilir olması ve geniş eğitim sürecine ihtiyaçları vardır. Bunların dışında literatürde yer alan sonar tabanlı iki resimleme sensörü Sonicguide ve Trisensor, kullanıcıya tespit edilen engellerin konum bilgilerini de sunmaktadır [30]. Fakat, bu iki prototip de düşük çözünürlüklü sensörler kullanmakta ve tahmin edilen uzaklıkların doğruluk oranları düşüktür.

Yakın zamanda, “Konuşan Braille” sistemi görme engelli bireylere kamusal kapalı alanlarda yardımcı olmak için tasarlanmıştır[32]. Sistem bilinen kablosuz ağlar içerisinde çalışmaktadır. Bu sebeple, sistemin bağlantı noktaları önceden belirlenmiş olan binalarla sınırlıdır. GuideCane[37], SmartCane[38], UltraCane[39], Batcane[40] ve Necklace cane[41], ultrasonik sensör ağlarını kullanarak görme engelli bireylerin genellikle kullandıkları beyaz bastonun özelliklerini geliştirmeyi amaçlamaktadır. Bu çatı altında geliştirilen sistemler, engelsiz ve en uygun yürüyüş yolunu tahmin ederek görme engelli bireyleri uyarmayı amaçlamaktadır. Fakat gerçek hayat senaryolarına uyarlandıkları zaman kullanıcının omuz hizasından üstteki nesnelere, üstten sarkan engelleri ve kaldırım sınırlarını tespit etmekte güçlük çekmektedir. Bahsedilen sistemlerin görüntüleri Şekil 2.7.’da gösterilmektedir.



Şekil 2.7. Guidance [37], SmartCane[38], Batcane[40], Necklace sonar[41]

Dokunsal duyu ile çalışan bir cihaz olan CyARM[42] ultrasonik sensör kullanarak görme engelli bireyleri nesnelere fark etmesine yardımcı olmaktadır. Gerçek yaşam ortamlarında denendiğinde sabit nesnelere fark etme oranının yüksek, hareketli nesnelere fark etme oranının ise %30 daha az olduğu görülmüştür. Aynı zamanda, ultrasonik sensörler hava koşullarından da etkilenmektedir. Şekil 2.8. 'da sistemin çalışma şekli basit olarak gösterilmiştir.



Şekil 2.8. CyARM[42]

Düşük fiyatlı, mikro denetleyici ile ultrasonik sensör kullanılarak tasarlanmış olan bir diğer rehber sistem "görme engelli bireyler için yapay rehber"[31] olarak adlandırılan bir çalışmadır. Bu sistemin amacı, kullanıcıların ellerindeki baston üzerine yerleştirilen ultrasonik sensör sayesinde bastonun önüne gelen nesnelere fark etmelerini sağlamaktır.

Bir diğer çalışma, "Smart Robot" (Şekil 2.9.) üzerinde yer alan RFID ve GPS modülleri sayesinde görme engelli bireylerin önceden belirlemiş mesafelere ulaşmalarını sağlamaktadır. Bu cihaz kullanıcılara sesli ve titreşimle geri bildirim vermektedir. Bu tarz GPS tabanlı geliştirilmiş sistemlerin sayısı fazla olsa da, kalabalık ve yüksek bina yapılarının olduğu şehirlerde sinyal kaybı nedeniyle sistemlerin çalışma doğruluğu oldukça düşmektedir[43].



Şekil 2.9. Smart-Robot[43]

Kapalı alanlarda görme engelli bireyleri yönlendirmek için tercih edilen teknolojilerden biri de RFID' dir[44, 45]. RFID teknolojisi kullanılarak geliştirilen, kullanıcıların belirlenmiş varış noktalarına ulaşmalarını sağlayan sistem Şekil 2.10.' da gösterilmiştir. Bu sisteme benzer olan bir diğer sistemde, RFID modülü ürünlerin yerini tespit edebilmek adına QR(Quick Response) kod ile birlikte kullanılmıştır[45]. RFID'li sistemlerin temel sınırlılığı, kullanılacakları bina yapılarının önceden bilinmesinin gerekmesidir. Çünkü çok geniş alanlarda çalışma kapasiteleri bulunmamaktadır.



Şekil 2.10. RoboCart[44]

Tek başına bir alışveriş için, bağımsızlık ve güvenilirlik görme engelli bireyler için büyük önem taşımaktadır. Fakat onlar için alışveriş merkezleri, marketler ve/veya mağazalar engel teşkil eden ortamlardır[46]. Bu problemi çözmek adına geliştirilmiş olan bir diğer sistem “ShopTalk” alışveriş asistanı olarak adlandırılmıştır[47]. Şekil 2.11.’da gösterildiği gibi sistem; sıradan bir başlık, barkot okuyucu, nümerik mini klavye ve bir işlem ünitesinden oluşmaktadır. Deneysel sonuçlara göre, ShopTalk, ürün alma konusunda çok başarılı olmuştur. Fakat sistemin taşınması zor ve marketlerin stok bilgisine ulaşma gereksiniminden dolayı sınırlılıkları olduğu da söylenmelidir.



Şekil 2.11. ShopTalk [47]

2.2.2 Video Tabanlı Sistemler

Donanımsal işleme cihazları, bilgisayar görüşü algoritmaları ve makine öğrenmesi teknolojilerinin yaygınlaşmasıyla birlikte görme engelli bireyler için çok sayıda yapay zekâ tabanlı yardımcı sistem geliştirilmiştir. Kamera sensörüne bağlı olarak, video tabanlı metotlar tek(monocular), stereo ve RGB-D olmak üzere üç bölüme ayrılmaktadır.

2.2.2.1 Tek(Monocular) Kamera Tabanlı Sistemler

“Arianna” olarak adlandırılan sistem, düşük fiyatlı olup artırılmış gerçeklik (Augmented reality-AR) tabanlı olarak tasarlanmıştır. Sistemin amacı, görme yetisini tamamen veya kısmen kaybeden bireyler için kapalı alanlarda güvenli yürüme yolu belirlemektir[48]. Donanım tarafında akıllı bir telefona entegre edilmiş bir video kamera yer almaktadır. Yürünecek yol QR

kodlar veya zemin rengi sayesinde belirlenerek kullanıcıya bildirimler titreşim yoluyla iletilmektedir(Şekil 2.12.).



Şekil 2.12. Arianna [48]

Bir diğer tek kamera tabanlı sistem "Mobile Vision" olarak adlandırılmıştır[49]. İç mekânlarda renk işaretlerini kullanarak belirleme yapan sistem akıllı telefona entegre olarak çalışmaktadır. Kırmızı, yeşil ve mavi renk işaretleri kullanılarak kullanıcılar dinlenme odası, asansör veya çıkış kapısı gibi belirli bölgelere yönlendirilmektedir. Kullanıcılara geri bildirimler metinden konuşmaya dönüştürme(Text-to-Speech) teknikleri kullanılarak oluşturulmuştur. Yukarıda bahsedilen Arianna ve Mobile Vision sistemlerinin ortak dezavantajı, her ikisinin de kullanılacağı mekânın önceden tespit edilerek hazırlanması gerekmektedir. Bu sebeple, her iki sistem de önceden tespit edilmeyen ortamlarda kullanılamamaktadır. Bir engel tespit ve sınıflandırma metodunu akıllı telefona entegre ederek tasarlanan sistemin[50] Şekil 2.13.'da bir üst versiyonu açıklanmıştır [51].



Şekil 2.13. Monacu ve Ark. [51]

Bu sistem görme engelli bireyleri hem iç hem de dış ortamlarda yönlendirmek için tasarlanmıştır. Geliştiricilerin amacı; ardı ardına gelen çerçeveler içerisinde ilgili noktaları geleneksel Lucas-Kanade [52] algoritmasını kullanarak takip edip engellerin konumunu belirlemektir. Nesne hareketlerini kamera hareketlerinden ayırt etmek için RANSAC[53] algoritması ile sınıflandırılmış olan bir set homografik dönüşümden faydalanılmıştır. Saptanan nesnelere HOG tanımlayıcısı[54] ile kategorize edilmektedir. Sistem iyi sonuçlar verse de geniş kat yapılarında nesne tanıma yapamamakta ve görme engelli kullanıcıyla engel arasındaki uzaklığı doğru bir şekilde tahmin edememektedir. Sistemin bir üst sürümünde, geliştiriciler yukarıda bahsedilen sınırlılıkları çözmek adına sisteme ultrasonik sensör entegre etmişlerdir. Bu yöntem ümit verici sonuçlar vermektedir.

Tian ve arkadaşları tarafından geliştirilen bilgisayar görüşü temelli yol bulma sistemi Şekil 4’de gösterilmiştir. Sistem kullanıcıların, bilindik olmayan kapalı ortamlarda bağımsız hareketine yardımcı olmak amacıyla tasarlanmıştır[55]. Donanım tarafında, bir video kamera, bir mikrofon, bir bilgisayar ve Bluetooth kulaklıklar kullanılmıştır. Sistem köşe ve kenar belirleme algoritmalarını kullanarak kapı, asansör ve kabin gibi nesnelere geometrik özelliklerinden faydalanarak tanıyabilmektedir. Daha sonra, optik karakter tanıma metodu kullanılarak, sistemin ön ve arka plandaki nesnelere tanıması sağlanmıştır. Kıyafet tanıma modülü ek bir modül olarak sisteme eklenmiştir[56]. Radon işaret tanımlayıcı kullanılarak dört farklı kıyafet dokusu algılanabilmektedir. Bunlar; kareli kumaş, çizgili kumaş, desensiz kumaş ve düzensiz kumaşlardır. Sistem gerçek zamanlı çalışamaması ve nesne kapanmalarını fark edememesi açısından dezavantajlı olarak görülmektedir.

“Crosswatch” isimli sistem bilgisayar görüşü algoritmaları ve coğrafi bilgi sistemi (GIS) kullanılarak geliştirilmiştir [57]. Crosswatch görme engelli kullanıcıların kavşaklardan karşıdan karşıya geçerken kullanabilecekleri, akıllı telefon tabanlı bir yön bulma asistanıdır. Crosswatch sesli uyarı mesajlarını kullanıcıya iletmek için telefon hoparlörünü kullanmaktadır. Bu sebeple, kabalık şehir ortamlarında sesli uyarıları duymak zor olabileceği düşünülmektedir. Crosswatch’a benzer bir diğer sistem “TL-recognizer” adıyla iki farklı sürümde geliştirilmiştir[53,54]. Renk filtreleri ve kontur tanıma metotları kullanılarak geliştirilen sistemin amacı trafik ışıklarını tanımadır. Dezavantaj olarak; sistemin kamera hareketlerinden etkilendiği görülmüştür.

Alışveriş asistanı olarak geliştirilen “ShopMobile”[60], akıllı telefona entegre edilmiş ve “ShopTalk”[47] sisteminin sınırlılıklarını azaltmayı amaçlayan bir sistemdir. Akıllı telefon kamerası ürün barkotlarını konumlandırmak amacıyla kullanılmıştır.

Görme engelli kullanıcıların yönlendirilmesi amacıyla geliştirilen bir diğer sistem kapalı alanlarda kullanılacak şekilde tasarlanmıştır[61]. Bir akıllı telefon üzerine gömülü kamera ile kaydedilen videolar işlenmesi için bir grafik işleme ünitesi(GPU) sunucusuna

gönderilmektedir. Sistem umut vadeden sonuçlar verse de sadece sınırlı sayıda kapalı ortamda denenmiştir. Geniş çaplı, farklı bina yapılarını içeren bir veri seti kullanılması bu açıdan uygun olabilir. Aynı zamanda işlem hızını arttırmak adına hiyerarşik ve içerik temelli metotlar kullanılabilir.

Tek kamera (monocular) kullanılarak geliştirilen, farklı donanımlar içerisine gömülü video kameraları kullanarak görme engelli kullanıcılar için gerçek zamanlı banknot tanıma amacıyla tasarlanmış farklı birçok uygulama bulunmaktadır [62],[63,64], [65],(Şekil 2.14.).



Şekil 2.14. Manikandan ve Ark.[64], Springer ve Ark.[65]

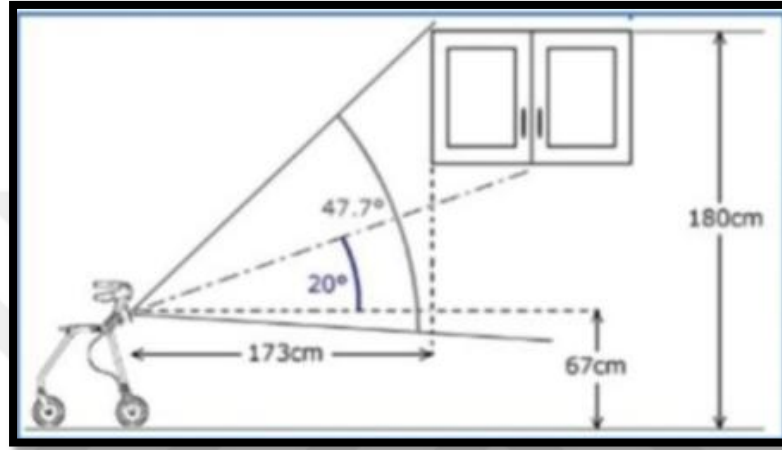
Bu tarz uygulamaların gerekliliği çok açık olmasına rağmen, bu uygulamalar hassasiyet, ortamın aydınlığı ve tıkanma gibi sınırlılıklara sahiptir. Örneğin, uygulamanın banknotu tanıyabilmesi için banknotun yüzünün kameraya dönük olması gerekir. Bunlara ek olarak düşük tanıma oranı ve doğruluk skorları da dezavantaj olarak görülmektedir. Görme engelli bireylerin tempolu ve yavaş koşu gibi yarışlarda kullanabilecekleri diğer bir asistan mekatronik bir sistem olarak geliştirilmiştir[66]. Donanım tarafında; bir kamera, işlem ünitesi ve bir adet dokunmatik cihaz bulunmaktadır(Şekil 2.15.). Yazılım tarafında; geliştiricilerin amacı yerdeki şeritleri ve koşu koridorunu tespit etmektir. Sistemin doğruluğu büyük oranda ortamın ışık durumuna bağlı olarak değişkenlik gösterebilmektedir. Buna ek olarak sistemin şarj ömrünün bir saat olduğu belirtilmiştir.



Şekil 2.15. Prototipin Görünümü[66]

2.2.2.2 Stereo Kamera Tabanlı Sistemler

“Smart Walker” adlı sistem dış ve iç ortamlarda, tehlikeli engelleri ve riskli yüzey değişimlerini tespit etmeyi amaçlamaktadır[67](Şekil 2.16.). “Smart Walker” aralıklı 3B bilgilerini kullanan gerçek zamanlı stereo engel tespit algoritması temellidir. Belirlenmiş 3B noktalarla çalışması sayesinde 3B harita çıkarımına göre daha avantajlıdır. Sistem saniyede 4 çerçeve işleyebilmekte ve tanıma oranı da %65’ten düşüktür.



Şekil 2.16. Smart Walker [67]

Diğer bir giyilebilir, kolay taşınabilen, dinamik görüş sensörleri ile oluşturulan yönlendirme cihazı Şekil 2.17.'da gösterilmektedir. Sistemin test edilmesinden sonra elde edilen rapordaki sonuçlara göre; sistemin baş hizasının altında kalan nesnelere algılamakta zorlandığı ve kulaklık sebebiyle kullanıcıların etraftaki diğer sesleri duymakta zorlandığı gözlemlenmiştir[68].



Şekil 2.17. Everding ve Ark. [68]

2.2.2.3. RGB-D Kamera Tabanlı Sistemler

Literatürde, RGB-D sensörüne sahip kameraları kullanarak geliştirilmiş olan birçok sistem yer almaktadır. RGB-D sensör RGB verileri üzerinden derinliği de algılayabildiği için yardımcı cihazlarda çok tercih edilmektedir. Görme engelli ve az gören bireyler için geliştirilmiş olan gerçek zamanlı bir yüz tanıma sistemi Şekil 2.18.'da gösterilmektedir. Sistem giyilebilir Kinect sensörler kullanılarak yüz tanıma yapmaktadır. Tanınan yüzler sesli olarak kullanıcılara iletilmektedir. Sistemde kullanılan bilgisayar görüşü algoritması gerekli olan işlemsel (hafıza, işlem gücü ve batarya ömrü) yükü azaltmaktadır. Sistemin tek dezavantajı Kinect sensöre sahip olduğu için sadece kapalı alanlarda verimli çalışabiliyor olmasıdır[69].



Şekil 2.18. Neto ve Ark[69]

Görme engelli bireyler için geliştirilmiş olan bir diğer kapalı alan yönlendirme sistemi "INSANA" olarak adlandırılmıştır. Engel tespiti yapabilen sistem Tango cihazına entegre edilmiştir. Şekil 2.19.'da ISANA'nın görüntüsü gösterilmektedir. Nesnenin konumu derinlik sensörü ile belirlenmektedir. Yönlendirme mesajları sesli olarak kullanıcılara iletilmektedir[70].



Şekil 2.19. ISANA [70]

3. MATERYAL ve YÖNTEM

Bu bölümde tez çalışmasının tasarımı ve uygulanmasında kullanılan materyal ve yöntemler ayrıntılı olarak incelenmiştir. Çalışmada, görme engelli bireyler için nesne tanıma ve uzaklık tespiti yapabilen yardımcı bir sistem oluşturulması amaçlanmıştır. Uzaklık tespiti modülünü geliştirmek için öncelikle 1 ve 10 metre arasında görüntüler alınmıştır. Özgün bir yaklaşım ile görüntülerden çıkarılan vücut boyu, genişliği ve yüz boyu, genişliği özelliklerine ait veriler 7 farklı regresyon modeli ile eğitilerek test sonuçları analiz edilmiştir. Test sonuçlarının anlamlı ve yüksek güvenilirlikte çıkması sonucunda uzaklık tespiti ve nesne tanıma modülünün Raspberry Pi[71] üzerinde gerçekleştirilerek prototipin oluşturulması aşamasına geçilmiştir. Sistem prototipinin istikrarlı ve doğruluk oranı yüksek olması adına Raspberry Pi kamera modülü[72] kullanılarak 2 ile 10 metre arasında görüntüler alınmış ve ikinci bir veri seti oluşturulmuştur. Aynı şekilde veriler 7 farklı regresyon modeli ile eğitilerek test edilmiş ve başarılı sonuçlar elde edilmiştir. Nesne tanıma modülü için konvolüsyonel yapay sinir ağları temelinde geliştirilen, gerçek zamanlı bir nesne tanıma sistemi olan YOLO(You Only Look Once)[20] NNPack Darknet Kütüphanesi[73] ile birlikte kullanılmıştır. Nesne tanıma ve mesafe tespiti sonuçları geçici olarak metin halinde depolanmış ve Google tarafından geliştirilmiş olan Text-to-Speech(gTTS)[74] Kütüphanesi kullanılarak sonuç metni gerçek zamanlı olarak sesli bildirimle dönüştürülmüştür. Sistemin akış şeması Şekil 3.1.'da verilmiştir.

Bu bölümün devamında materyal ve yöntemler aşağıdaki başlıklar altında açıklanmıştır.

I. Ham Veri Setinin Oluşturulması

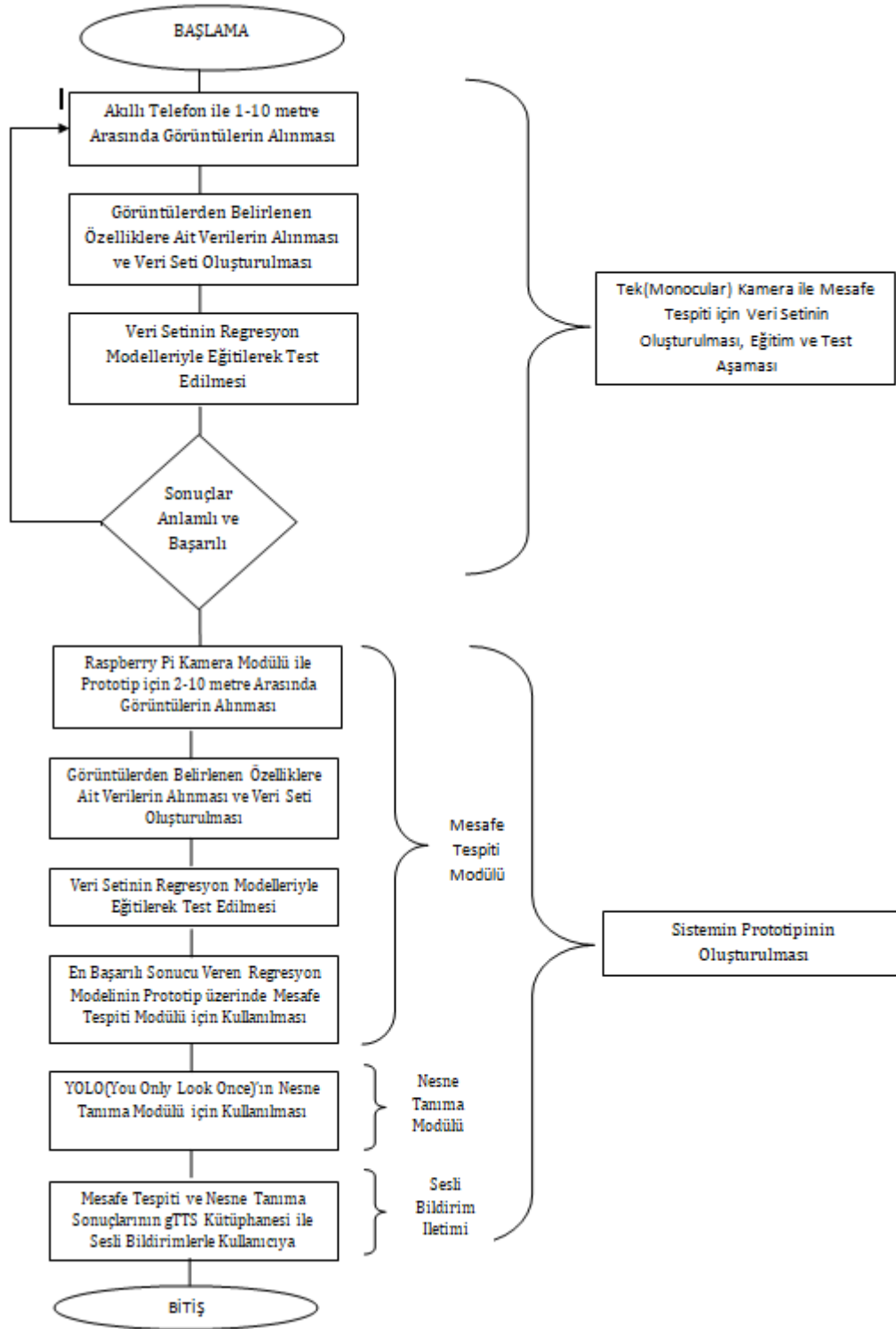
II. YOLO'nun Çalışma Şekli ve NNPack Darknet Kütüphanesi

- Bilgisayar Görüşü ve Konvolüsyonel Yapay Sinir Ağları
- YOLO'nun Çalışma Şekli
- NNPack Darknet Kütüphanesi

III. Regresyon

- Karar Ağacı
- Adaboost Karar Ağacı
- K En Yakın Komşu (KNN)
- Adaboost KNN
- Ensemble KNN
- Rassal Orman
- MLP Regresör

IV. Google Text to Speech Modülü ile Sesli Bildirim Oluşturulması



Şekil 3.1. Sistemin Akış Şeması

3.1. Ham Veri Setinin Oluşturulması

Mesafe tespitini tek(monocular) kamera kullanarak yapmak sensör veya RGB-D kamera kullanarak yapmaya göre daha az maliyetli ve kullanışlı olsa da daha zor bir yöntemdir. Tek bir resim veya video üzerinden mesafe tespiti yapabilmek adına kişilerin boyutlarının mesafeye

göre değişimini incelemek anlamlı sonuçlar veren bir yaklaşım olmuştur. Çünkü kişilerin ve diğer tüm nesnelerin boyutları(boy, genişlik) kameradan uzaklaştıkça azalmakta ve yakınlaştıkça da artmaktadır. Bu doğrultuda, mesafe tespiti modülü için bir akıllı telefon kamerası kullanılarak 1 ve 10 metre arasında, 480x800 boyutlarında görüntüler alınmış ve Excel dosyasına kaydedilmiştir. Bu görüntülerden kişilerin her uzaklık için vücut boyu, genişliği ve yüz boyu, genişliği verileri alınarak veri setine kaydedilmiştir. Veri setinde yer alan örnek bir kişinin verisi tablo 3.1.'da gösterilmektedir. Burada "name" kişinin numarasını, "bh" kişinin vücut uzunluğunu, "bw" kişinin vücut genişliğini, "fh" kişinin yüz uzunluğunu, "fw" kişinin yüz genişliğini ve "distance" uzaklığı temsil etmektedir.

Tablo 3.1. Veri Setinden Örnek Bir Veri Kümesi

Name	bh	bw	fh	fw	distance
p_1	676	324	132	102	1
p_1	475	150	60	46	2
p_1	318	103	39	31	3
p_1	242	75	30	24	4
p_1	189	61	22	18	5
p_1	162	54	19	15	6
p_1	136	44	15	13	7
p_1	121	40	13	11	8
p_1	108	32	12	9	9
p_1	96	30	11	8	10

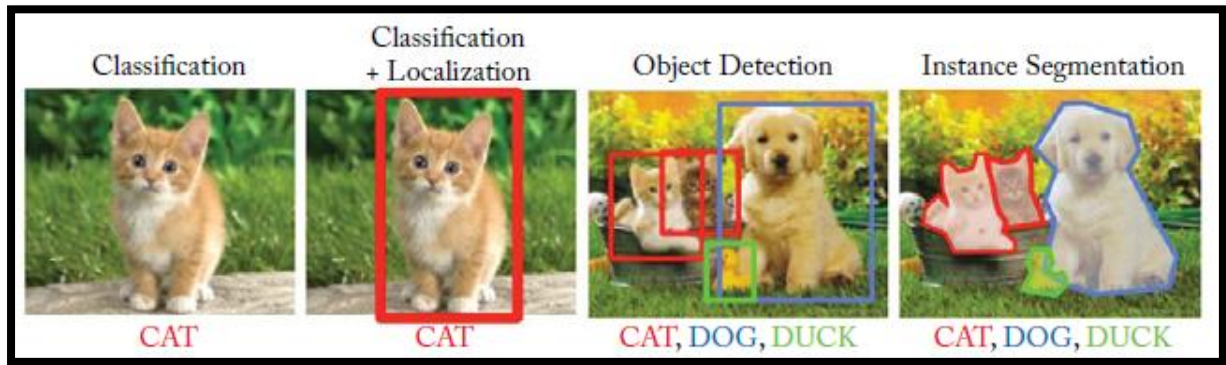
Oluşturulan veri seti Karar Ağacı, Adaboost Karar Ağacı, K En Yakın Komşu, Adaboost K En Yakın Komşu, Topluluk K En Yakın Komşu, Rassal Orman, Çok Katmanlı Algılayıcı (MLP) regresyon modelleri kullanılarak eğitilmiş ve test edilmiştir. Python programlama dili ile modellerin algoritmaları kodlanmış ve geliştirme ortamı olarak IDLE Python 3.7 kullanılmıştır. Her modelin uygulama sonuçları Bulgular ve Tartışma bölümünde farklı parametreler üzerinden ayrıntılı olarak anlatılmıştır.

3.2. YOLO'nun Çalışma Şekli ve NNPACK Darknet Kütüphanesi

Bu bölümde sistemin nesne tanıma modülü için kullanılan konvolüsyonel yapay sinir ağları temelinde geliştirilmiş olan gerçek zamanlı nesne tanıma sistemi olan YOLO(You Only Look Once) ve sinir ağlarının ARM CPU'larda çalışması için bir hızlandırma kütüphanesi olan NNPACK Darknet hakkında bilgiler verilmiştir.

3.2.1. Bilgisayar Görüşü ve Konvolüsyonel Yapay Sinir Ağları

İnsanlar etraflarındaki 3 boyutlu dünyayı görmek ve anlamak için gözlerini ve beyinlerini kullanırlar. Örneğin, Şekil 3.2.'da gösterildiği gibi bir görüntü verildiğinde, insanlar görüntüdeki bir “kedi”yi kolaylıkla görebilirler. Böylece görüntüyü sınıflandırır (sınıflandırma görevi); görüntüdeki kediye konumlandırır (sınıflandırma artı lokalizasyon görevi); resimde mevcut olan tüm nesnelere konumlandırır ve etiketler (nesne algılama görevi); ve mevcut olan bireysel nesnelere bölümlere ayırır (örnek segmentasyon görevi). Bilgisayarlı görme bilimi, bilgisayarlara daha iyi olmasa da benzer bir yetenek vermeyi amaçlamaktadır. Daha doğrusu bilgisayarlı görme, en şaşırtıcı yeteneklerden birini çoğaltabilen yöntemleri geliştirmeyi amaçlamaktadır. [75]



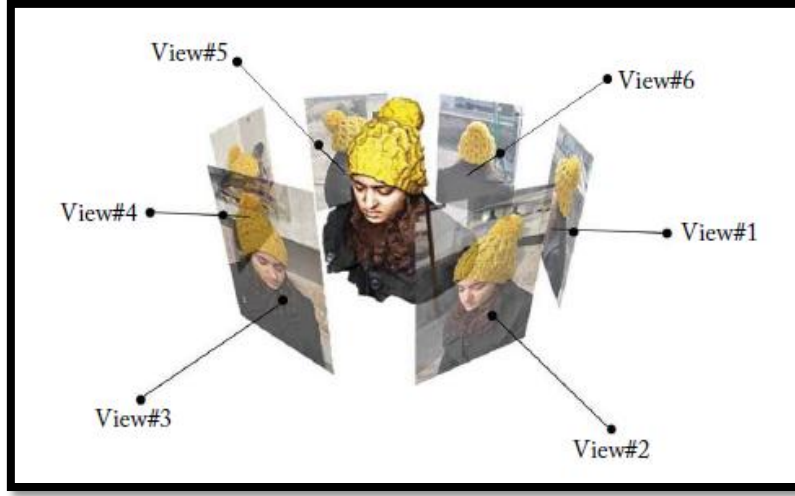
Şekil 3.2. Nesne Tanıma için Örnek Görüntü[75]

Bilgisayarların görüntü verileriyle;

- Görüntüye bakmak ve sınıflandırma yapması
- Sınıflandırma artı lokalizasyonu gerçekleştirilmesi (örneğin, etrafında bir sınırlayıcı kutu bulmak için)
- Görüntüde mevcut olan tüm nesnelere konumlandırılması
- Nesnelere etiketlemek veya semantik örnek segmentasyonunu gerçekleştirilmesi, yani aynı türden olsalar bile, bir sahnede tek tek nesnelere segmentasyonunu yapması mümkündür.

Ancak, dünyanın üç boyutlu yapısını iki boyutludan geri kazanmak ve anlamak kameralar tarafından çekilen görüntüler için zorlu bir iştir. Bilgisayar ortamında araştırmacılar, görüntülerden üç boyutlu şekil ve görünüşü geri kazanmak için matematiksel teknikler geliştirmektedir. Örneğin, yeterince geniş bir resim grubu ve çeşitli görüşlerden yakalanan nesne (Şekil 3.3.), bilgisayar görüş algoritmaları ile yeniden yapılandırabilmektedir. Birden fazla görünüş kullanarak nesnenin doğru olan 3 boyutlu yüzey modeli görünümünü ortaya

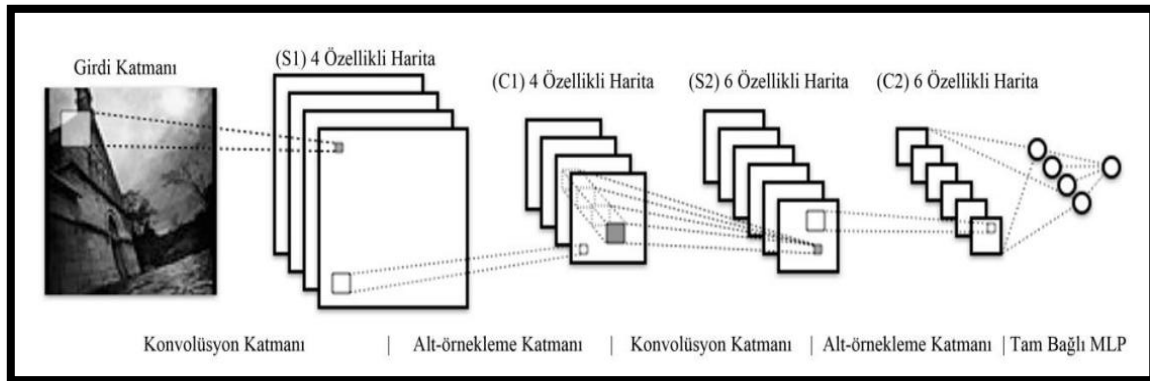
çıkarılabilmektedir. Ancak, tüm bu ilerlemelere rağmen, görüntüleri insanlarla aynı seviyede anlamak hala zorlayıcı olmaktadır.



Şekil 3.3. Altı farklı bakış açısından çekilen bir nesne[75]

Altı farklı bakış açısından çekilen bir nesnenin (ör., Üst insan vücudu) bir dizi görüntüsü verildiğinde, nesnenin yoğun bir 3B modeli bilgisayarlı görme algoritmaları kullanılarak yeniden oluşturulabilmektedir.

Konvolüsyonel yapay sinir ağları (Convolutional Neural Network-CNN) ileri yönlü bir yapay sinir ağı türüdür. Algoritmasının temeli, hayvanların görme merkezinden esinlenerek ortaya çıkmıştır. CNN ağı, ilk kez 1988 yılında Yann LeCun tarafından, 1998 yılına kadar geliştirme ve optimizasyon çalışmaları devam eden LeNet (Şekil 3.4.) isimli mimari olarak ortaya çıkmıştır[76].



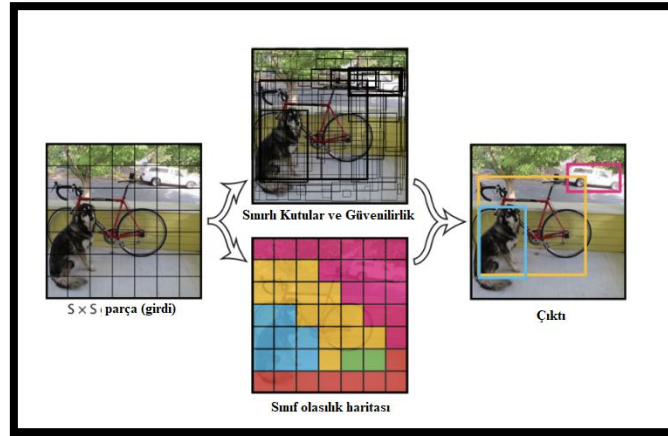
Şekil 3.4. LeNet Mimarisi [76]

CNN algoritmaları ses ve görüntü işleme alanlarında oldukça yaygın halde kullanılmaya başlanmış ve özellikle görüntü işleme üzerinde en iyi sonuçlar(state-of-art) elde edilmiştir.

Günümüzde, CNN tabanlı algoritmalar arasında en popüler olanlar R-CNN(Region Based CNN), Fast R-CNN, Faster R-CNN ve YOLO(You Only Look Once)'dur.

3.2.2. YOLO(You Only Look Once)'nun Çalışma Şekli

Yeni ve çok hızlı işleyen bir başka yaklaşım olan YOLO gerçek zamanlı üst düzey(satae-of-art) bir nesne tanıma sistemidir. Kısaltmasından da anlaşılacağı gibi; "Sadece Bir Kere Bakarsın (You Only Look Once)" cümlesinden türetilmiş olup söylenmek istenen insanların tek bir karedeki görüntüde olan objeleri hızlı bir şekilde bir bakışta bulabilmesidir. YOLO'nun da amacı bu hıza erişmek olmuştur. YOLO tasarımcılarına göre görüntü karesini soldan sağa taramak (ve o sırada ufak bir kutu içine düşen piksellere bakmak) ya da obje yeri hakkında fikir üretmek gerekli görülmemiştir. Tüm kare ve o karedeki tüm obje tipleri, yerleri hakkında global bir karar verilebilir. YOLO'da tek bir konvolüsyonel sinir ağı, sınırlı kutuları ve bu kutuların içindeki nesnelere göre sınıf olasılıklarını tahmin edebilmektedir.[20] Bunu yaparken, Şekil 3.5.'da gösterildiği gibi önce görüntüyü SxS boyutlarında parçalara böler ve her bir parça için m tane sınırlı kutu oluşturur. Her bir sınırlı kutu için ağ, sınıf olasılığını ve aralık değerini çıktı olarak verir. YOLO'nun temelinde denetimli bir eğitim verisi yatmaktadır. Burada denetimli eğitim ile kast edilen, bir görüntü için görüntü içindeki objeler, o objelerin yerleri ve büyüklük verilerinin olmasıdır. Ünlü Pascal VOC [77] eğitim verisi bu eğitim setlerinden biridir.



Şekil 3.5. YOLO'nun Çalışma Şekli[20]

3.2.3. NNPACK Darknet Kütüphanesi

NNPACK, sinir ağı hesaplamaları için bir hızlandırma paketidir. Çok çekirdekli işlemciler için yüksek performanslı konvolüsyonel ağ katmanları uygulamaları sağlamayı amaçlamaktadır[78]. Kütüphane, Georgia Tech'ten Marat Dukhan tarafından, Nicolas Vasilache

ve Facebook Yapay Zeka Araştırmaları'ndan Soumith Chintala'nın kapsamlı önerileri ile geliştirilmiştir.

NNPACK, Georgia Vuduc'un Georgia Teknoloji Enstitüsü, Bilgi İşlem Fakültesi, Hesaplamalı Bilim ve Mühendislik Fakültesi'ndeki HPC Garaj laboratuvarında bir araştırma projesidir. NNPACK' in özellikleri aşağıdaki gibi sıralanabilir[78].

Konvolüsyonel katmanlar için çoklu algoritmalar:

- Fourier dönüşümünü temel alan hızlı evrişim (baskın olmadan 16x16'ya kadar olan çekirdekler için)
- Winograd dönüşümüne dayanan hızlı evrişim (3x3 çekirdeksiz)
- Kapalı matris-matris çarpım algoritması (sınırlama yok)
- Doğrudan evrişim algoritması
- Nöral ağ katmanlarının çok iş parçacıklı SIMD uyumlu uygulamaları
- C99 ve Python'da harici bağımlılıklar olmadan uygulandı
- Birim testleriyle kapsamlı kapsama

NNPACK' in katmanları aşağıdaki gibi sıralanmaktadır.

- Dönüşümlü katman
- Çıkarım için optimize edilmiş ileri yayılım (nnp_convolution_inference)
- Eğitim için optimize edilmiş ileri yayılım (nnp_convolution_output)
- Antrenman için optimize edilmiş geriye doğru giriş gradyanı güncellemesi (nnp_convolution_input_gradient)
- Antrenman için optimize edilmiş geriye doğru çekirdek gradyan güncellemesi (nnp_convolution_kernel_gradient)
- Tamamen bağlı katman
- Çıkarım için optimize edilmiş ileri yayılım (FP16 ağırlıkları için nnp_fully_connected_inference ve nnp_fully_connected_inference_f16f32 sürümü)
- Eğitim için optimize edilmiş ileri yayılım (nnp_fully_connected_output)
- Max havuzlama katmanı
- Hem eğitim hem de çıkarım için ileriye doğru yayılım (nnp_max_pooling_output)
- ReLU katmanı (parametrelili negatif eğim ile)
- İsteğe bağlı olarak yerinde eğitim, hem çıkarım, hem de çıkarım için ileriye doğru yayılma (nnp_relu_output)
- Geriye doğru giriş gradyanı güncellemesi (nnp_relu_input_gradient)
- Softmax katmanı
- İsteğe bağlı olarak yerinde eğitim ve çıkarım için ileriye doğru yayılma (nnp_softmax_output)

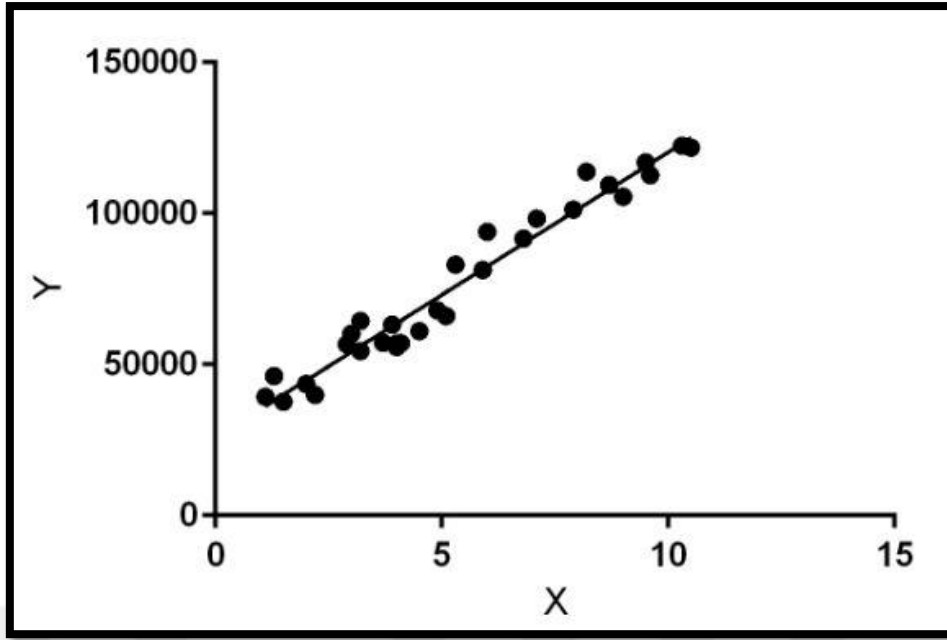
3.3. Regresyon

Büyük verilerdeki ve işlem yapılabilir iç görümlere yol açan kalıpları keşfetmek için birçok farklı türde makine öğrenme algoritması kullanılmaktadır. Yüksek düzeyde, bu farklı algoritmalar, tahminlerde bulunma konusunda veriler hakkında “öğrenme” ye dayalı olarak iki gruba ayrılabilir:

1. Denetimli Öğrenme
2. Denetimsiz Öğrenme

Pratik makine öğrenmesinin çoğunluğu denetimli öğrenmeyi kullanmaktadır. Denetimli öğrenme, girdi değişkenlerinin (x) ve bir çıktı değişkeninin (Y) olduğu ve girişten $Y = f(X)$ çıkışına kadar eşleme fonksiyonunu öğrenmek için bir algoritma kullanmaktadır.[79] Hedef, haritalama fonksiyonunu o kadar iyi tahmin etmektir, yeni giriş verileri (x) olduğunda, bu verinin çıkış değişkenleri (Y) tahmin edebilmektedir. Denetimli Makine Öğrenmesi algoritmalarının teknikleri doğrusal ve lojistik regresyon, çok sınıflı sınıflandırma, karar ağaçları ve destek vektör makinelerini içermektedir. Denetimli öğrenme, algoritmayı eğitmek için kullanılan verilerin doğru cevaplarla etiketlenmesini gerektirmektedir. Örneğin, bir sınıflandırma algoritması, hayvanın türleriyle ve bazı tanımlayıcı özelliklerle uygun şekilde etiketlenmiş bir görüntü veri kümesi üzerinde eğitildikten sonra hayvanları tanımlamayı öğrenebilmektedir. Denetimli öğrenme problemleri, regresyon ve sınıflandırma problemlerine olarak iki gruba ayrılabilir. Her iki sorunun da öznelik değişkenlerinden bağımlı özneliğin değerini tahmin edebilen bir modelin oluşturulması amaçlanmaktadır. İki görev arasındaki farkın, bağımlı özelliğin regresyon için sayısal ve sınıflandırma içinse kategorik olduğu bilinmektedir. Regresyon, bir veya daha fazla tahmin ediciye (sayısal ve kategorik değişkenler) dayanan bir model oluşturarak hedefin değerini (sayısal değişken) tahmin etme veri bilimi görevidir [80]. Bir regresyon probleminde, çıktı değişkeninin “maaş” veya “ağırlık” gibi gerçek veya sürekli bir değer olması gerekmektedir. Birçok farklı model kullanılabilir, en basit olan doğrusal regresyondur [79].

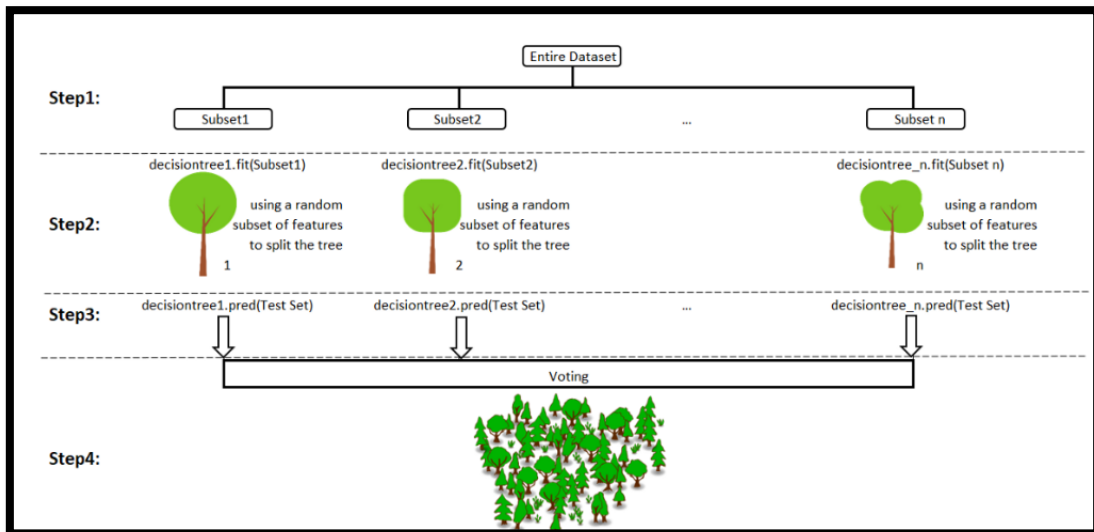
Doğrusal Regresyon, denetimli öğrenmeye dayanan bir makine öğrenme algoritmasıdır ve belirli bir bağımsız değişkeni (x) temel alarak bağımlı değişken bir değeri (y) tahmin etme görevini yerine getirmektedir. Dolayısıyla, bu regresyon tekniği, x (giriş) ve y (çıkış) arasında doğrusal bir ilişki bulmaktadır. Şekil 3.6.’da, X (girdi) ve Y (çıkış) ‘yi temsil etmektedir. Regresyon çizgisi, model için en uygun çizgi olmaktadır [81].



Şekil 3.6. X ve Y İlişkisi[81]

3.3.1. Rassal Orman

Rassal orman, toplanma yöntemi olarak torbalama ve bireysel model olarak karar ağacını kullanan bir topluluk modelidir. Rastgele ormanlar veya rastgele karar ormanları, sınıflandırma, regresyon ve eğitim zamanında çok sayıda karar ağacı oluşturularak ve sınıfların modu (sınıflandırma) veya ortalama tahmin (regresyon) sınıfı çıktısı olarak çalışan diğer görevler için kullanılmaktadır [82]. Rastgele karar ormanları için ilk algoritma, Ho'nun formülünde Eugene Kleinberg tarafından önerilen sınıflandırma için "stokastik ayrımcılık" yaklaşımını uygulamanın bir yolu olan, rastgele alt uzay yöntemi kullanılarak Tin Kam Ho [90] tarafından yaratılmıştır [91]. Çalışma prensibi Şekil 3.7.'de gösterilmektedir.



Şekil 3.7. Rassal Orman Çalışma Şekli[82]

Modelin algoritmasının adımları aşağıdaki gibidir.

Adım 1: Eğitim setinden n (örn. 1000) rastgele altkümeyi seç.

Adım 2: n (örn. 1000) karar ağacını eğit.

- Bir karar ağacını eğitmek için bir rastgele altküme kullanılmaktadır.
- Her karar ağacı için en uygun bölme şekli özelliklerin rastgele bir alt kümesine dayanmaktadır(örneğin toplamda 10 özellik varsa, bölmek için 10 özellikten 5'i rastgele seçilir).

Adım 3: Her bir ağaç test setindeki kayıtları / adayları bağımsız olarak tahmin eder.

Adım 4: Son tahminde bulun.

Test setindeki her aday için, Rassel Orman, bu adayın nihai tahmini olarak çoğunluk oyu olan sınıfı (örneğin kedi veya köpek) kullanır[82]. Rassel Orman modelinin algoritmasının Python kodu Şekil 3.8.'da gösterilmektedir.

```
import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.ensemble import RandomForestRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

df = pd.read_excel('..\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

kVals = range(1, 15, 1)
accuracies = []

for k in kVals:

    model = RandomForestRegressor(max_depth=7, n_estimators=20)
    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 score=%.2f%%" % (k, score * 100))
    print("k=%d, mae = %.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

# find the value of k that has the largest accuracy
i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))
```

Şekil 3.8. Rassel Orman Python Kodu

3.3.2. Karar Ağacı

Karar ağacı, bir ağaç yapısı şeklinde regresyon veya sınıflandırma modelleri oluşturabilmektedir. Bir veri kümesini daha küçük ve daha küçük alt gruplara ayırırken, aynı zamanda ilişkili bir karar ağacı kademeli olarak geliştirebilmektedir. Sonuç, karar düğümleri ve yaprak düğümleri olan bir ağaçtır. Bir karar düğümü (örneğin, Hava Durumu), her biri test edilen özellik için değerleri temsil eden iki veya daha fazla dalı (örneğin, Güneşli, Bulutlu ve Yağmurlu) içermektedir(Şekil 3.9.). Yaprak düğümü (örneğin, Futbol Oyna) sayısal hedef için bir karardır. Ağaçtaki en üst karar düğümü, kök düğümü adı verilen en iyi tahmin ediciye karşılık gelir. Karar ağaçları hem kategorik hem de sayısal verileri ele alabilmektedir [83].



Şekil 3.9. Karar Ağacı Örneği[83]

J. R. Quinlan tarafından ID3 olarak adlandırılan karar ağaçları oluşturmak için temel algoritma, geriye dönüşü olmayan olası dallar arasında yukarıdan aşağıya, açgözlü(greedy search) bir arama kullanmaktadır. ID3 algoritması, regresyon için bir Karar Ağacı oluşturmak için, bilgi kazancını Standart Sapma Azaltma ile değiştirerek kullanılabilir. Hedef için standart sapma Şekil 3.10.'da ki gibi hesaplanmaktadır[83].

Hedef
Futbol Oyna
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$S = \sqrt{\frac{\sum (x - \mu)^2}{n}} = 9.32$$

Şekil 3.10. Standart Sapma Hesaplaması[83]

Örneğin hava durumu ve futbol oyna ilişkisi aşağıdaki gibi hesaplanabilir.

$$\begin{aligned}
 S(\text{Hava Durumu}, \text{Futbol Oyna}) &= P(\text{Güneşli}) * S(\text{Güneşli}) + P(\text{Bulutlu}) * S(\text{Bulutlu}) + \\
 &P(\text{Yağmurlu}) * S(\text{Yağmurlu}) \\
 &= 5/14 * 10.870 + 4/14 * 3.491 + 5/14 * 7.782 \\
 &= 7.659 \\
 SDR(\text{Hava Durumu}, \text{Futbol Oyna}) &= S(\text{Güneşli}) - S(\text{Hava Durumu}, \text{Futbol Oyna}) \\
 &= 1.662
 \end{aligned}$$

3.3.3. Adaboost Karar Ağacı

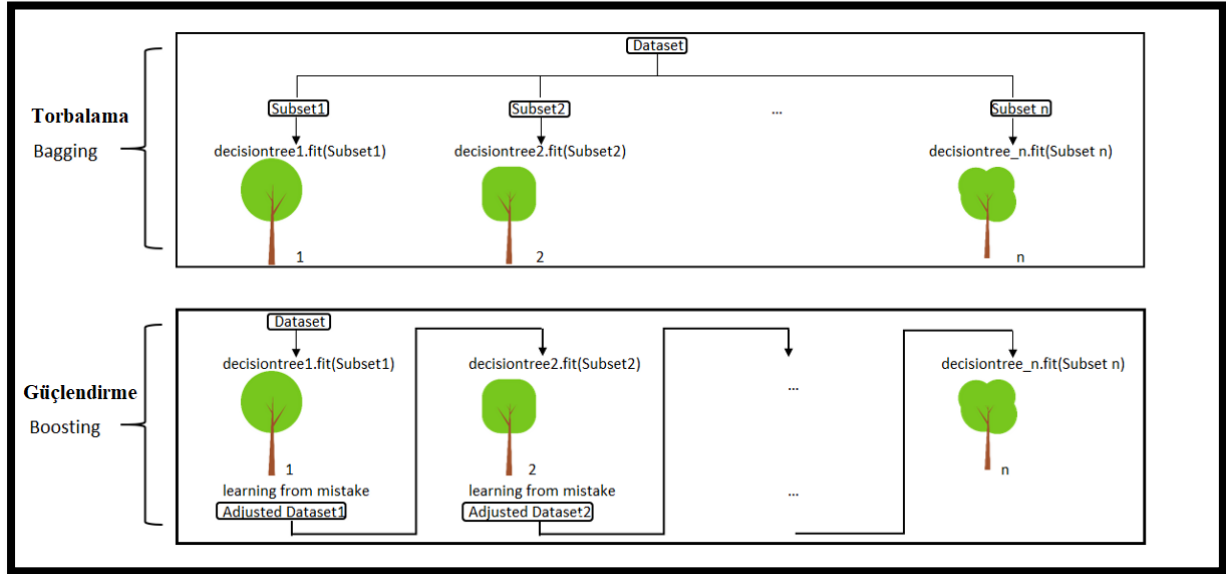
Topluluk öğrenme (Ensemble Learning), genel olarak, birkaç farklı modele dayalı tahminler yapan bir modeldir. Bireysel modelleri birleştirerek, topluluk modeli daha esnek (daha az önyargılı) ve daha az veri duyarlı (daha az değişken) olma eğilimindedir.[82] En popüler iki topluluk metodu torbalama (bagging) ve güçlendirme (boosting).

- Torbalama: Bir grup bireysel modeli paralel olarak eğitmekte ve her model verinin rastgele bir alt kümesi ile eğitilmektedir.
- Güçlendirme: Bir dizi bireysel modeli sıralı bir şekilde eğitilmekte ve her bir model, önceki model tarafından yapılan hatalardan ders alarak eğitilmektedir.

Önyükleme Toplaması (veya kısaca torbalama), basit ve çok güçlü bir topluluk yöntemidir. Torbalama, Bootstrap prosedürünün tipik olarak karar ağaçları olan yüksek değişkenlikli bir makine öğrenme algoritmasına uygulanmasıdır. Güçlendirme, zayıf öğrencileri daha güçlü öğrenenler haline getirmek için ağırlıklı ortalamaları kullanan bir algoritma grubunu ifade eder. Her bir modelin bağımsız çalışmasını sağlayan torbalamanın aksine, daha sonra

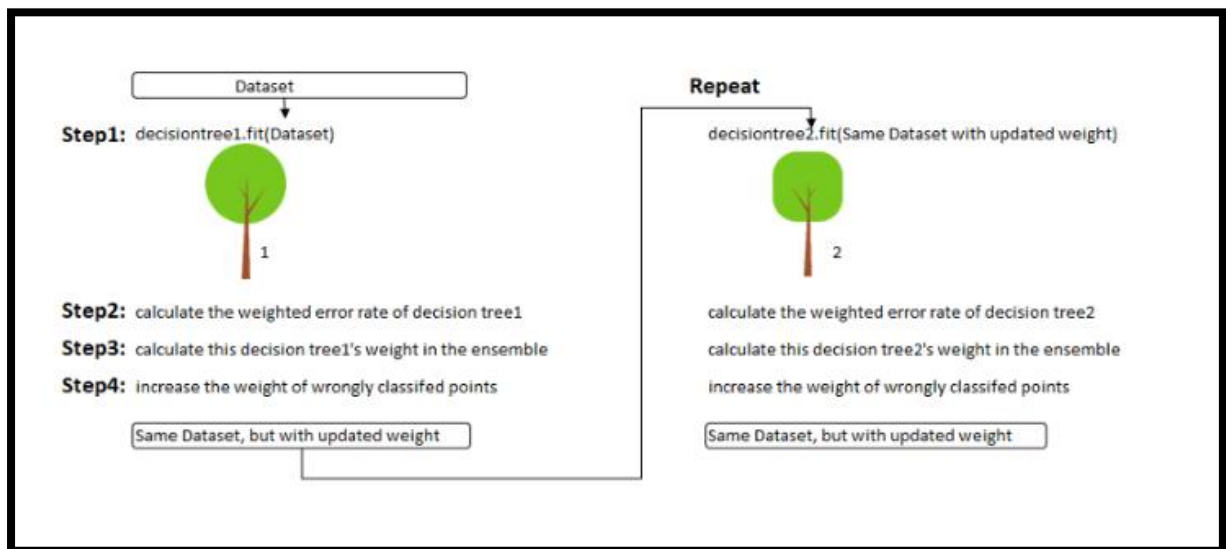
herhangi bir modeli tercih etmeden çıktılarını toplar. Güçlendirme tamamen “ekip çalışması” ile ilgilidir. Çalışan her model, bir sonraki modelin hangi özelliklere odaklanacağını belirlemektedir.

Torbalama ve güçlendirme metodlarının çalışma prensibi Şekil 3.11.’da gösterilmiştir.



Şekil 3.11. Torbalama ve Güçlendirme Metodlarının Çalışma Prensibi[82]

AdaBoost, güçlendirici bir topluluk modelidir ve özellikle karar ağacıyla birlikte çalışmaktadır. Güçlendirme modelinin anahtarı önceki hatalardan öğrenebilmesidir (örn. yanlış sınıflandırma veri noktaları.) Yani AdaBoost, yanlış sınıflandırılmış veri noktalarının ağırlığını artırarak hataları öğrenebilmektedir. Adaboost Karar Ağacı’nın çalışma prensibi Şekil 3.12.’da gösterilmiştir.



Şekil 3.12. Adaboost Karar Ağacı’nın Çalışma Prensibi[82]

Şekil 3.12.'da görüldüğü gibi Adaboost Karar Ağacı'nın çalışma adımları aşağıdaki gibi sıralanabilir.

Adım 0: Veri noktalarının ağırlıklarını başlat. Eğer eğitim setinde 100 veri noktası varsa, o zaman her bir noktanın başlangıç ağırlığı $1/100 = 0.01$ olmalıdır.

Adım 1: Karar Ağacını eđit.

Adım 2: Karar ağacının ağırlıklı hata oranını (e) hesapla. Ağırlıklı hata oranı (e) toplamda kaç yanlış tahmin olduđu ve yanlış tahminleri veri noktasının ağırlığına göre farklı şekilde ele alır. Ağırlık ne kadar yüksek olursa, (e) 'nin hesaplanması sırasında karşılık gelen hata o kadar fazla olacaktır.

Adım 3: Bu karar ağacının ağırlığını hesapla.

$$\text{Karar Ağacının Ağırlığı} = \text{Öğrenme Oranı} * \log\left(\frac{1-e}{e}\right) \quad (3.1)$$

Adım 4: Yanlış sınıflandırılmış noktaların ağırlıklarını güncelle.

Her veri noktasının ağırlığı;

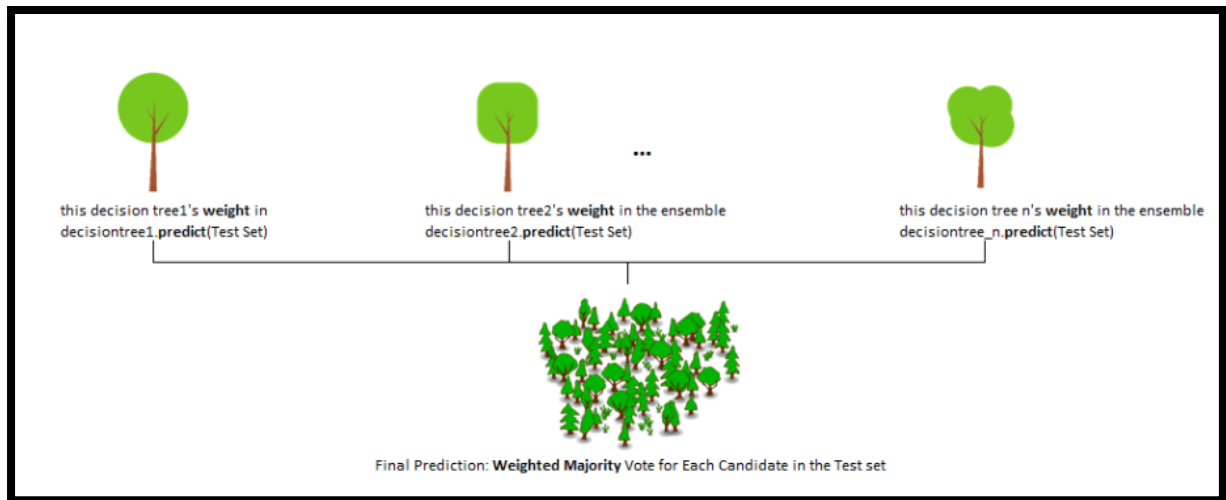
- Model bu veri noktasını doğruladıysa, ağırlık aynı kalır.
- Model bu veri noktasını yanlış yaptıysa,

$$\text{Bu Noktanın Yeni Ağırlığı} = \text{Eski Ağırlık} * np.exp(\text{Ağacın Ağırlığı}) \quad (3.2)$$

Adım 5: Adım 1'i tekrarla (eđitmek için belirlediğimiz ağaç sayısına ulaşılanaya kadar)

Adım 6: Son tahminde bulun.

AdaBoost, her ağacın ağırlığını toplayarak yeni bir tahmin yapar ve her ağacın tahminini çarpır. Sonuç olarak, daha yüksek bir ağırlığa sahip olan ağaç nihai kararı etkilemek için daha fazla güce sahip olacaktır(Şekil 3.13.).[82]



Şekil 3.13. Adaboost Karar Ağacı Sonucu[82]

Adaboost Karar Ağacı regresyon modelinin Python kodu Şekil 3.14.'da verilmiştir.

```
import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.neural_network import MLPRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
#best is model = AdaBoostRegressor(DecisionTreeRegressor(max_depth=10),n_estimators=25,loss='exponential',learning_rate=1)
df = pd.read_excel('..\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

kVals = range(1, 15, 1)
accuracies = []

for k in kVals:

    model = AdaBoostRegressor(DecisionTreeRegressor(max_depth=10),n_estimators=25,loss='exponential',learning_rate=1)
    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 score=%.2f%%" % (k, score * 100))
    print("k=%d, mae = %.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

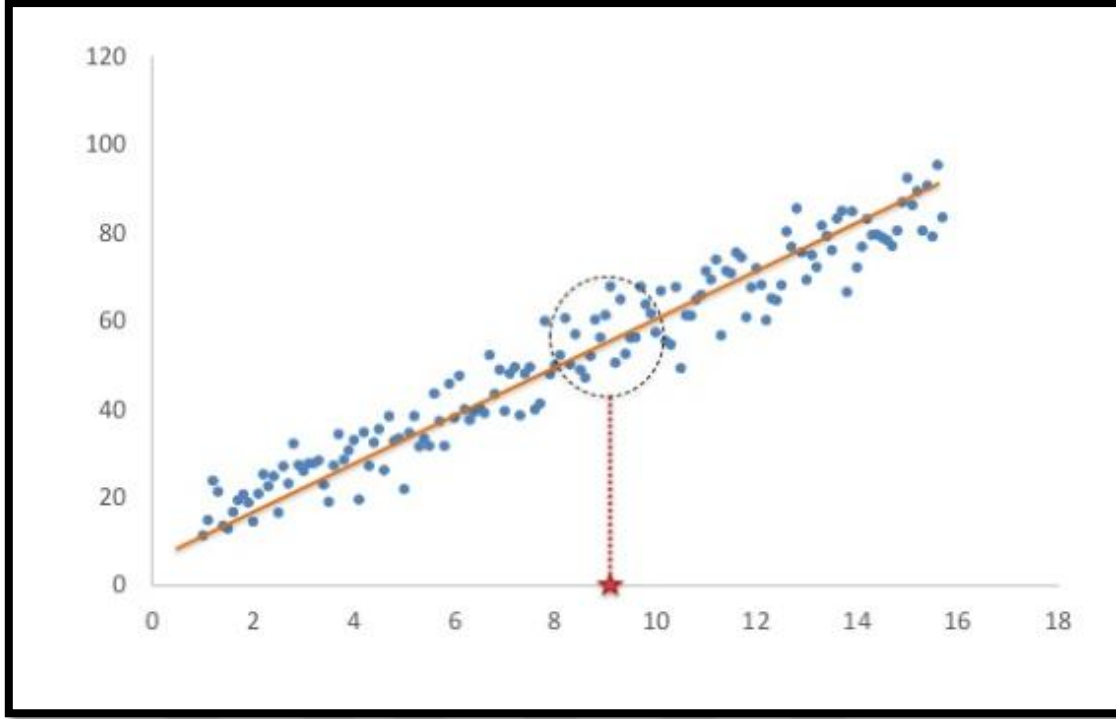
# find the value of k that has the largest accuracy

i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))
```

Şekil 3.14. Adaboost Karar Ağacı Python Kodu

3.3.4. K-En Yakın Komşu

K en yakın komşu, test verilerinin çoğunluğun olduğu sınıfa göre sınıflandırıldığı denetimli bir metottur. Bu algoritmanın amacı, yeni verilerin niteliklerine ve eğitim verilerine dayanarak sınıflandırma yapmaktır. Sınıflandırma herhangi bir modele kullanmaz ve sadece hafızaya dayanır. KNN' in prensibi, test edilen veri setine eğitim setinden en yakın k tane nesneyi bulmaktır. KNN algoritması çok basittir, k-en yakın komşuyu belirlemek için test verisi ile eğitim verisi arasındaki en yakın uzaklığı bulmaya ve sonra KNN'in çoğunluğuna göre test verisi için tahminde bulunmaya dayanmaktadır(Şekil 3.15.). KNN metotunun diğer metotlara göre birçok avantajı bulunmaktadır. Örneğin, gürültülü eğitim verilerinde güçlü ve küçük eğitim verilerinde de verimlidir. Bu yöntem, alakasız veya gereksiz değişkenlere karşı çok hassastır, çünkü tüm değişkenler benzerlik ve sınıflandırmaya katkıda bulunur [86].



Şekil 3.15. KNN'in Yapısı[87]

KNN, sınıflandırma yaklaşımlarını kullanarak çıkış değişkenlerini tahmin etmek için kullanılan yöntemlerden biridir. Sınıflandırma yaklaşımında, veri seti eğitim verilerine ve test verilerine bölünmektedir. KNN, verilen test verilerini eğitim verisiyle karşılaştırmak için benzerlik ölçüsü kullanmaktadır. Kullanılan benzerlik ölçütlerinden biri, eğitim verileri (x_{train}) ve veri testi (x_{test}) arasındaki Öklid mesafesidir ve 3.3'de yer alan formül kullanılarak hesaplanmaktadır[88].

$$d(x_{train,i}, x_{test,i}) = \sqrt{\sum_{i=1}^n (x_{train,i} - x_{test,i})^2} \quad (3.3)$$

KNN, çıkış değişkenlerini öngörmeye test verilerine yakın olan k eğitim verisini seçmektedir. KNN regresyonu, değeri tahmin etmek için 3.4'de yer alan formülü kullanmaktadır.

$$y_t^1 = \frac{1}{k} \sum_{j=1}^k y_j \quad (3.4)$$

Formülde k , y_j 'ye en yakın komşuların sayısıdır. Bu formül zaman serisi verilerinde kullanıldığında daha az etkilidir, çünkü gözlemler arasındaki ilişkiyi dikkate almaz (zaman), bu yüzden veri testinin tahmin edilmesinde 3.5' de gösterilen genel formül kullanılmaktadır.

$$y_t^1 = \sum_{j=1}^k w_j y_j \quad (3.5)$$

Formülde j. komşusu için w_j ağırlığıdır. Bu ağırlık gözlemlenen verilere göre ayarlanabilir, $w = n/\text{eğitim verisi ile } w_j = j/n\text{'dir}$ [88]. Ayrıca, bu model KNN'nin bir zaman serisi modeli olarak adlandırılmaktadır. K en yakın komşu regresyon modelinin Python kodu Şekil 3.16.'da verilmiştir.

```
import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.neighbors import KNeighborsRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

#k=7 için en iyi degeri yakaladık

df = pd.read_excel('..\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

kVals = range(1,10, 1)
accuracies = []

# loop over various values of `k` for the k-Nearest Neighbor classifier

for k in kVals:
    # train the k-Nearest Neighbor classifier with the current value of `k`
    model = KNeighborsRegressor(n_neighbors=k)
    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 score=%.2f%%" % (k, score * 100))
    print("k=%d, mae = %.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

# find the value of k that has the largest accuracy

i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))
```

Şekil 3.16. KNN Python Kodu

3.3.5. Adaboost K-En Yakın Komşu

AdaBoost'un temel ilkesi, verilerin tekrar tekrar değiştirilmiş versiyonlarında bir dizi zayıf öğrenici grubuna (yani küçük karar ağaçları gibi rastgele tahminlerden sadece biraz daha iyi olan modeller) uymaktır. Hepsinden gelen tahminler, nihai tahminin üretilmesi için ağırlıklı

çoğunluk oyu (veya toplamı) ile birleştirilmektedir[82]. Bu modelde, K en yakın komşu regresörü Adaboost Regresör ile birlikte kullanılmıştır. Adaboost K en yakın komşu regresyon modelinin Python kodu Şekil 3.17.'da verilmiştir.

```
import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.neural_network import MLPRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import AdaBoostRegressor
from sklearn.neighbors import KNeighborsRegressor
#best is model = AdaBoostRegressor(DecisionTreeRegressor(max_depth=10),n_estimators=25,loss='exponential',learning_rate=1)
df = pd.read_excel('..\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

print(len(X_train))

kVals = range(1, 20, 1)
accuracies = []

for k in kVals:

    model = AdaBoostRegressor(KNeighborsRegressor(n_neighbors=5),n_estimators=5,loss='exponential',learning_rate=1)
    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 error=%.2f%%" % (k, score * 100))
    print("k=%d, mae = %.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

# find the value of k that has the largest accuracy

i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))
```

Şekil 3.17. Adaboost K en yakın komşu Python Kodu

3.3.6. Topluluk (Ensemble) K-En Yakın Komşu

Topluluk tekniği, en iyi modeli seçmeyen bir tekniktir. Birçok aday arasında ve en iyi modelden modelleri tahmin ederek birleştirir ve her birine ağırlık(weight) vermektedir. Topluluk tekniği, standart modellerin tahmin kabiliyetini geliştirmede en iyi modellerden biri haline gelmiştir. Topluluk modeli iki aşamada açıklanır, ilk aşama en iyi topluluk üyeleri arasından çıktı değerini seçerek son tahmini elde etmektir. İkinci aşama ise bazı algoritma [84] kombinasyonları kullanarak topluluk üyelerinin çıkış değişkenini birleştirmektir. Topluluk tekniği özellikle hava durumu ve iklim tahminleri alanlarında oldukça popüler hale gelmiştir. Son zamanlarda gerçekleştirilen bir çalışma birkaç modelin kombinasyonunun sağlamlık ve güvenilirliği arttırdığını göstermektedir [85].

Model çeşitliliği, temel öğrencilerini geride bırakan topluluklar oluşturmanın kilit bir bileşenidir. Önceden tasarlanmış KNN topluluklarında bu temel olarak, özelliklerin rastgele alt kümelerini veya optimize edilmiş özellik alt kümelerini seçerek elde edilmekte; özellikler seçildikten sonra k, modeller bir araya getirilmeden önce her bir alt küme için tipik olarak optimize edilmektedir. K, bu kadar önemli bir parametre olduğundan ve farklı k değerleri farklı regresyon tahminleri vereceğinden, topluluklar içinde k'nin rastgele seçilmesinin, aynı numunelerde veya özellik setlerinde bile önemli model çeşitliliği üretmesi muhtemel olmaktadır. bu Şekilde, değişken k, veri noktalarının etrafında tanımlanan topolojik komşuluk alanını değiştirir, tahmin sonuç ilişkilerini görüntülemek ve bir regresyon fonksiyonunun düzgünleşmesini sağlamak için farklı lensler sağlamaktadır. Topluluk K en yakın komşu regresyon modelinin Python kodu Şekil 3.18.'da verilmiştir.

```
import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import BaggingRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

df = pd.read_excel('.\\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

kVals = range(1, 20, 1)
accuracies = []

# loop over various values of `k` for the k-Nearest Neighbor classifier
for k in kVals:

    basemodel = KNeighborsRegressor(n_neighbors=k)
    model = BaggingRegressor(base_estimator=basemodel, n_estimators=20,bootstrap_features=True,max_features=1,random_state=1)
    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 score=%.2f%%" % (k, score * 100))
    print("k=%d, mae = %.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

# find the value of k that has the largest accuracy
i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))
```

Şekil 3.18. Topluluk(Ensemble) KNN Python Kodu

3.3.7. MLP(Çok Katmanlı Algılayıcı) Regresör

Çok Katmanlı Algılayıcı (MLP), bir fonksiyonu ($f(\cdot): R^m \rightarrow R^o$) veri seti üzerinde eğiterek öğrenen denetimli bir algoritmadır. Burada m, giriş için boyutların ve o, çıktı için boyut

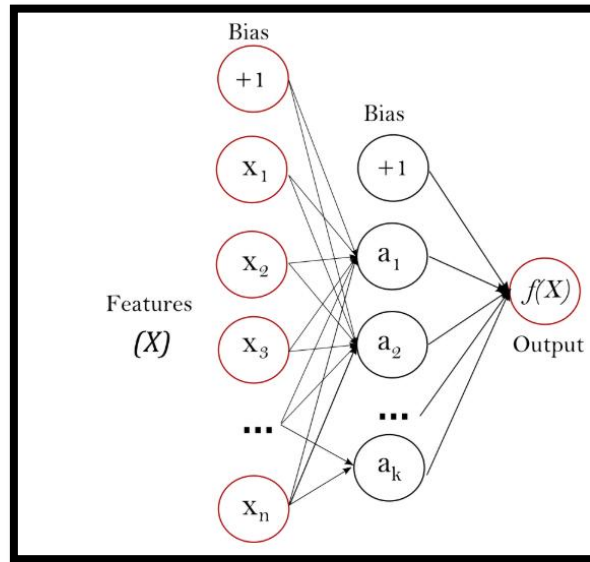
sayısıdır. Bir dizi özellik göz önüne alındığında $X=x_1, x_2, x_3, \dots, x_m$ ve bir hedef y , sınıflandırma veya regresyon için doğrusal olmayan bir fonksiyon tahminicisini öğrenebilmektedir. Lojistik regresyondan farklıdır; giriş ve çıkış katmanı arasında, gizli katmanlar adı verilen doğrusal olmayan bir veya daha fazla katman olabilir. Şekil x, skaler çıktılı bir gizli katman MLP'yi göstermektedir. MLPRegressor Sınıfı, çıktı katmanında aktivasyon fonksiyonu olmayan, aynı zamanda kimlik fonksiyonunun aktivasyon fonksiyonu olarak kullanıldığı da görülebilen, geri yayılma kullanımıyla çalışan çok katmanlı bir algılayıcı (MLP) uygular. Bu nedenle, kayıp fonksiyonu olarak kare hatasını kullanır ve çıktı sürekli bir değerler kümesidir.[89]

Çok Katmanlı Algılayıcı'nın avantajları:

- Lineer olmayan modelleri öğrenebilme kapasitesi
- Modelleri gerçek zamanlı öğrenebilme kapasitesi (çevirim içi öğrenme)

Çok Katmanlı Algılayıcı'nın dezavantajları:

- Gizli katmanlara sahip MLP, birden fazla yerel minimumun olduğu dışbükey olmayan bir kayıp fonksiyonuna sahiptir. Bu nedenle farklı rasgele ağırlık başlatmaları farklı doğrulama doğruluğuna yol açabilmektedir.
- MLP, gizli nöronların, katmanların ve yinelemelerin sayısı gibi bazı hiperparametrelerin ayarlanmasını gerektirmektedir.
- MLP özellik ölçeklendirmeye duyarlıdır.



Şekil 3.19. MLP Sinir Ağı Yapısı[89]

Çok Katmanlı Algılayıcı modelinin algoritmasının Python kodu Şekil 3.20.'da gösterilmektedir.

```

import pandas as pd
from pandas import ExcelFile
from sklearn.model_selection import train_test_split as split
from sklearn.neural_network import MLPRegressor
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
|
df = pd.read_excel('..\results.xlsx')
X=df.iloc[:,[1,2,3,4]]
y=df.iloc[:,5]

X_train, X_test, y_train, y_test = split(X, y, test_size=0.2, random_state=1)

print(len(X_train))

kVals = range(1, 10, 1)
accuracies = []

for k in kVals:

    model = MLPRegressor(hidden_layer_sizes=(50,),
                          activation='tanh',
                          solver='lbfgs',
                          learning_rate='adaptive',
                          max_iter=2000,
                          learning_rate_init=0.07,
                          alpha=0.001)

    model.fit(X_train, y_train)
    # evaluate the model and update the accuracies list
    y_predicted=model.predict(X_test)
    score = model.score(X_test, y_test)
    print("k=%d, R^2 score=%0.2f%%" % (k, score * 100))
    print("k=%d, mae = %0.2f" % (k, mean_absolute_error(y_test,y_predicted)))
    print("k=%d, mse =%0.2f%%" % (k, mean_squared_error(y_test,y_predicted) * 100))
    pred=np.array(y_predicted)
    org=np.array(y_test)
    dif= abs( pred- org )
    dif=1-dif/org
    div = pred/org
    ix=np.where(div>1)
    div[ix]=1/div[ix]
    print("k=%d, dist accuracy =%0.4f%%" % (k, dif.mean() * 100))
    print("k=%d, dist mmm accuracy =%0.4f%%\n\n" % (k, div.mean() * 100))

    accuracies.append(score)

# find the value of k that has the largest accuracy

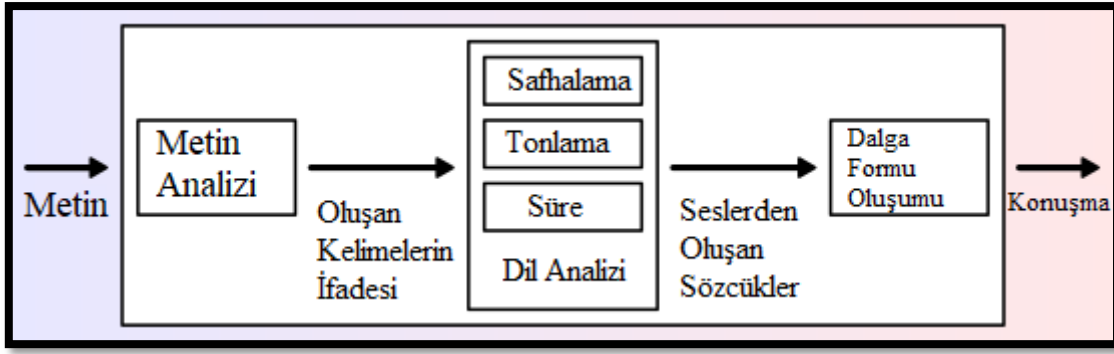
i = np.argmax(accuracies)
print("k=%d achieved best R^2 of %0.2f%% on validation data" % (kVals[i],
accuracies[i] * 100))

```

Şekil.3.20. MLP Python Kodu

3.4. Google Text to Speech Modülü ile Sesli Bildirim Oluşturulması

Metin-Konuşma (Text-to-Speech, TTS), bilgisayarların insanlar gibi insan sözcükleri söylemesine olanak sağlayan Konuşma Sentezi teknolojisidir[92]. Bilgisayarlar metni yorumlayabilir ve uygun tonlama ile kullanıcılara (birkaç farklı dilde) konuşma yeteneği ile 'insan benzeri' konuşma sesini üretebilmektedir. Google Metin-Seslendirme (gTTS) modülü, Google'ın Metin-Konuşma servisiyle etkileşime girmeye ve kulaklık / hoparlörden çalınabilecek Python'da bir mp3 dosyası oluşturmaya yarayan bir Python kütüphanesidir. Konuşma sentezinin çalışma şekli Şekil 3.21.'da gösterilmektedir.



Şekil 3.21. Konuşma Sentezi Çalışma Şekli

Google Metin-Seslendirme modülü sistemde elde edilen nesne tanıma ve mesafe tespiti sonuçlarını sesli bir şekilde görme engelli kullanıcılara iletmek amacıyla kullanılmıştır. Bu sayede, text halinde tutulan bilgiler kullanıcılara belirli süre aralıkları ile sesli bildirim olarak gönderilmiştir.

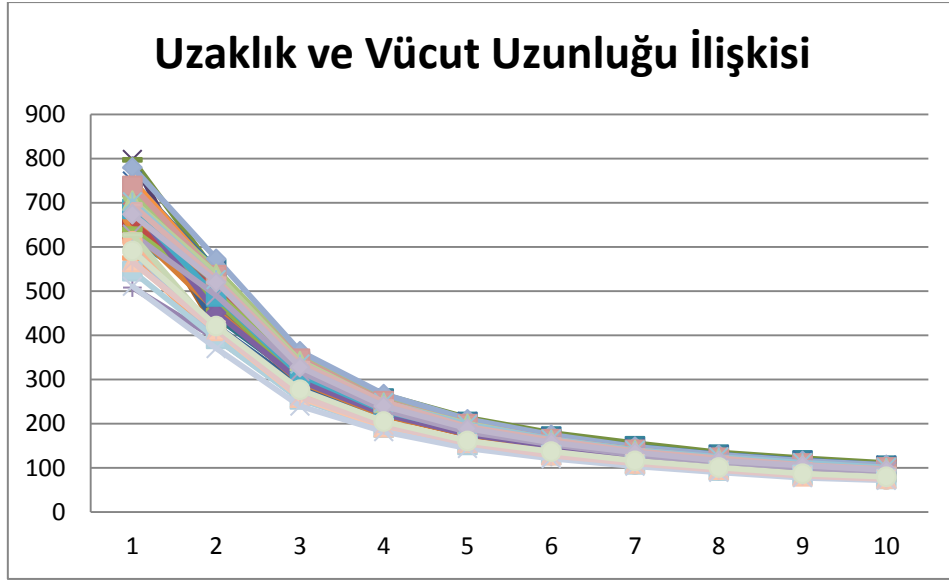
4. BULGULAR ve TARTIŞMA

Bu çalışmada, görme engelli bireylerin günlük yaşamlarında, açık ve kapalı ortamlarda kullanabilecekleri, kendi kendilerine bağımsız hareket etmelerine yardımcı olacak, taşınabilirliği ve kullanımı kolay, gerçek zamanlı nesne tanıma ve uzaklık tespiti yapabilen bir sistem tasarlanarak gömülü bir cihaz üzerinde uygulanmıştır. Raspberry pi 3 üzerinde gerçekleştirilen sistemde, nesne tanıma modülü olarak gerçek zamanlı nesne tanıma sistemi olan YOLO kullanılmıştır. Tek kamera ile uzaklık tespiti yapabilmek adına öncelikle akıllı telefon kamerası kullanılarak 1 metreden 10 metreye kadar resimler alınmış ve 500 adet vücut ve 500 adet yüz resmi olmak üzere toplam 1000 resimden oluşan bir veri seti oluşturulmuştur. Vücut ve yüzlerin boy/genişlik özelliklerindeki değişimler, Karar Ağacı, K-En Yakın Komşu, Adaboost Karar Ağacı, Adaboost K-En Yakın Komşu, MLP Regressör, Rastal Orman ve Destek Vektör Regresör metotları kullanılarak analiz edilmiştir. Bu metotlar arasında en yüksek doğruluk oranını veren %98,5 doğrulukla Rastal Orman metodu olmuştur. Sistem prototipi oluşturulurken daha istikrarlı ve güvenilir bir yapı oluşturmak adına Raspberry Pi kamera modülü kullanılarak 2 metreden 10 metreye kadar aynı şekilde görüntüler alınarak oluşturulan veri seti, Raspberry Pi üzerinde yukarıdaki metotlar kullanılarak analiz edilmiştir. Aynı şekilde kullanılan metotlar arasında en yüksek mesafe doğruluk oranını veren %98,8 doğrulukla Rastal Orman metodu olmuştur.

4.1. Uzaklık Tespiti için Uygulanan Regresyon Tekniklerinin Teorik Test Sonuçları

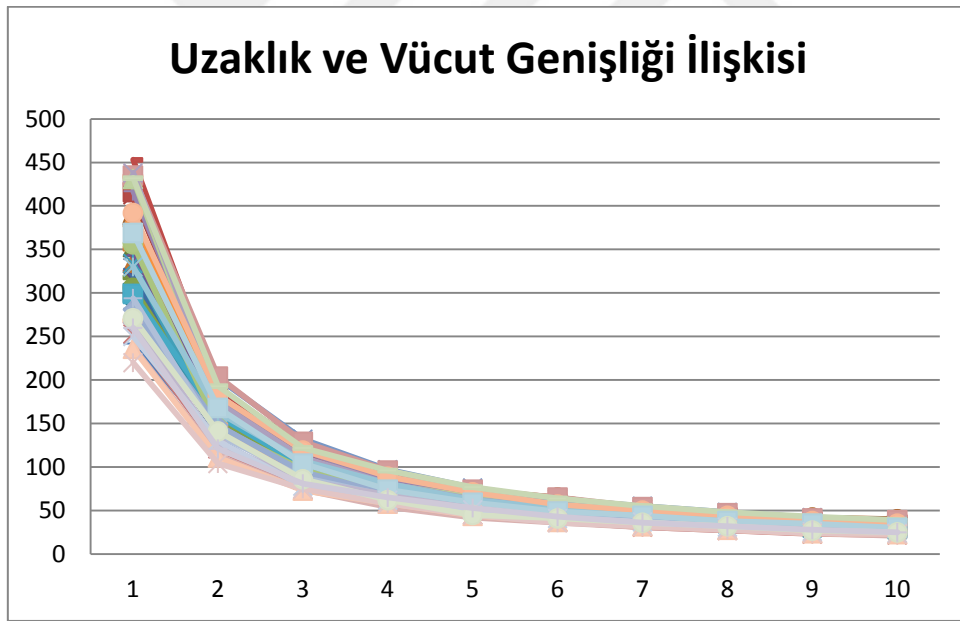
Tasarlanan sistemde uzaklık tespiti modülü için; ilk olarak akıllı telefon tarafından çekilen adet görüntüden alınan, yüz ve vücut boyutlarından oluşan veri seti yukarıda bahsedilen regresyon teknikleri ile eğitilerek test edilmiştir.

Veri setinde, vücut uzunluğu (B_h), vücut genişliği (B_w), yüz uzunluğu (F_h) ve yüz genişliği (F_w) olan vücut ve yüzün sınırlayıcı kutusu boyutları, eğitim sürecinde kullanılan özellik vektörünü oluşturmaktadır. Bu özellikler ile uzaklık arasındaki ilişki Şekil 4.1., 4.2., 4.3., 4.4.' daki grafiklerde gösterilmiştir.



Şekil 4.1. Uzaklık ve Vücut Uzunluğu İlişkisi Grafiği

Şekil 4.1.'da vücut uzunluğunun uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi vücut uzunluğu uzaklık arttıkça azalmaktadır.



Şekil 4.2. Uzaklık ve Vücut Genişliği İlişkisi Grafiği

Şekil 4.2.'da vücut genişliğinin uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi vücut genişliği uzaklık arttıkça azalmaktadır.



Şekil 4.3. Uzaklık ve Yüz Uzunluğu İlişkisi Grafiği

Şekil 4.3.'da yüz uzunluğunun uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi yüz uzunluğu uzaklık arttıkça azalmaktadır.



Şekil 4.4. Uzaklık ve Yüz Genişliği İlişkisi Grafiği

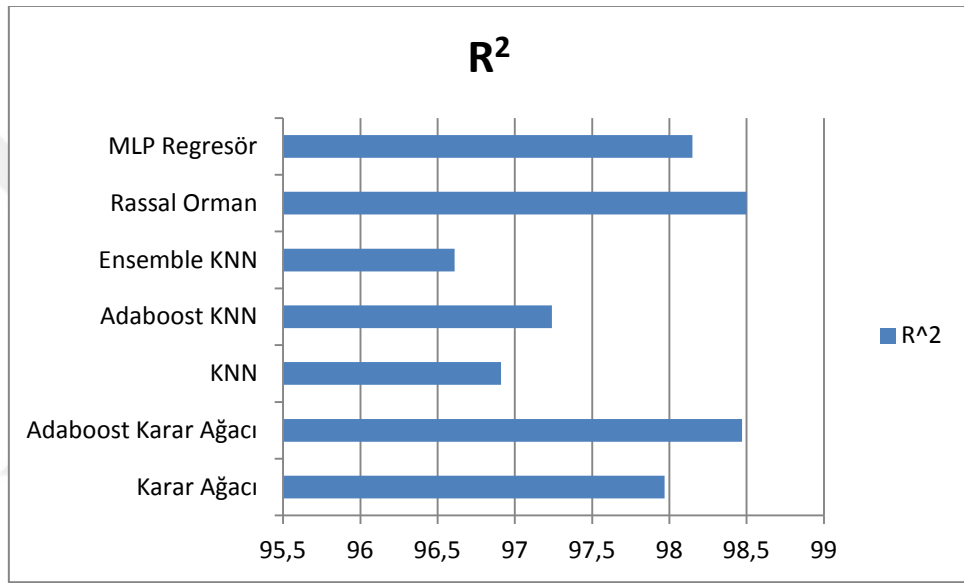
Şekil 4.4.'da yüz genişliğinin uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi yüz genişliği uzaklık arttıkça azalmaktadır.

Modellerin eğitimi için, girdi veri seti rastgele bir eğitim setine (verilerin% 80'i) ve bir test setine (verilerin %20'si) bölünmüştür. Sonuçlar 5 farklı parametre ile gösterilmiştir. Bu parametreler;

- R-kare (R^2), bir regresyon modelinde bağımsız bir değişken veya değişkenler tarafından açıklanan bağımlı değişken için varyans oranını temsil eden istatistiksel bir ölçüdür. Formülü aşağıdaki gibidir [93].

$$R^2 = 1 - \frac{\text{Açıklanan Varyasyon}}{\text{Toplam Varyasyon}} \quad (4.1)$$

Uygulanan tekniklerin sonuçlarına göre R^2 değerleri yüzde olarak Şekil 4.5.'da ki grafikte verilmiştir.



Şekil 4.5. R^2 Değerleri Grafiği

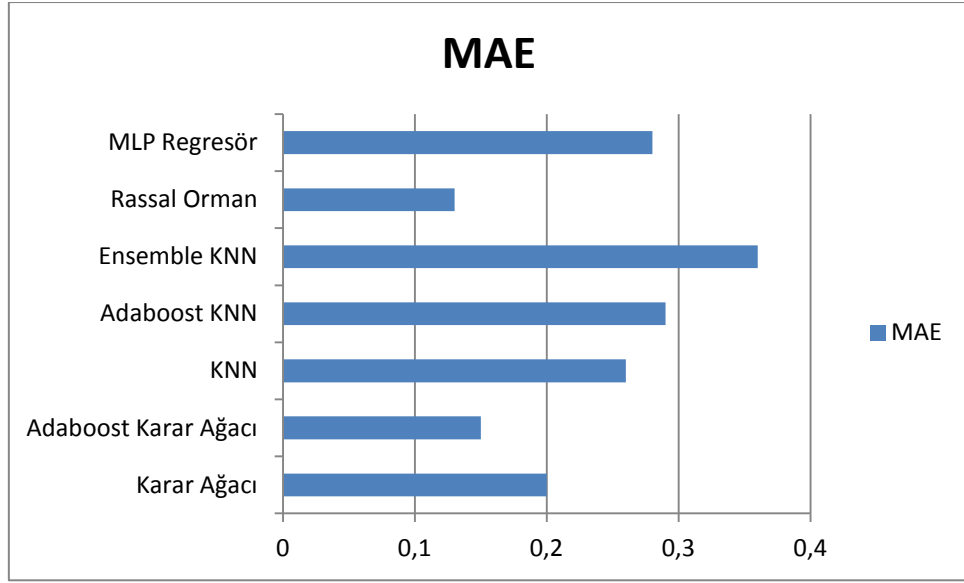
- Ortalama Mutlak Hata (Mean Absolute Error-MAE), iki sürekli değişken arasındaki farkın bir ölçüsüdür. MAE y_i ve x_i arasındaki ortalama mutlak fark olarak açıklanır. Formülü aşağıdaki gibidir [94].

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_i - y_i^1| \quad (4.2)$$

y_i : Gerçek değer

y_i^1 : Tahnin edilen değer

Uygulanan tekniklerin sonuçlarına göre MAE değerleri metre cinsinden Şekil 4.6.'da ki grafikte verilmiştir.



Şekil 4.6. MAE Değerleri Grafiği

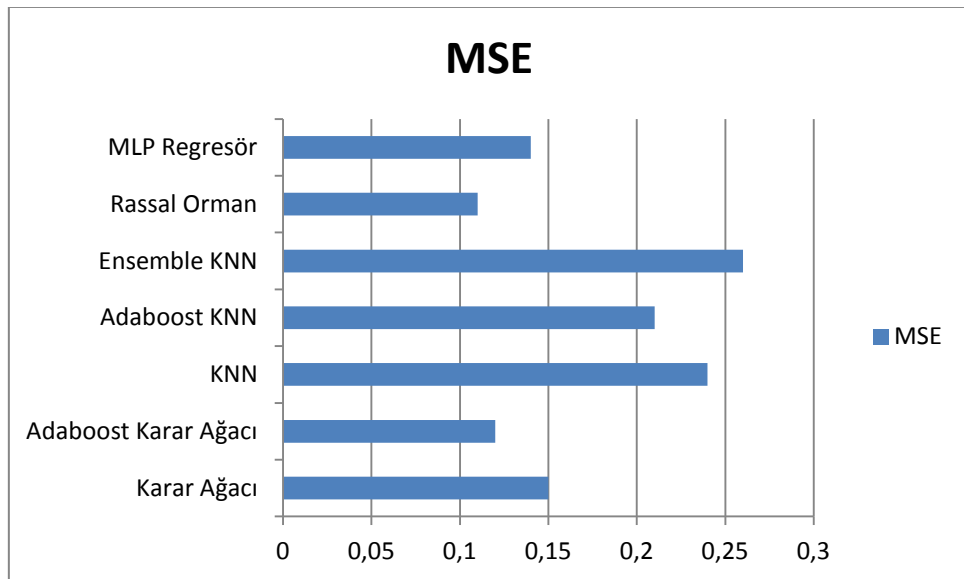
- Ortalama Kare Hata(Mean Square Error-MSE) bir regresyon çizgisinin bir dizi noktaya ne kadar yakın olduğunu ölçer. Bunu, noktalardan regresyon çizgisine kadar olan mesafeleri aldıktan sonra (bu mesafeler "hata"yı temsil eder.) karelerini hesaplayarak yapar. Formülü aşağıdaki gibidir [95].

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^l)^2 \quad (4.3)$$

y_i : Gerçek değer

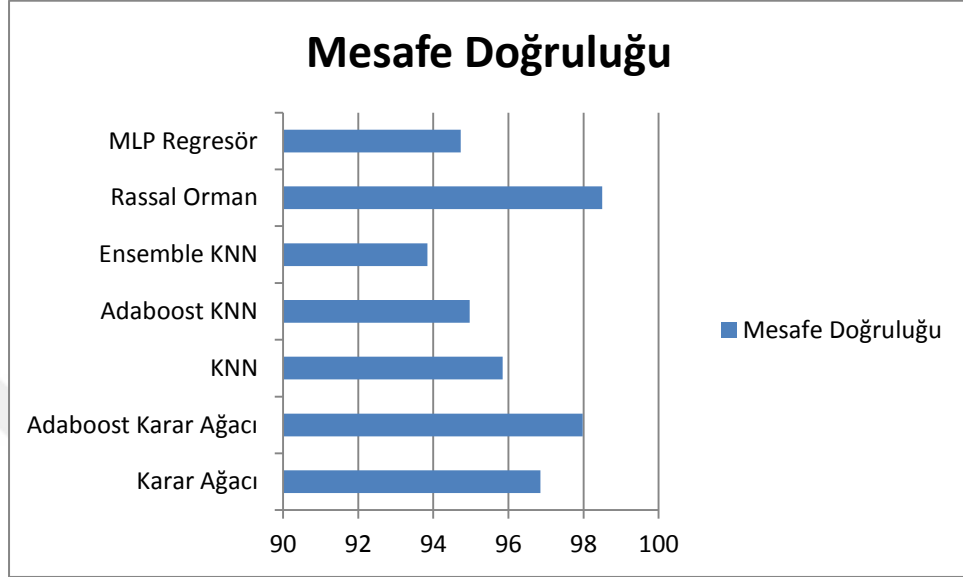
y_i^l : Tahnin edilen değer

Uygulanan tekniklerin sonuçlarına göre MSE değerleri yüzde olarak Şekil 4.7.'da ki grafikte verilmiştir.



Şekil 4.7. MSE Değerleri Grafiği

- Mesafe Doğruluğu(Distance Accuracy), gerçek değer ile tahmin edilen değer arasındaki fark alınır ve bu fark gerçek değere bölünerek 1'den çıkarılır. Bu değer genellikle yüzdesel olarak hesaplanır. Uygulanan tekniklerin sonuçlarına göre mesafe doğrulukları yüzde olarak Şekil 4.8.'da ki grafikte verilmiştir.



Şekil 4.8. Mesafe Doğruluk Değerleri Grafiği

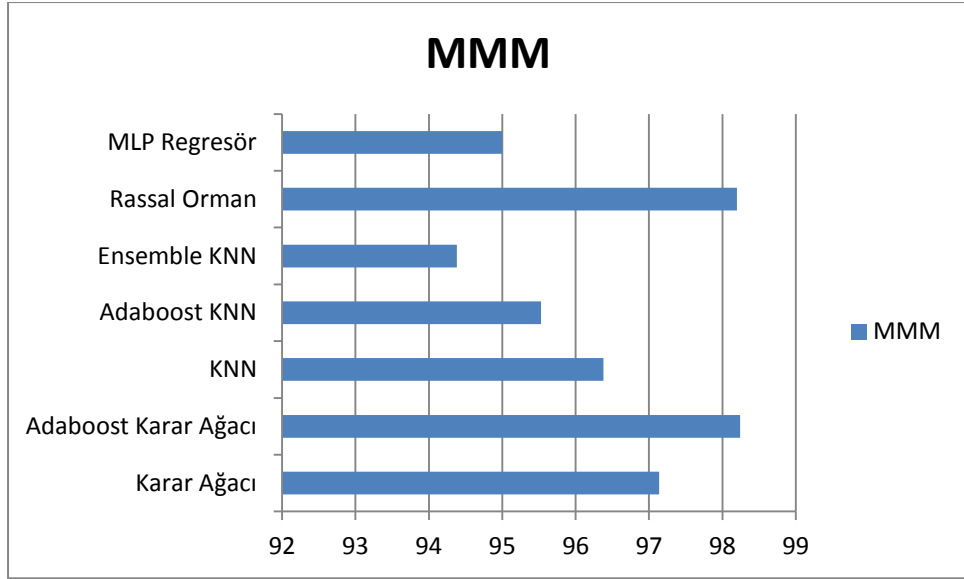
- MMM(Mean of Min over Max), tarafımızdan ilk kez uygulanan bir parametredir. Bu parametre ile test sonuçlarının doğruluk oranı aşağıdaki formül kullanılarak hesaplanmaktadır.

$$MMM = \frac{\sum_{i=1}^N \frac{\min(y_i, y_i^1)}{\max(y_i, y_i^1)}}{N} \quad (4.4)$$

y_i : Gerçek değer

y_i^1 : Tahmin edilen değer

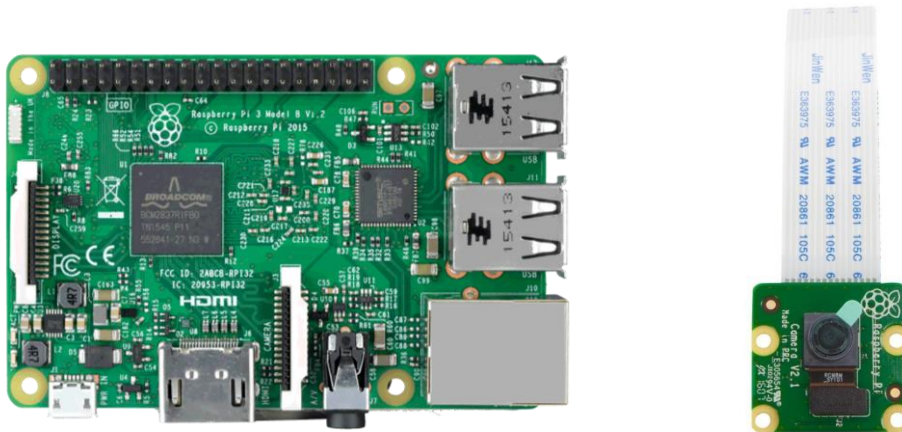
Uygulanan tekniklerin sonuçlarına göre MMM değerleri yüzde olarak Şekil 4.9.'da ki grafikte verilmiştir. MMM parametresinde en yüksek değeri Rassal Orman algoritmasının verdiği görülmektedir.



Şekil 4.9. MMM Değerleri Grafiği

4.2. Prototipin Yapısı

Tasarlanan sistem Raspberry Pi 3 Model B üzerinde gerçekleştirilerek prototipi oluşturulmuştur. Prototipin maliyetinin düşük ve taşınabilir olması için Raspberry Pi tercih edilmiştir. Raspberry Pi, bilgisayar monitörü veya TV'ye takılan ve standart bir klavye ve fare kullanan düşük maliyetli, kredi kartı boyutunda bir bilgisayardır. Her yaşta insanın bilgisayarları keşfetmesini ve Scratch ve Python gibi dillerde nasıl programlanabileceğini öğrenmesini sağlayan küçük bir cihazdır. Bir masaüstü bilgisayarın yapabileceği her şeyi, internette gezinmek, yüksek tanımlı video oynatmak, elektronik tablo hazırlamak gibi işlevleri gerçekleştirebilmektedir. Raspberry Pi 3 Model B üçüncü nesil Raspberry Pi'dir. Prototip için kullanılan V2 kamera modülü bir Sony IMX219 8 mega piksel sensöre sahiptir [96].



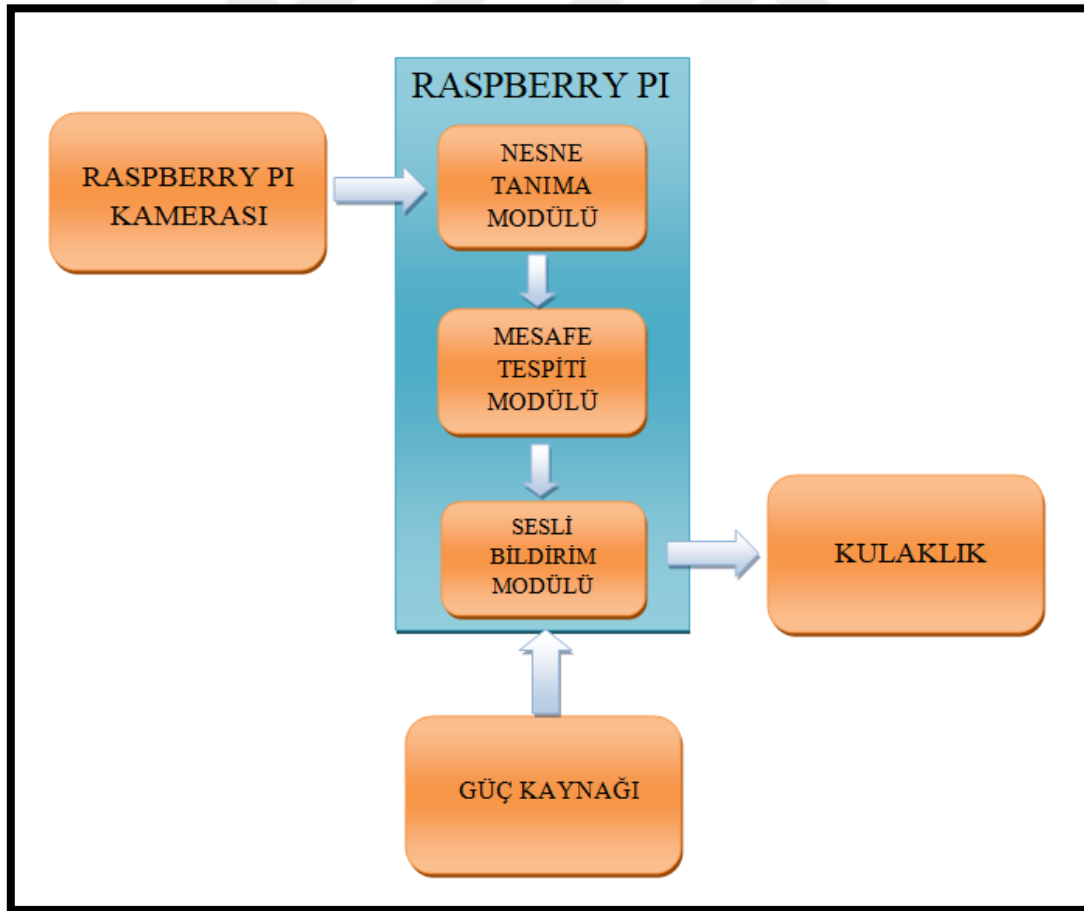
Şekil 4.10. Raspberry Pi 3 Model B ve Kamera Modülü[96]

Raspberry Pi 3 Model B'nin özellikleri tablo 4.1.'da verilmiştir.

Tablo 4.1. Raspberry Pi 3 Model B Özellikleri

Özellik	
İşlemci	Broadcom BCM2837
CPU Çekirdeği	Quadcore ARM Cortex-A53, 64 Bit
Saat Hızı	1.2 GHz
RAM	1 GB
GPU	400 MHz VideoCore IV ®
Ağ Bağlantısı	1x10/100 Ethernet (RJ45 Port)
Kablosuz Ağ Bağlantısı	802.11n wireless LAN(WiFi)/Bluetooth
USB Portları	4 x USB 2.0
GPIO'lar	2x20 Pinli Header
Güç Sarfiyatı	2.5 A
Uzunluk	85.6 mm
Genişlik	53.98 mm
Ağırlık	45 g

Sistem birkaç modül içermektedir ve her modül belirli işlevlerden sorumludur. Sistemin bileşenleri (modülleri) Şekil 4.11.'da gösterilmektedir.



Şekil 4.11. Sistemin Modülleri

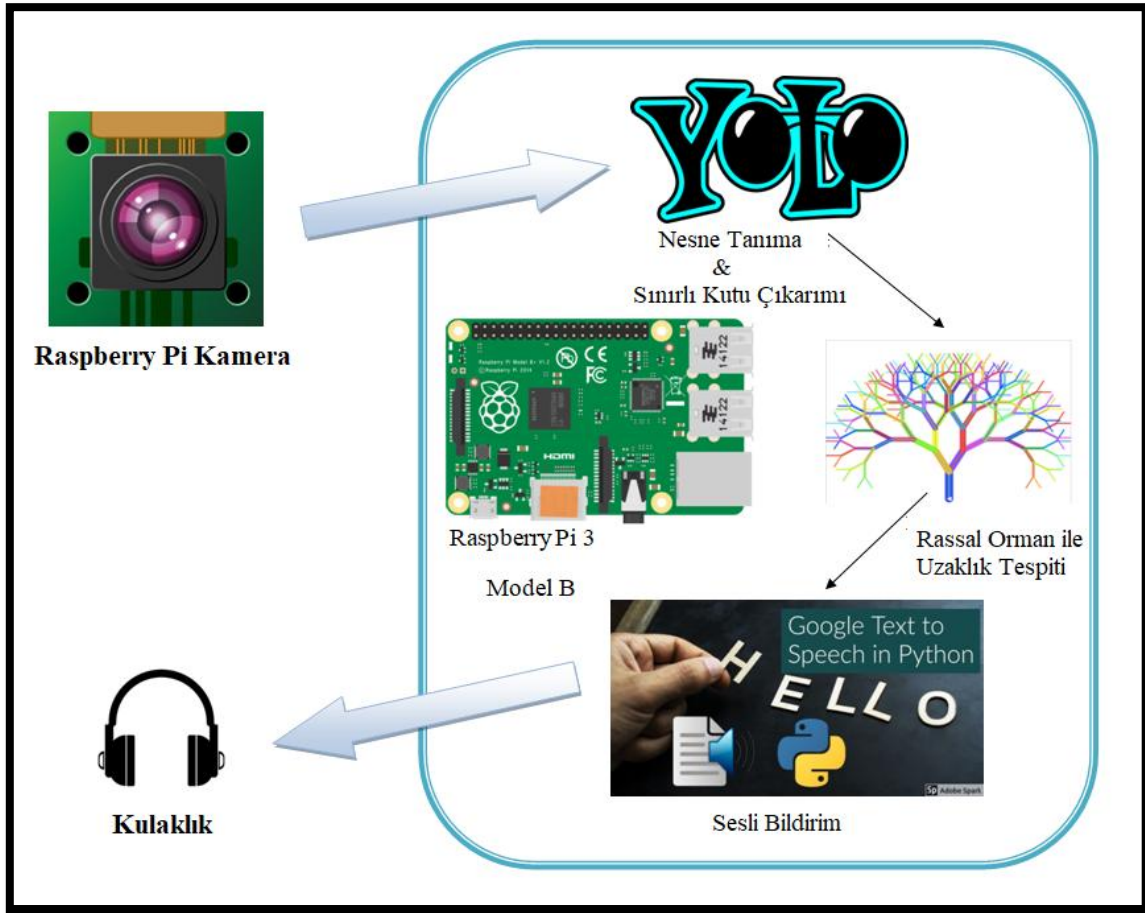
Prototip, yazılım ve donanım bileşenlerini içermektedir. Donanım bileşenleri:

1. Raspberry Pi 3 Model B
2. Raspberry Pi Kamera Modülü
3. Güç Kaynağı (5 V)
4. Kulaklık

Yazılım bileşenleri:

1. NNPACK Darknet ile YOLOv3 (Tiny) Gerçek Zamanlı Nesne Tanıma Sistemi
2. Rastgele Orman Regresyon Modeli
3. Google Metinden Konuşmaya Modülü

Şekil 4.12. prototipin veri akış şemasını göstermektedir.



Şekil 4.12. Veri Akış Şeması

Prototip oluşturulurken öncelikle, Raspberry Pi kamera modülünün ve Python ortamının kurulumu gerçekleştirilmiştir. Kurulum için gerekli olan komut Şekil 4.13.'da gösterilmektedir.

```
sudo apt-get install python-pip
```

Şekil 4.13. Python Ortamı Kurulum Komutu

“sudo raspi-config” komut kullanılarak Raspberry Pi’in ayarlar ekranına ulaşarak “P1/Camera” seçeneği aktif hale getirilir.

YOLO’nun Raspberry Pi üzerinde gerçek zamanlı olarak çalışabilmesi için YOLOv3-tiny versiyonu tercih edilmiştir. Bununla birlikte, sinir ağlarının ARM CPU’larda çalışması için bir hızlandırma kütüphanesi olan NNPACK Darknet kullanılmıştır. YOLO ve NNpack Darknet kütüphanesinin kurulumu aşağıdaki gibidir.

a. NNPack’in Kurulumu

Öncelikle Ninja’nın kurulumu gerekmektedir. Ninja hız odaklanan küçük bir yapı sistemidir. Diğer yapı sistemlerinden iki ana açıdan farklıdır: girdi dosyalarının daha üst düzey bir derleme sistemi tarafından üretilmesi için tasarlanmıştır ve yapıların olabildiğince hızlı çalışması için tasarlanmıştır [97]. Kurulum komutları Şekil 4.14. ve 4.15.’da gösterilmektedir.

```
sudo pip install --upgrade git+https://github.com/Maratyszczca/PeachPy
sudo pip install --upgrade git+https://github.com/Maratyszczca/confu
```

Şekil 4.14. Ninja Kurulum Komutları

```
git clone https://github.com/ninja-build/ninja.git
cd ninja
git checkout release
./configure.py --bootstrap
export NINJA_PATH=$PWD
cd
```

Şekil 4.15. Ninja Kurulum Komutları

NNPack’in kurulum komutları Şekil 4.16.’da gösterilmektedir.

```
git clone https://github.com/shizukachan/NNPACK
cd NNPACK
confu setup
python ./configure.py --backend auto
```

Şekil 4.16. NNPack Kurulum Komutları

Ninja ile NNPack’i oluşturmak için gerekli komut Şekil 4.17.’da gösterilmektedir.

```
$NINJA_PATH/ninja
```

Şekil 4.17. NNPack'i Oluşturmak İçin Gerekli Komut

b. NNPack Darknet'in Kurulumu

Kurulum komutları Şekil 4.18.'da gösterilmektedir.

```
cd
git clone -b yolov3 https://github.com/zxzhaixiang/darknet-nnpack
cd darknet-nnpack
git checkout yolov3
make
```

Şekil 4.18. Darknet Kurulum Komutları

YOLOv3-tiny versiyonunu kullanmak için gerekli komutlar Şekil 4.19.'da gösterilmektedir.

```
yolo_proc = Popen(["./darknet",
                  "detect",
                  "./cfg/yolov3-tiny.cfg",
                  "./yolov3-tiny.weights",
                  "-thresh", "0.1"],
                  stdin = PIPE, stdout = PIPE)
```

Şekil 4.19. YOLOv3-tiny Kullanma Komutları

Yüz tespiti için, WIDER FACE: A Face Detection Benchmark[] veri seti ile eğitilmiş olan YOLOv3 ağırlık dosyasını indirip model-weights/ dizinine yerleştirmek gerekmektedir. Daha sonra Şekil 4.20.'da yer alan komutu çalıştırarak yüz tespiti yapılabilmektedir. Yüz tespiti sonuçları Şekil 4.21.'da gösterilmektedir.

```
$ python yoloface.py --image samples/outside_000001.jpg --output-dir outputs/
```

Şekil 4.20. Yüz Tespiti Çalıştırma Komutu

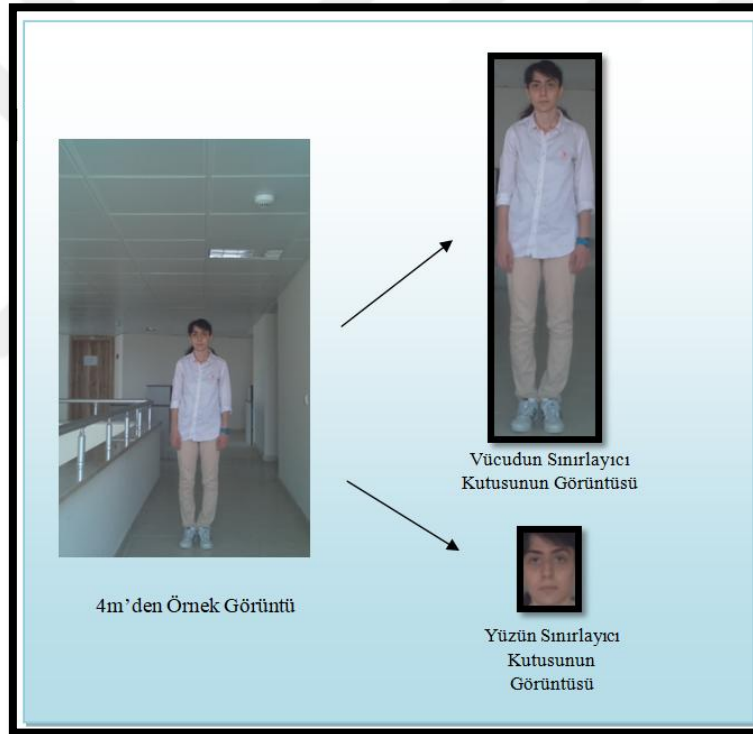


Şekil 4.21. YOLO ile Yüz Tespiti Sonuçları

Sistemde öncelikle, video taşınabilir bir Raspberry Pi kamera kullanılarak yakalanır. Ardından, nesne tanıma modülü, gerçek zamanlı nesne tanıma için YOLO'yu çalıştırmakta ve sınırlayıcı kutu boyutu çıkarma işlemine (insanlar için) ihtiyaç duyulmaktadır. YOLO görüntü içerisinde bulunduğu nesnelere ve yüzleri boyutlarına göre sınırlı kutu (bounding box) içerisine almaktadır. Bu kutular nesne kameradan uzaklaştıkça küçülmekte ve yakınlaştıkça da büyümektedir. Uzaklık tespiti için tarafımızdan ortaya konan teknik kişilerin yüz ve vücut boyutlarının kameradan olan uzaklıklarına göre değişimini temel aldığı için bu sınırlı kutuların yükseklik ve genişlik bilgileri YOLO kaynak kodu içerisine eklenen kod parçası ile alınarak sistemin Python programlama dili ile yazılmış olan algoritmasını beslemektedir. Bu sayede, kişilerin yüz ve vücut boyutlarındaki değişime göre uzaklık tespiti yapılabilmektedir. Sınırlayıcı kutunun boyutu, tespit edilen kişinin mesafesini tahmin etmek için mesafe tahmin modülünü beslemektedir. Algılanan nesnelere etiketi (nesnenin adı) ve tahmini mesafe sonuçları soldan sağa doğru bir "txt" dosyasına kaydedilmekte ve bu metin kayıtları gTTS modülü tarafından sese dönüştürülmektedir. Ses bildirimi, aşırı yüklenmeyi azaltmak için uyku fonksiyonu (5 saniye) kullanılarak zaman aralığında çalışmaktadır.

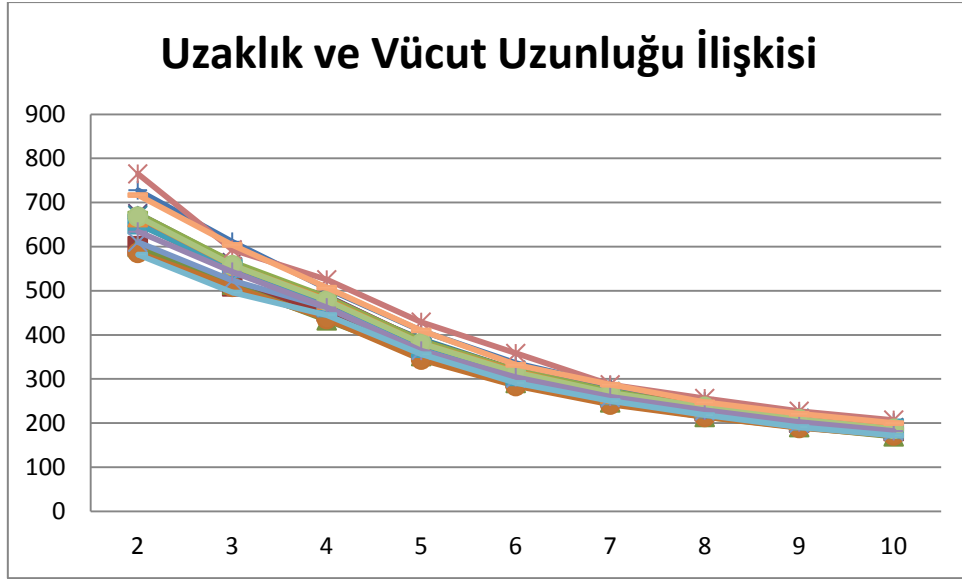
4.2.1. Mesafe Tespiti Sonuçları

Raspberry Pi kullanılarak oluşturulan prototipin istikrarlı ve güvenilir olması adına Raspberry Pi kamera modülü ile ikinci bir veri setini oluşturmak için 2 ile 10 metre arasındaki her bir mesafe için görüntüler toplanmıştır. Her bir kişi için toplamda 480x800 çözünürlükte dokuz görüntü elde edilmiştir. Kişilerin vücut ve yüz boyutları (yükseklik / genişlik) mesafenin değişimini analiz etmek için sınırlayıcı kutu boyutlarına göre çıkarılmıştır. Böylece, 9 vücut ve 9 yüz, toplam 18 görüntü, her kişi için veri setine kaydedilmiştir. Bu sayede toplamda 450 görüntüden oluşan bir veri kümesi oluşturulmuştur. Daha sonra görüntüler analiz edilerek gövde ve yüzlerin yüksekliği / genişliği (piksel cinsinden) bir Excel dosyasına kaydedilmiştir. Şekil 4.22.'de, 4 metre mesafeden vücut ve yüz boyutları hakkında bilgi edinmek için çıkarılan örnek bir resmi göstermektedir.



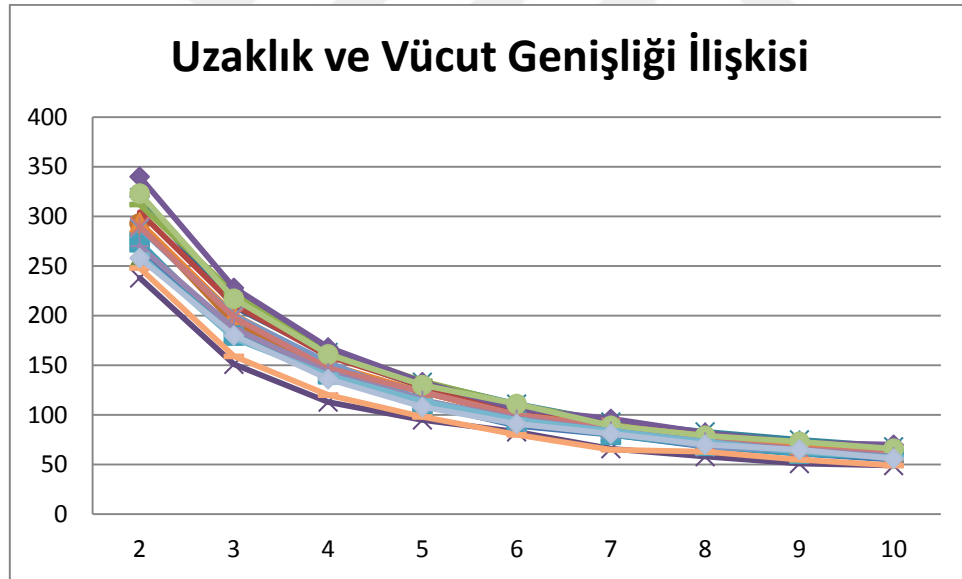
Şekil 4.22. 4 metre Mesafeden Vücut ve Yüz Boyutları

Prototip için oluşturulmuş olan ikinci veri setinde, vücut uzunluğu (Bh), vücut genişliği (Bw), yüz uzunluğu (Fh) ve yüz genişliği (Fw) olan vücut ve yüzün sınırlayıcı kutusu boyutları, eğitim sürecinde kullanılan özellik vektörünü oluşturmaktadır. Bu özellikler ile uzaklık arasındaki ilişki Şekil 4.23., 4.24., 4.25., 4.26.'da ki grafiklerde gösterilmiştir.



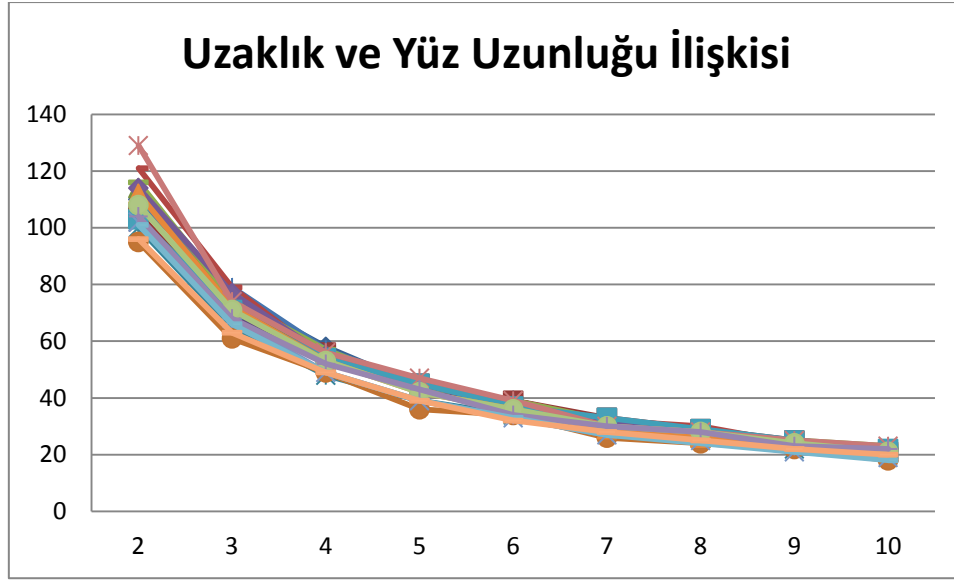
Şekil 4.23. Uzaklık ve Vücut Uzunluğu İlişkisi Grafiği

Şekil 4.23.'da vücut uzunluğunun uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi vücut uzunluğu uzaklık arttıkça azalmaktadır. Bu bağlamda kameradan uzaklaşan nesnenin boyunun gitgide kısılacacağı söylenebilir.



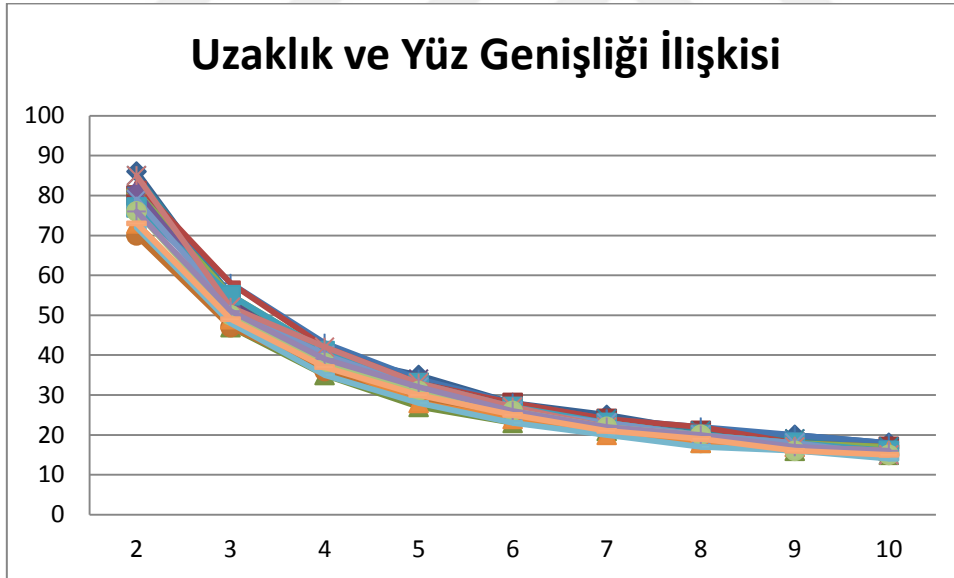
Şekil 4.24. Uzaklık ve Vücut Genişliği İlişkisi Grafiği

Şekil 4.24.'da vücut genişliğinin uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi vücut genişliği uzaklık arttıkça azalmaktadır. Bu bağlamda kameradan uzaklaşan nesnenin genişliğinin gitgide azalacağı söylenebilir.



Şekil 4.25. Uzaklık ve Yüz Uzunluğu İlişkisi Grafiği

Şekil 4.25.'da yüz uzunluğunun uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi yüz uzunluğu uzaklık arttıkça azalmaktadır. Burada özellik vektöründe kullanılan yüz uzunluğu diğer nesnelere için kullanılmayacağından, farklı özellik seçimi yapılması uygun olacaktır.

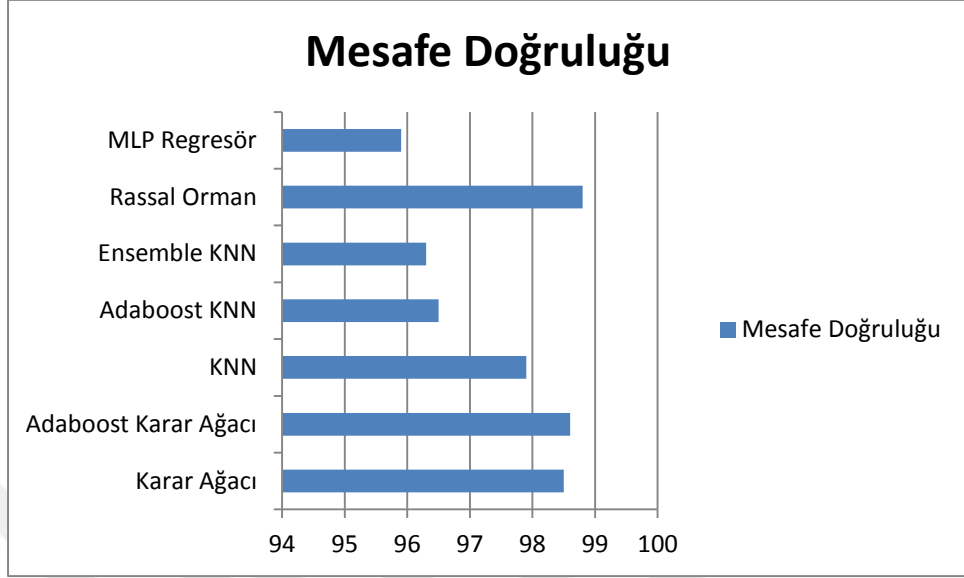


Şekil 4.26. Uzaklık ve Yüz Genişliği İlişkisi Grafiği

Şekil 4.26.'da yüz genişliğinin uzaklığa göre değişim grafiği verilmiştir. Grafikten de anlaşılacağı gibi yüz genişliği uzaklık arttıkça azalmaktadır.

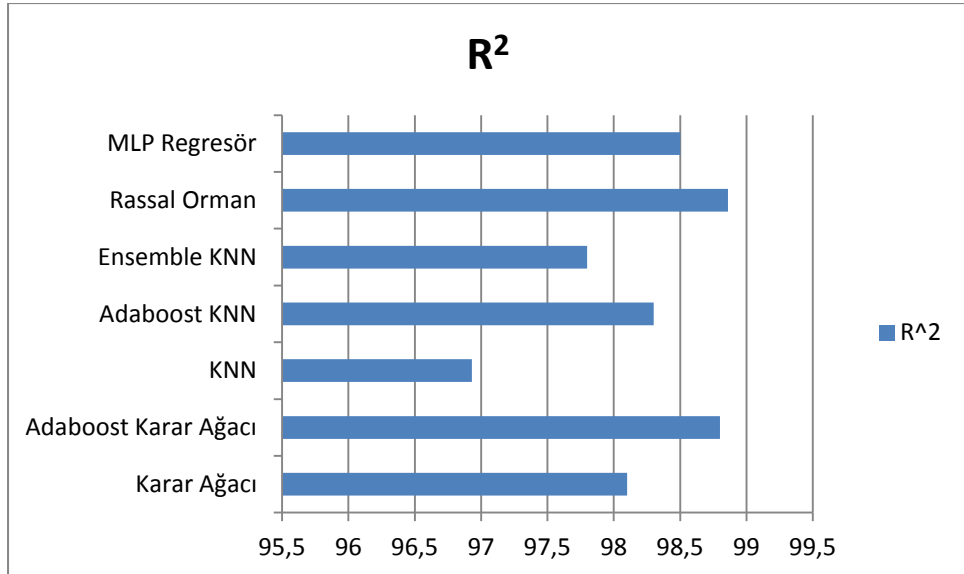
Özellikler (B_h , B_w , F_h , F_w) ve uzaklık arasındaki ilişkilerin doğrusal olmayan bir yapıya sahip olduğu gözlemlenmiş ve bu nedenle, mesafe tahmini için doğrusal olmayan regresyon modellerinden olan Rastal Orman modeli %98,8 sonucuyla en yüksek mesafe doğruluğunu ortaya koymuştur. Prototip üzerinde uzaklık tespiti için kullanılan regresyon tekniklerinden

elde edilen tüm sonuçlar farklı parametreler üzerinden Şekil 4.27., 4.28., 4.29., 4.30., 4.31.'da ki grafiklerde ayrıntılı olarak gösterilmektedir.



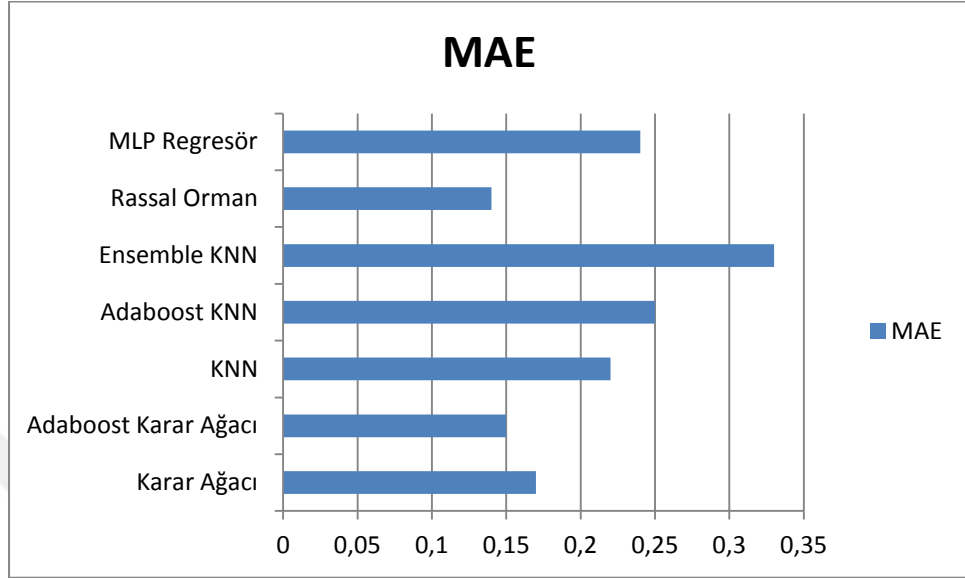
Şekil 4.27. Mesafe Doğruluk Değerleri Grafiği

Elde edilen sonuçlara göre Şekil 4.27.'da ki grafikte görüldüğü gibi mesafe doğruluğu parametresinde en yüksek değeri veren ilk iki algoritma %98,8 ile Rassal Orman ve %98,6 ile Adaboost Karar Ağacı algoritmaları olmuştur. En düşük değer ise %95,9 ile MLP algoritması olmuştur.



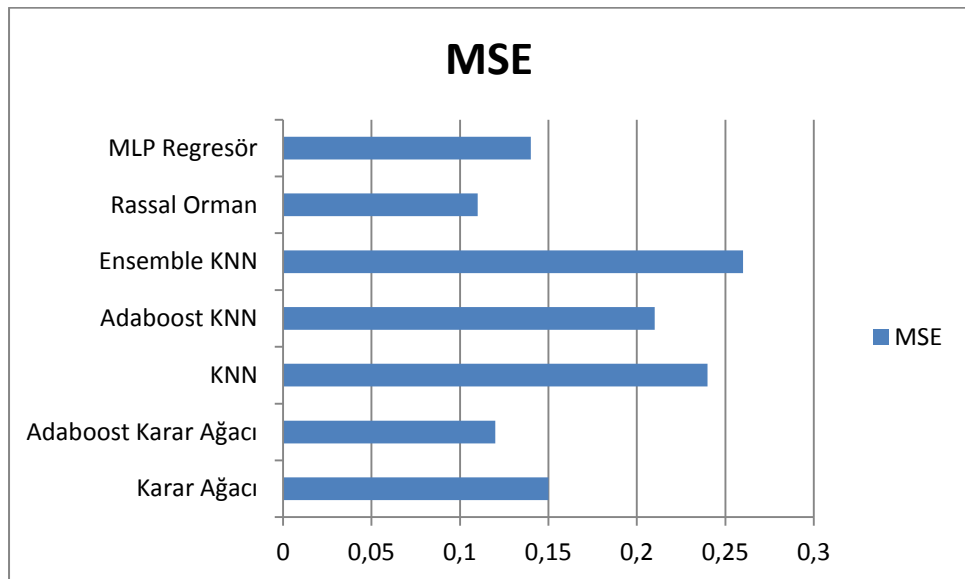
Şekil 4.28. R² Değerleri Grafiği

Elde edilen sonuçlara göre Şekil 4.28.'da ki grafikte görüldüğü gibi R^2 parametresinde en yüksek değeri veren ilk iki algoritma %98,86 ile Rassal Orman ve %98,8 ile Adaboost Karar Ağacı algoritmaları olmuştur. En düşük değer ise %97,8 ile Ensemble KNN algoritması olmuştur.



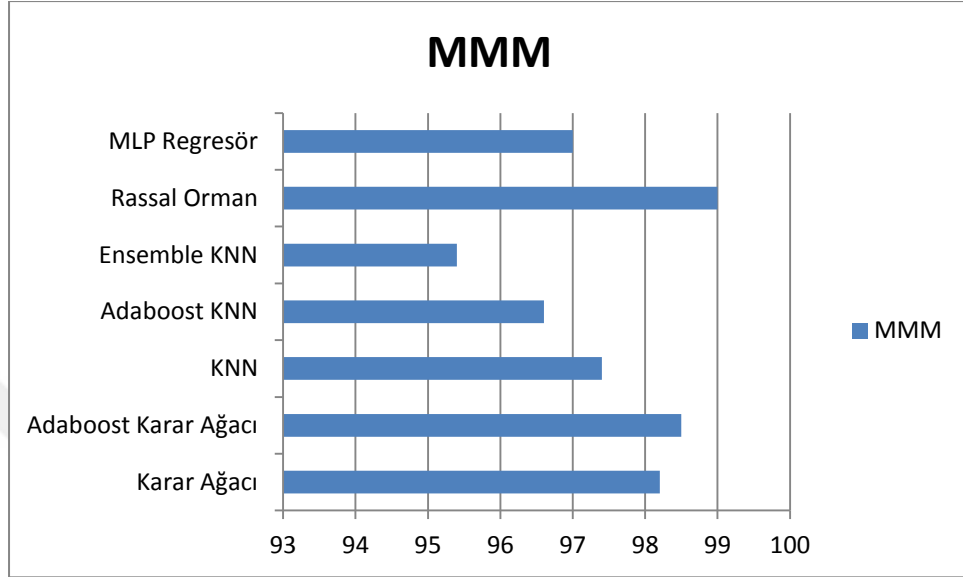
Şekil 4.29. MAE Değerleri Grafiği

Elde edilen sonuçlara göre Şekil 4.29.'da ki grafikte görüldüğü gibi MAE parametresinde en yüksek değeri veren ilk iki algoritma 0,14 metre hata ile Rassal Orman ve 0,15 hata ile Adaboost Karar Ağacı algoritmaları olmuştur. En düşük değer ise 0,33 metre hata ile Ensemble KNN algoritması olmuştur.



Şekil 4.30. MSE Değerleri Grafiği

Elde edilen sonuçlara göre Şekil 4.30.'da ki grafikte görüldüğü gibi MSE parametresinde en yüksek değeri veren ilk iki algoritma 0,11 metre hata ile Rassal Orman ve 0,12 hata ile Adaboost Karar Ağacı algoritmaları olmuştur. En düşük değer ise 0,26 metre hata ile Ensemble KNN algoritması olmuştur.



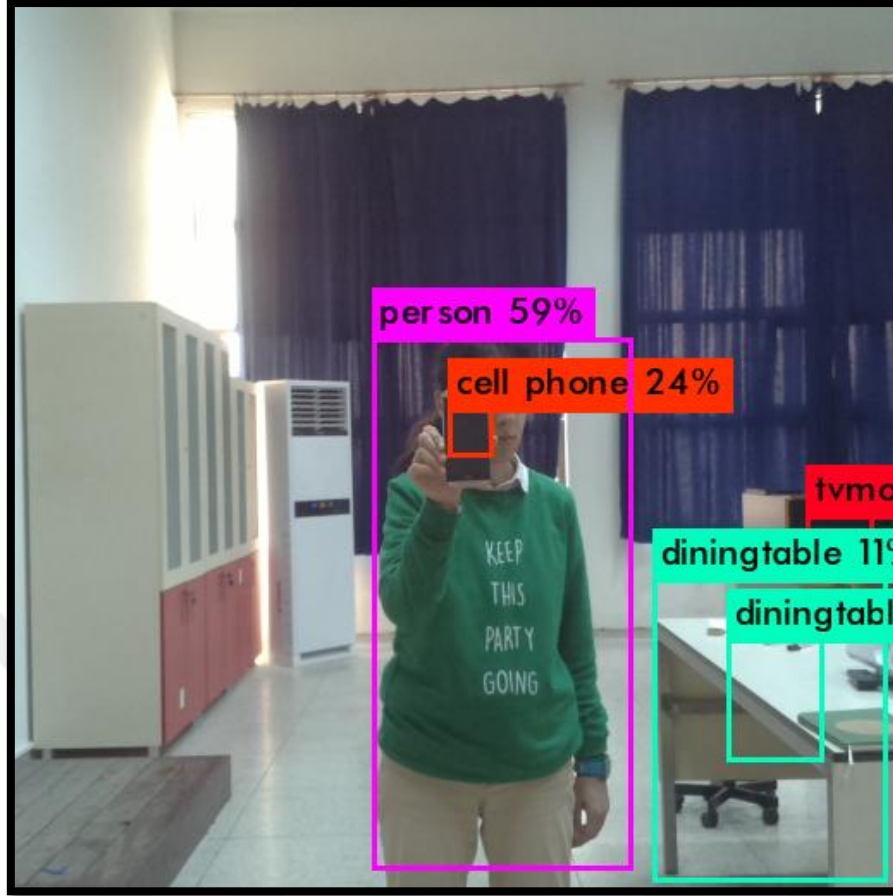
Şekil 4.31. MMM Değerleri Grafiği

Elde edilen sonuçlara göre Şekil 4.31.'da ki grafikte görüldüğü gibi MMM parametresinde en yüksek değeri veren ilk iki algoritma %99 ile Rassal Orman ve %98,5 ile Adaboost Karar Ağacı algoritmaları olmuştur. En düşük değer ise %95,4 ile Ensemble KNN algoritması olmuştur.

Sonuçlarda elde edilen tüm parametreler incelendiğinde en başarılı sonuçları veren algoritmanın Rassal Orman algoritması olduğu görülmüştür.

4.2.2. Gerçek Zamanlı Nesne Tanıma Sonuçları

Tasarlanan sistemde, nesne tanıma modülü olarak gerçek zamanlı nesne tanıma sistemi olan YOLO NNPACK Darknet kütüphanesi ile birlikte kullanılmıştır. Raspberry pi'nin düşük işleme kabiliyeti nedeniyle, YOLO'nun son sürümü olan v3'ün normal versiyonu Raspberry pi üzerinde çalışmamaktadır. Bu nedenle, kaynak kısıtlı platformlar için çok hafif bir model olan YOLOv3'ün tiny versiyonu kullanılmıştır. YOLOv3-tiny algoritması, sinir ağlarının ARM CPU'larda çalışması için bir hızlandırma kütüphanesi olan NNPACK Darknet'i kullanarak nesne tespiti için çalıştırılmıştır. YOLOv3 'ün tiny versiyonu NNPACK Darknet Kütüphanesi olmadan bir çerçeveyi yaklaşık olarak 32 saniyede işlerken, NNPACK Darknet kütüphanesi kullanılarak bu süre 2 saniyeye düşürülmüştür.



Şekil 4.32. YOLOv3 tiny ile Nesne Tespiti Anı

Şekil 4.32.'da Raspberry pi kamera modülü tarafından çekilen bir görüntünün YOLOv3 tiny versiyonu kullanılarak (NNPACK Darknet kütüphanesi ile) işlenmesiyle tespit edilen nesnelere gösterilmiştir.

5. SONUÇLAR ve ÖNERİLER

Bu çalışmada, görme engelli bireylerin açık ve kapalı ortamlarda karşılaştıkları nesne ve insanları tanımalarına yardımcı olan ve etraflarındaki insanların ne kadar uzakta olduklarını anlayabilmelerini sağlayan yardımcı bir sistem geliştirilerek prototipi yapılmıştır. Sistem temel olarak nesne tanıma ve mesafe tespiti yapabilmekte ve elde edilen sonuçları sesli bildirim modülü ile kullanıcılara sesli olarak iletebilmektedir. Mesafe tespiti modülü için öncelikle, 50 farklı kişiden alınan görüntüler analiz edilerek bir veri seti oluşturulmuştur. Bu veri seti Karar Ağacı, Adaboost Karar Ağacı, K En Yakın Komşu, Adaboost K En Yakın Komşu, Ensemble K En Yakın Komşu, Rassal Orman ve MLP Regresör olmak üzere 7 farklı regresyon algoritması kullanılarak eğitilip test edilmiştir. Elde edilen sonuçlar tasarlanan mesafe tespiti yönteminde kullanılan özelliklerin doğru seçildiğini ve bu yöntemin prototip üzerinde de uygulanabileceğini göstermiştir. Bu doğrultuda, sistemin mesafe tespiti modülünün prototip üzerinde de stabil ve güvenilir şekilde çalışması için Raspberry Pi 3 Model B ile Raspberry Pi kamera modülü kullanılarak ikinci bir veri seti oluşturulmuştur. Teorik olarak denenmiş olan regresyon algoritmaları bu veri seti ile Raspberry Pi üzerinde de yeniden denenmiş ve en yüksek doğruluk %98,8 oranı ile Rassal Orman algoritması kullanılarak elde edilmiştir. Nesne tanıma modülü için kullanılan gerçek zamanlı bir nesne tanıma sistemi olan YOLO'nun tiny V3 versiyonu, NNPack Darknet kütüphanesi ile kullanılmış ve bir çerçevede yer alan nesnelere 2 saniyede tespit edilmiştir.

Görme engelli bireyler için geliştirilen yardımcı teknolojilerin yaygın olarak kullanılmamalarının temel sebebi, bu sistemlerin maliyetlerinin yüksek, zor taşınan ve zor kullanılan sistemler olmalarıdır. Bu sebeple, geliştirilen sistemin düşük maliyette, taşınabilir ve kolay kullanıma sahip olması bugüne kadar geliştirilmiş olan sistemler arasında avantajlarının oldukça yüksek olduğunu göstermektedir. Yapılan çalışmanın geliştirilmesi ve daha güvenilir olması amacıyla, veri setindeki kayıt sayısının artırılması, kullanıcıların kolay açıp kapaması için sesli arayüz oluşturulması, sesli bildirimlerin daha istikrarlı hale getirilmesi gibi yenilikler üzerinde çalışılması planlanmaktadır. Görme engelli bireylerin günlük yaşamlarını kolaylaştırarak bağımsız hareket edebilmelerini sağlayacak olan yardımcı sistemleri geliştirme konusunda daha fazla çalışma yapılmasının, bu sistemleri daha güvenilir, ucuz ve kullanıcı dostu sistemler haline getirebileceğini göstermektedir.

Bu tez çalışmasında geliştirilen sistemin, daha güvenilir ve istikrarlı çalışması adına veri setinin genişletilmesi düşünülmektedir. Veri setindeki kayıtlar çeşitlenip sayısal olarak arttıkça uzaklık tespitindeki tahmin oranı artacak ve sistem daha güvenilir olacaktır. Aynı zamanda, cihazın taşınabilirliğini arttırmak adına cihazın şarj edilebilir versiyonunu oluşturmak adına çalışmalar yapılmaktadır. Sesli bildirim özelliğini daha istikrarlı kılmak adına optimizasyon

çalışmaları gelecek çalışmalar arasında yer almaktadır. Yeni eklenen özellikler ve düzeltmeler sayesinde, geliştirilen cihaz görme engelli bireylerin kullanımına daha uygun hale gelerek ilerleyen zamanlarda beyaz baston kadar yaygın kullanılan yardımcı bir cihaz haline gelebilecektir.

Bugüne kadar yapılan bu çalışmaya benzer çalışmalarda oluşturulan veri setlerinin erişime açılmadığı görülmüştür. Bu sebeple, çalışma kapsamında oluşturulan veri seti yapılacak olan benzer çalışmalarda kullanılabilmesi adına <https://github.com/sosodum/Dataset-for-Distance-Estimation.git> adresinde erişime açılmıştır.



KAYNAKLAR

- [1]. World Health Organization, Blindness and vision impairment, <http://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment> (20.06.2019).
- [2]. Engelsiz Erişim Derneği, GÖRME ÖZÜRLÜLER İÇİN BASTON KULLANMANIN ÖNEMİ VE TARİHÇESİ, <http://www.engelsizerisim.com/detay/gorme-ozurluler-icin-baston-kullanmanin-onemi-ve-tarihcesi/> (20.06.2019).
- [3]. Khan S., Rahmani H., Shah S. A. A., Bennamoun, M., *A Guide to Convolutional Neural Networks for Computer Vision*. 1st ed.; Morgan & Claypool Publishers: United States, 2018; p10-16.
- [4]. TapTapSee, About TapTapSee, <https://taptapseeapp.com/> (17.07.2019).
- [5]. Blind Help Project, Text Detective, <https://blindhelp.net/software/text-detective> (17.07.2019).
- [6]. Microsoft.com, Seeing AI | Talking camera app for those with a visual impairment. <https://www.microsoft.com/en-us/seeing-ai> (20.07.2019).
- [7]. Albraheem, L., AlDosari, R., AlKathiri, S., AlMotiry, H., Abahussain, H., AlHammad, L. and Alshehri, M. *Third Eye: An Eye for the Blind to Identify Objects Using Human-Powered Technology*. 2015 International Conference on Cloud Computing (ICCC), Saudi Arabia, 26-29 April 2015.
- [8]. vOICe, The vOICe Next Generation Human Technology, <https://www.seeingwithsound.com/about.htm> (17.07.2019).
- [9]. Fukushima, K. . N., A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position., *Biological Cybernetics* 1980, 36, (4), 193–202.
- [10]. Hubel, D. H.; Wiesel, T. N.; Receptive fields and functional architecture of monkey striate cortex, *Journal of Physiol.* 1968, 195, (1), 215–243.
- [11]. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner P., Gradient-based learning applied to document recognition, *Proc. IEEE* 1998, 86, (11), 2278–2324.
- [12]. LeCun, Y.; Bengio Y.; Hinton, G., Deep learning, *Nature*. 2015, 521, 436–444.
- [13]. Jiang, R., Lin, Q. and Qu, S, Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio, http://cs231n.stanford.edu/reports/2016/pdfs/218_Report.pdf (17.07.2019).
- [14]. Wikipedia, Kinect, <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvS2luZWNO> (17.07.2019).
- [15]. GoPro, Cameras, <https://gopro.com/en/tr/> (17.07.2019).
- [16]. Unity, Unity for All. <https://unity.com/> (17.07.2019).

- [17]. Girshick, R.; Donahue, J.; Darrell, T. and Malik, J.. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2016, 38,(1),142-158.
- [18]. Girshick, R.. *Fast R-CNN*. IEEE International Conference on Computer Vision (ICCV), Chile, 71-13 December 2015, pp.1440-1448.
- [19]. Ren, S.; He, K., Girshick; R.; Sun, J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2017, 39,(6),1137-1149.
- [20]. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A..*You Only Look Once: Unified, Real-Time Object Detection*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, 27-30 June 2016, pp.779-788.
- [21]. Mocanu, B., Tapu, R. and Zaharia, T.. *Seeing Without Sight — An Automatic Cognition System Dedicated to Blind and Visually Impaired People*. IEEE International Conference on Computer Vision Workshops (ICCVW), Italy, 22-29 October 2017, pp.1452-1459.
- [22].Votchallenge.net, VOT2016 Challenge | Dataset, <http://www.votchallenge.net/vot2016/dataset.html> (17.07.2019).
- [23]. D. Held, S. Thrun, and S. Savarese, Learning to track at 100 fps with deep regression Networks,https://www.researchgate.net/publication/301875917_Learning_to_Track_at_100_FPS_with_Deep_Regression_Networks (17.07.2019).
- [24]. Eckert, M., Blex, M. and Friedrich, C.. *Object Detection Featuring 3D Audio Localization for Microsoft HoloLens - A Deep Learning based Sensor Substitution Approach for the Blind*. Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies, Portugal, 19-21 January 2018, pp.555-561.
- [25].HoloToolkit,HoloToolkit, <http://www.webcitation.org/6u0tUB8dz> (17.07.2019).
- [26]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D, Erhan, D., Vanhoucke, V., Rabinovich, A., *Going deeper with convolutions*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 15), USA, 7-12 June 2015, pp.1-9
- [27]. Krizhevsky, A., Sutskever, I., Hinton, G. E., *Imagenet classification with deep convolutional neural Networks*. Advances in neural information processing systems, 2012, pp. 1097–1105.
- [28]. Monteiro, J., Aires, J. P., Granada, R., Barros, R. C., Meneguzzi, F., *Virtual guide dog: An application to support visually-impaired people through deep convolutional neural Networks*. International Joint Conference on Neural Networks (IJCNN), USA, 14-19 May 2017, pp.2267-2274.
- [29]. Iwashita, Y., Takamine, A., Kurazume, R., Ryoo, M. S., *First-person animal activity recognition from egocentric videos*. International Conference on Pattern Recognition (ICPR), Sweden, 24-28 August 2014, pp.4310-4315.

- [30]. L. Key R.-G.; Busnel, J.F., Air sonars with acoustic display of spatial information, *Animal Sonar Systems*. SpringerLink **1980**, 769–816.
- [31] Ali, A. J.; Sankar A .H., Artificial guidance system for visually impaired people, *Bonfring Int. J. Man Mach. Interface* **2012**, 2 ,(1) , 10–12.
- [32]. Ross, D., Lightman, A., *Talking braille: a wireless ubiquitous computing network for orientation and wayfinding*, Seventh International ACM SIGACCESS Conference on Computers and Accessibility, 2005, pp. 98–105 .
- [33]. Yelamarthi, K., Haas, D., Nielsen, D., Motherse, S., *RFID and GPS integrated navigation system for the visually impaired*, 53rd IEEE International Midwest Symposium on Circuits and Systems, 2010, pp. 1149–1152.
- [34]. Márkus, N., Arató, A., Juhász, Z., Bognár, G., Késmárki L., *MOST-NNG: an accessible GPS navigation application integrated into the Mobile Slate Talker (MOST) for the blind*, Computers Helping People with Special Needs, Lect. Notes Computer Science, 2010, pp. 247–254.
- [35]. Bessho, M., Kobayashi, S., Koshizuka, N., Sakamura, K., *Assisting mobility of the disabled using space-identifying ubiquitous infrastructure*, Tenth International ACM SIGACCESS Conference on Computer and Accessibility, 2008, pp. 283–284.
- [36] Martinez-Sala, A., Losilla, F., Sánchez-Aarnoutse, J., García-Haro, J., Design, Implementation and Evaluation of an Indoor Navigation System for Visually Impaired People. *Sensors* **2015**,15, (12), 32168-32187.
- [37]. Ulrich, I., Borenstein, J., The GuideCane-applying mobile robot technologies to assist the visually impaired. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 2001,31, (2), 131-136.
- [38]. Wahab, A., Helmy, M., Talib, A.A., Kadir, H.A., Johari, A., Noraziah, A. Sidek, R.M., Mutalib, A .A ., SmartCane: assistive cane for visually-impaired people, *Int. J. Comput. Sci* **2011**, 8, (4), 21–27.
- [39]. Penrod, W., Corbett, M.D., Blasch, B., A master trainer class for professionals in teaching the UltraCane electronic travel device, *J. Visual Impairment Blindness* 2005, 99, (11) ,711–714.
- [40]. Hoyle, B., Waters, D., *Mobility at: the batcane (ultracane)*, Assistive Technology for Visually Impaired and Blind People, London, 2008, pp. 209–229.
- [41]. Villamizar, L.H., Gualdrón, M., Gonzalez, F., Aceros, J., Rizzo-Sierra C.V., *A necklace sonar with adjustable scope range for assisting the visually impaired*, 35th Annual International Conference IEEE Engineering Medical Biology Society, 2013, pp. 1450–1453.
- [42]. Akita, L., Komatsu, T., Ito, K., Ono, T., Okamoto, M., CyARM: haptic sensing device for spatial localization on basis of exploration by arms, *Adv. Human-Comp. Interact* **2009**, 1–6.
- [43]. Tapu, R., Mocanu, B., Zaharia, T.B., ALICE: a smartphone assistant used to increase the mobility of visual impaired people, *J. Ambient Intell. Smart Environ* **2015**, 7 , (5), 659–678.

- [44]. Kulyukin V., Gharpure C., Nicholson J., *RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Kanada 2005, pp. 2845–2850.
- [45]. López-De-Ipiña, D.; Lorido, T.; López, U., Indoor Navigation and Product Recognition for Blind People Assisted Shopping. *Ambient Assisted Living Lecture Notes in Computer Science* **2011**,33-40.
- [46]. Hakobyan, L., Lumsden, J., O’Sullivan, D., & Bartlett, H. , Mobile assistive technologies for the visually impaired. *Survey of Ophthalmology* **2013**,58,(6), 513-528.
- [47]. Nicholson, J., Kulyukin, V., & Coster, D. , ShopTalk: Independent Blind Shopping Through Verbal Route Directions and Barcode Scans. *The Open Rehabilitation Journal* **2009**,2,(1), 11-23.
- [48]. Croce D. , Giarré L., Rosa F.G., Montana E., Tinnirello I., *Enhancing tracking performance in a smartphone-based navigation system for visually impaired people*, 24th Mediterranean Conference on Control and Automation (MED), Athens 2016, pp. 1355–1360.
- [49]. Manduchi R., *Mobile vision as assistive technology for the blind: An experimental study*, 13th International Conference on Computers Helping People with Special Needs, 2012, pp. 9–16.
- [50]. Tapu R., Mocanu B. , Bursuc A., Zaharia T., *A smartphone-based obstacle detection and classification system for assisting visually impaired people*, IEEE International Conference on Computer Vision Work- shops, Sydney 2013, pp. 444–451.
- [51]. Mocanu B. , Tapu R. , Zaharia T., When ultrasonic sensors and computer vision join forces for efficient obstacle detection and recognition, *Sensors* **2016** 16, 1807.
- [52]. Lucas B., Kanade T., *An iterative technique of image registration and its application to stereo*, IJCAI’81 7th International Joint Conference on Artificial Intelligence, USA 1981, pp. 674–679.
- [53]. Lee J.J., Kim G., *Robust estimation of camera homography using fuzzy RANSAC*, International Conference on Computational Science and its Applications, Kuala Lumpur, Malaysia 2007, pp. 992–1002.
- [54]. Learnopencv, Histogram of Oriented Gradients. <https://www.learnopencv.com/histogram-of-oriented-gradients/> (17.07.2019)
- [55]. Tian Y.L., Yang X., Yi C., Arditi A., Toward a computer vision-based way finding aid for blind persons to access unfamiliar indoor environments, *Mach. Vis. Appl.* **2013**, 24 , 521–535.
- [56]. Yang X., Yuan S., Tian Y., Assistive clothing pattern recognition for visually impaired people, *IEEE Trans. Human-Mach. Syst.* **2014**, 44,(2) ,234–243.
- [57]. Coughlan M., Shen , Crosswatch: a system for providing guidance to visually impaired travelers at traffic intersections, *J. Assist Technol.* **2013**, 7 ,(2) , 131–142.
- [58]. Mascetti S., Ahmetovic D., Gerino A., Bernareggi C., Busso M., Rizzi A., *Supporting pedestrians with visual impairment during road crossing: a mobile application for traffic lights*

detection, 15th International Conference On Computers Helping People With Special Needs, Springer International Publishing, Cham 2016, pp. 198–201.

[59]. Mascetti S., Ahmetovic D., Gerino A., Bernareggi C., Busso M., Rizzi A., Robust traffic lights detection on mobile devices for pedestrians with visual impairment, *Comput. Vision Image Underst.* **2016**,148 ,123–135.

[60]. Kulyukin V.A ., Kutiyawala A ., *Demo: ShopMobile II: Eyes-free supermarket grocery shopping for visually impaired mobile phone users*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, „San Francisco 2010, pp. 31–32.

[61]. Hu F., Zhu Z., Zhang J., *Mobile panoramic vision for assisting the blind via indexing and localization*, 2nd Workshop Assistive Computer Vision Robotics Conjunction (ECCV), 2014, pp. 600–614.

[62]. Hasanuzzaman F.M., Yang X., Tian Y., Robust and effective component-based banknote recognition for the blind, *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2012**, 42, 1021–1030.

[63]. Grijalva F., Rodriguez J.C., Larco J., Orozco L., *Smartphone recognition of the U.S. banknotes' denomination for visually impaired people*, IEEE ANDESCON Conference, Colombia 2010, pp. 1–6.

[64] Manikandan K., Sumithra T., Currency recognition in mobile application for visually challenged, *Discovery* **2015**, 30 ,245–248.

[65]. Springer K., Subramanian P., Turton T., Australian banknotes: Assisting people with vision impairment, *Bull. Reserve Bank* **2015**, 1–12.

[66]. Mancini, A., Frontoni, E., & Zingaretti, P., Mechatronic System to Help Visually Impaired Users During Walking and Running. *IEEE Transactions on Intelligent Transportation Systems* **2018**,19,(2), 649-660.

[67]. Cloix S., Weiss V., Bologna G., Pun T., Hasler D., *Obstacle and planar object detection using sparse 3D information for a smart walker*, International Conference on Computer Vision Theory and Applications (VISAPP), Portugal 2014, pp. 292–298.

[68]. Everding L., Walger L., Ghaderi V.S., Conradt J., *A mobility device for the blind with improved vertical resolution using dynamic vision sensors*, IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), Munich 2016, pp. 1–5.

[69]. Neto L.B., Grijalva F., Maïke V., César Martini L., Florencio D., Baranauskas M.C., Anderson R., Siome G., A kinect-based wearable face recognition system to aid visually impaired users, *IEEE Trans. Human-Mach. Syst.* **2017**, 47, 52–64.

[70]. Li B., Munoz J.P, Rong X., Xiao J., Tian Y., Arditi A., *ISANA: wearable context-aware indoor assistive navigation with obstacle avoidance for the blind*, ECCV - Workshops, 9914, 2016, pp. 448–462.

[71].Raspberry Pi, Products, <https://www.raspberrypi.org/products/> (17.07.2019).

- [72]. Raspberry Pi, Camera, <https://www.raspberrypi.org/products/> (17.07.2019).
- [73]. GitHub, Darknet with NNPACK, <https://github.com/shizukachan/darknet-nnpack> (17.07.2019).
- [74]. Google Cloud, AI&Machine Learning Products, <https://cloud.google.com/text-to-speech/> (17.07.2019).
- [75]. Khan, S. *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan & Claypool, 2018; p10-16.
- [76]. Y. Le Cun et al., Handwritten digit recognition: applications of neural network chips and automatic learning, *IEEE Commun. Mag.*, 27, (11), 41–46.
- [77]. The PASCAL Visual Object Classes, The PASCAL VOC Project, <http://host.robots.ox.ac.uk/pascal/VOC/> (17.07.2019).
- [78]. GitHub, NNPACK, <https://github.com/Maratyszczka/NNPACK> (17.07.2019).
- [79]. GeeksforGeeks, Regression and Classification | Supervised Machine Learning, 17 Temmuz 2019 tarihinde <https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/> (17.07.2019).
- [80]. S.Sayad, Regression, https://www.saedsayad.com/decision_tree_reg.htm (17.07.2019).
- [81]. Towards Data Science, Regression in Machine Learning, <https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec> (17.07.2019).
- [82]. Towards Data Science , Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained, <https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725> (17.07.2019).
- [83]. E.Uzun , Decision Tree- Regression, https://www.e-adys.com/makine_ogrenmesi/decision-tree-regression-regresyon/ (17.07.2019).
- [84]. Lim C. P, Goh W. Y. 2007. The Application of an Ensemble of Boosted Elman Networks to Time Series Prediction: A Benchmark Study. Malaysia: International Journal of Information and Mathematical Sciences 3:2 University of Science Malaysia.
- [85]. Lusia D. A, Suhartono. 2013. Ensemble Method Based on Two Level ARIMAXFFNN for Rainfall Forecasting in Indonesia. Indonesia: Institut Teknologi Sepuluh Nopember, Department of Statistics, Surabaya 60111.
- [86]. Chitra A, Uma S. 2010. An Ensemble Model of Multiple Classifiers for Time Series Prediction. *International Journal of Computer Theory and Engineering* 2010, 2 ,(3), 454 - 458.
- [87]. Analyticsvidhya, K nearest neighbor introduction to regression python, <https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/> (01.07.2019).

- [88]. Sinta, D., Wijayanto, H., & Sartono, B., Ensemble K-nearest neighbors method to predict rice price in Indonesia. *Applied Mathematical Sciences* **2014**, *8*, 7993-8005.
- [89].Scikit-learn, Neural Network Models (supervised), https://scikit-learn.org/stable/modules/neural_networks_supervised.html (01.07.2019).
- [90]. Ho, T. K. , *Random Decision Forests* . 3rd International Conference on Document Analysis and Recognition, Montreal QC 14–16 August 1995.
- [91]Wikipedia, Random Forest. <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUmFuZG9tX2ZvcmlvZdA> (17.07.2019).
- [92].Wikipedia, Google Text-to-Speech,<http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvR29vZ2xlX1RleHQtdG8tU3BlZWNo> (17.07.2019).
- [93].Wikipedia, Coefficient of determination. <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvQ29lZmZpY2llbnRfb2ZfZGV0ZXJtaW5hdGlvbg> (17.07.2019).
- [94].Wikipedia, Mean absolute error, <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTWVhbl9hYnNvbHV0ZV9lcnJvcg> (01.07.2019)
- [95].Wikipedia, Mean square error, <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTWVhbl9zcXVhcmVkaX2Vycm9y> (01.07.2019).
- [96] Raspberry pi, What is Raspberry pi?. <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/> (01.07.2019).
- [97] Ninja-Build ,Ninja. <https://ninja-build.org/>(17.07.2019).

ÖZGEÇMİŞ

Adı ve Soyadı : Sonay DUMAN

Doğum Tarihi : 13.07.1992

E-mail : sonayduman13@gmail.com

Öğrenim Durumu :

Derece	Bölüm/Program	Üniversite	Yıl
Lisans	Bilgisayar Mühendisliği	Toros Üniversitesi	2011-2016
Yüksek Lisans	Bilgisayar Mühendisliği	Mersin Üniversitesi	2017-2019

Görevler :

Görev Ünvanı	Görev Yeri	Yıl
Kurucu Ortak/Bilgisayar Mühendisi	Küp AR-GE ve Yazılım Ltd. Şti.	18/05/2015-2018
Bilgisayar Mühendisi	ACTED – UNHCR – MESEDER	16/10/2017-04/12/2017
Bilgisayar Mühendisi	Urban Express Lojistik ve Yazılım Geliştirme Ltd. Şti.	10/06/2017-09/09/2017

ESERLER (Makaleler ve Bildiriler)

1. DUMAN SONAY, AVAROĞLU ERDİNÇ, DUMAN FIRAT, ELEWI ABDULLAH; Smart Cinema Seat Finding System Using Interaction Design with Arduino; Conference: International Conference on Computer Science and Engineering (UBMK'18), Sarajevo, 2018
2. DUMAN SONAY, ELEWI ABDULLAH, YETGİN ZEKİ; A survey of Assistive Technologies For Visually Impaired Individuals Developed Using Convolutional Neural Network; Conference: International Engineering and Natural Sciences Conference (IENSC 2018), Diyarbakır, Nov.2018
3. DUMAN SONAY, ELEWI ABDULLAH, YETGİN ZEKİ; Design and Implementation of an Embedded Real-Time System for Guiding Visually Impaired Individuals; Conference: International Conference on Artificial Intelligence and Data Processing (IDAP 2019), Malatya, 2019