

**T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**YÜKSEK LİSANS TEZİ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ BİLİM DALI**

**SAYISAL GÖRÜNTÜ İŞLEME VE YAPAY ZEKA YÖNTEMİ İLE
SON KULLANICI ODAKLI ENDÜSTRİYEL ÜRÜN TASARIMI**

Atahan UYANIK

**Danışman
Doç. Dr. Ömer AYDIN**

MANİSA-2024

Atahan
UYANIK

SAYISAL GÖRÜNTÜ İŞLEME VE YAPAY ZEKA YÖNTEMİ İLE SON
KULLANICI ODAKLI ENDÜSTRİYEL ÜRÜN TASARIMI

2024

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik ve Elektronik Mühendisliği Ana Bilim Dalında akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını, tamamen kendi çalışmam olduğunu, her alıntıya kaynak gösterdiğimi, tezin yazımında akademik ve etik kurallara aykırı herhangi bir yapay zeka ve program kullanmadığımı beyan ederim.

İmza

Atahan UYANIK



ÖZET

Yüksek Lisans

Atahan UYANIK
Manisa Celal Bayar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Anabilim Dalı

Danışman: Doç. Dr. Ömer AYDIN

Klima, iklimlendirme amacı ile yaygın olarak kullanılan bir sistemdir. İç mekân hava kalitesini artırmak, konforu sağlamak ve hava koşullarına uyumlu bir ortam oluşturmak amacıyla tasarlanmıştır. Yapay zeka ise, bilgisayar sistemlerinin karmaşık verileri analiz etmek, öğrenmek ve belirli görevleri yerine getirmek için insan benzeri akıl ve öğrenme yetenekleri kazanmasını sağlayan bir teknolojidir. Bu teknoloji, genellikle endüstriyel alanlarda üretim süreçlerini iyileştirmek için kullanılmaktadır. Son yıllarda ise günlük yaşamın pek çok alanında yaygın olarak kullanılmaya başlanmıştır. Geliştirilen yapay zeka destekli klima sistemi, bu iki önemli kavramı bir araya getirerek kullanıcı deneyimini optimize etmeyi amaçlamaktadır. Bu sistem, Raspberry Pi ve Pi kamera gibi yaygın ve düşük maliyetli bileşenlerle geliştirilmiştir. Yüz tanıma ve insan algılama gibi temel yapay zeka işlevleri kullanılarak çalışmaktadır.

Yüz tanıma işlevi, Haar Cascade algoritması aracılığıyla gerçekleştirilir ve sistem tarafından tanınan kullanıcıların tercihlerine uygun şekilde klima ayarları otomatik olarak belirlenmektedir. İnsan algılama sistemi ise TensorFlow makine öğrenme kütüphanesi ve YOLO yapay zeka algoritması kullanılarak odadaki insan sayısını ve insanların bulundukları konumları tespit edebilmektedir. Böylece hava akışı optimize edilmekte ve klimadan üflenene hava doğrudan insanların üzerine gelmeden iklimlendirme daha konforlu bir hale getirilmektedir. Ayrıca sistem, odada belirli bir süre boyunca insan algılanmadığı durumda klimayı otomatik olarak enerji tasarruf moduna geçirir. Böylelikle enerji verimliliği sağlayarak sürdürülebilirliğe katkıda bulunmaktadır.

Geliştirilen sistem, yalnızca iklimlendirme alanında değil, aynı zamanda enerji tasarrufu sağlama ve kullanıcı konforunu artırma açısından da önemli bir ilerlemedir. Bu çalışma, yapay zeka teknolojisinin sınırlı işlem gücüne sahip platformlarla bile günlük yaşamda yaygın olarak kullanılabilecek pratik uygulamalar için potansiyel taşıdığını vurgulamaktadır.

Anahtar Kelimeler: Yapay Zeka, Derin Öğrenme, Yüz Tanımlama, İnsan Algılama, Raspberry Pi

2024, 104 sayfa



ABSTRACT

M.Sc. Thesis

Atahan UYANIK

**Manisa Celal Bayar University
Graduate School of Education
Department of Electrical and Electronics Engineering**

Supervisor: Doç. Dr. Ömer AYDIN

Air conditioner is a widely used system for the purpose of climate control. It is designed to improve indoor air quality, provide comfort and create an environment compatible with weather conditions. Artificial intelligence is a technology that enables computer systems to acquire human-like intelligence and learning capabilities to analyse complex data, learn and perform specific tasks. This technology is generally used in industrial areas to improve production processes. In recent years, it has been widely used in many areas of daily life. The developed artificial intelligence-supported air conditioning system aims to optimise the user experience by combining these two important concepts. This system is developed with common and low-cost components such as Raspberry Pi and Pi camera. It works using basic artificial intelligence functions such as face recognition and human detection.

The face recognition function is performed by the Haar Cascade algorithm and the air conditioning settings are automatically adjusted according to the preferences of the users who are recognised by the system. The human detection system can detect the number of people in the room and their locations by using the TensorFlow machine learning library and YOLO artificial intelligence algorithm. In this way, air flow is optimised and air flow is made more comfortable without the air blowing from the air conditioner directly on people. In addition, the system automatically sets the air conditioner to energy saving mode when no people are detected in the room for a certain period of time. Thus, it contributes to sustainability by providing energy efficiency.

The developed system is an important advance in not only the area of air-conditioning, at the same time saving energy and increasing user comfort. This study highlights that artificial intelligence technology has the potential for practical applications that can be widely used in daily life, even with platforms with limited processing capacity.

Keywords: Artificial Intelligence, Deep Learning, Face Recognition, Human Detection, Raspberry Pi

2024, 104 pages



ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada, farklı yapay zeka ve derin öğrenme algoritmaları kullanılarak klimalar için bir sistem geliştirilmiştir. Python dilinde geliştirilen algoritmalar, Raspberry Pi platformu üzerinde test edilmiş ve yapılan optimizasyonlar sonucunda son kullanıcı odaklı bir ürüne entegre edilebilir hale getirilmiştir.

Tez çalışmamın tüm aşamalarında benden desteğini esirgemeyen, tecrübe ve bilgi birikimleriyle yol gösteren ve kendisini tanımaktan onur duyduğum danışman hocam Sayın Doç. Dr. Ömer AYDIN'a teşekkürlerimi sunarım.

Yüksek Lisans savunma sınavı jürisinde görev alarak bilgi, tecrübe ve yapıcı eleştirileri ile tezime değerli katkılarından dolayı Doç. Dr. Gökhan DALKILIÇ ve Doç. Dr. İlhan BAŞTÜRK'e teşekkürlerimi sunarım.

Geliştirdiğim sistem için sağladıkları maddi ve manevi desteklerden dolayı çalıştığım şirket olan VESTEL Beyaz Eşya Sanayi ve Ticaret A.Ş.'ye teşekkürlerimi sunarım.

Son olarak, tez çalışmam boyunca her türlü konuda destek veren sevgili eşim Ayça UYANIK'a tüm kalbimle teşekkür ederim.

Atahan UYANIK
Manisa, 2024

SİMGELER VE KISALTMALAR DİZİNİ

°C	Derece Santigrat
Adaboost	Adaptive Boosting (Adaptif Güçlendirme)
ADC	Analog-Digital Coverter (Analog-Dijital Dönüştürücü)
AES-256	Advanced Encryption Standart (Gelişmiş Şifreleme Standartı)
API	Application Programming Interface
ASM	Active Shape Model (Aktif Şekil Modeli)
ATM	Automatic Teller Machine (Otomatik Vezne Makinesi)
BSD	Berkeley Software Distribution (Berkeley Yazılım Dağıtımı)
CART	Classification and Regression Trees
CCD	Charge-Coupled Device
CMOS	Tamamlayıcı Metal Oksit Yarı İletken
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
CO₂	Karbondioksit
COCO	Common Objects in Context (Bağlam İçindeki Ortak Nesneler)
COVID-19	Koronavirüs Hastalığı 2019
DenseNet	Dense Convolutional Network
FPS	Frame Per Second (Saniye Başına Kare)
GHz	Giga Hertz
GPS	Global Positioning System Küresel (Konumlandırma Sistemi)
GPIO	General Purpose Input/Output (Genel Amaçlı Giriş/Çıkış)
HOC	Histogram of Oriented Gradients
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things (Nesnelerin İnterneti)
IDE	Integrated Development Environment (Entegre Geliştirme Ortamı)
JPEG	Joint Photographic Experts Group (Ortak Fotoğraf Uzmanları Grubu)
KDEF	Karolinska Directed Emotional Faces
LBP	Local Binary Pattern (Yerel İkili Örüntü)
LIDAR	Light Detection and Ranging (Işık Algılama ve Mesafe Ölçümü)
mA	Miliamper
mm	Milimetre
mNm	Mili Newtonmetre
ms	Milisaniye

NTC	Negative Temperature Coefficient (Negatif Sıcaklık Katsayısı)
OpenCV	Open Source Computer Vision (Açık Kaynaklı Bilgisayar Görüsü)
PCA	Principal Component Analysis (Temel Bileşen Analizi)
PIN	Personal Identification Number (Kişisel Tanımlama Numarası)
PIR	Passive Infrared (Pasif Kızılötesi)
PWM	Pulse Width Modulation (Darbe Genlik Modülasyonu)
PyAutoGUI	Python Automate Graphical User Interface
SMS	Short Message Service (Kısa Mesaj Servisi)
SPI	Serial Peripheral Interface (Seri Çevresel Arayüz)
SVM	Support Vector Machine (Destek Vektör Makinesi)
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus (Evrensel Seri Yolu)
WHO	World Health Organization (Dünya Sağlık Örgütü)
Wi-Fi	Wireless Fidelity (Kablosuz Bağlantı)
YOLO	You Only Look Once (Sadece Bir Kez Bak)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3. 1. Raspberry Pi 4 8GB [33]	24
Şekil 3. 2. Raspberry Pi Kamera v2 [33]	27
Şekil 3. 3. NTC Termistör [41]	29
Şekil 3. 4. MCP3008 ADC Entegresi [42]	30
Şekil 3. 5. Step Motor Topolojileri [45]	31
Şekil 3. 6. Step Motor [46]	32
Şekil 3. 7. ULN2003A Step Motor Sürücü Kartı [48]	33
Şekil 3. 8. OpenCV Temel Bileşenleri [51]	34
Şekil 3. 9. Haar Cascade Algoritmasının Yüz Algılama Adımları	38
Şekil 3. 10. CNN Genel Mimarisi [61]	41
Şekil 3. 11. YOLO Algoritması Nesne Algılama Süreci [10]	43
Şekil 3. 12. Tensorflow Çalışma Prosesi [65]	45
Şekil 3. 13. MCP3008 ve NTC Bağlantı Şeması	47
Şekil 3. 14. İç Ortam Sıcaklığı Ölçüm Algoritması	49
Şekil 3. 15. İç Ortam Sıcaklığı Ölçüm Algoritmasının Akış Şeması	50
Şekil 3. 16. Haar Cascade Sınıflandırıcı ile Yüz Algılanması [68]	51
Şekil 3. 17. Veri Setleri Hazırlığı Yazılım Kodu	52
Şekil 3. 18. Veri Seti Oluşturma Algoritması Akış Şeması	53
Şekil 3. 19. Eğitimden Önce ve Sonra Algılanan Yüzlerin Etiketlenmesi	55
Şekil 3. 20. Algılanan Yüzün Etiketlenmesi	56
Şekil 3. 21. Aynı Anda Algılanan İki Yüzün Etiketlenmesi	57
Şekil 3. 22. Etiket İçerisindeki Karakterler ile Oluşturulan Liste	57
Şekil 3. 23. Tanımlanan Yüz Etiketinden Gerekli Verilerin Ayırıştırılması	58
Şekil 3. 24. Sistemin Eğitilme Akış Diyagramı	59
Şekil 3. 25. Yüz Tanıma Sistemi Akış Diyagramı	61
Şekil 3. 26. Step Motor Hareket Testi	63
Şekil 3. 27. Fotoğraflardaki Farklı Nesnelerin Algılanması 1	66
Şekil 3. 28. Fotoğraflardaki Farklı Nesnelerin Algılanması 2	66
Şekil 3. 29. Gerçek Zamanlı Görüntülerdeki Nesnelerin Algılanması	67
Şekil 3. 30. Algılanan Nesnelerden İnsanların Ayırıştırılması	68
Şekil 3. 31. İnsan Algılama Algoritması Akış Şeması	68
Şekil 3. 32. Fotoğrafta Yer Alan İnsanların Tespit Edilmesi	69
Şekil 3. 33. Gerçek Zamanlı İnsan Tespiti	69
Şekil 3. 34. Algılanan İnsanların Konumunun Tespit Edilmesi	71
Şekil 3. 35. Algılanan İnsanların Konum Tespit Algoritması Akış Şeması	72
Şekil 3. 36. Gerçek Zamanlı Algılanan İnsanın Konumunun Tespit Edilmesi	73
Şekil 3. 37. Gerçek Zamanlı Birden Çok İnsanın Konumunun Tespit Edilmesi	73
Şekil 3. 38. Algılanan İnsanın Konumuna Göre Step Motor Kontrolü	75
Şekil 3. 39. İnsan Konumuna Göre Step Motor Kontrol Algoritması Akış Şeması	76
Şekil 3. 40. Algılanan İnsanların Sayısının Tespiti	78
Şekil 3. 41. Algılanan İnsan Sayısının Hesaplanma Algoritması Akış Şeması	79

Şekil 3. 42. Gerçek Zamanlı Birden Çok İnsanın Algılanması 1. Durum	80
Şekil 3. 43. Gerçek Zamanlı Birden Çok İnsanın Algılanması 2. Durum	80
Şekil 3. 44. İnsan Algılama Sistemi Akış Şeması	82
Şekil 4. 1. Seri Ekran Üzerinden Okunan Sıcaklık Değerleri	84
Şekil 4. 2. Veri Tabanına Kaydedilen Veri Setlerinin Formatı	85
Şekil 4. 3. Anlamlı Bilgilerin Seri Ekranda Gösterilmesi	86
Şekil 4. 4. Yansıyan Görüntülerin Algılanması	90
Şekil 4. 5. Yansıyan Görüntülerin Algılanmaması	91



TABLO DİZİNİ

Sayfa

Tablo 3. 1. Raspberry Pi 4 Teknik Özellikleri [33]	24
Tablo 3. 2. Raspberry Pi Kamera v2 Teknik Özellikleri [33]	27
Tablo 3. 3. NTC Termistör Teknik Özellikleri [39].....	28
Tablo 3. 4. MCP3008 ADC Entegresi Teknik Özellikleri [40]	30
Tablo 3. 5. Step Motor Teknik Özellikleri [44]	32
Tablo 3. 6. ULN2003A Sürücü Entegresi Teknik Özellikleri [46].....	33
Tablo 4. 1. İç Ortam Sıcaklık Ölçüm Testleri	84
Tablo 4. 2. Gerçekleştirilen Test Sonucunda Elde Edilen Parametreler	88
Tablo 4. 3. Sınır Değeri Belirleme Testi Sonuçları.....	92
Tablo 4. 4. Farklı Sensörler ile Yapay Zekalı Kameranın Karşılaştırılması	93

İÇİNDEKİLER

ÖZET.....	I
ABSTRACT.....	III
ÖNSÖZ VE TEŞEKKÜR.....	V
SİMGELER VE KISALTMALAR DİZİNİ.....	VI
ŞEKİLLER DİZİNİ	VIII
TABLO DİZİNİ	X

BİRİNCİ BÖLÜM

1. GİRİŞ.....	1
1.1. Tezin Amacı.....	4
1.2. Tezin Önemi.....	5
1.3. Tezin Katkısı.....	5

İKİNCİ BÖLÜM

2. LİTERATÜR ARAŞTIRMASI.....	6
-------------------------------	---

ÜÇÜNCÜ BÖLÜM

3. MATERYAL VE YÖNTEMLER	23
3.1. Materyaller	23
3.1.1. Raspberry Pi	23
3.1.2. Kamera.....	25
3.1.3. Sıcaklık Sensörü.....	27
3.1.4. MCP3008 ADC Entegresi	29
3.1.5. Step Motor	30
3.1.6. Step Motor Sürücüsü	32
3.1.7. OpenCV Görüntü İşleme Kütüphanesi.....	33
3.1.8. Haar Cascade Yöntemi	36
3.1.9. YOLO Derin Öğrenme Algoritması.....	39
3.1.9.1. CNN Tabanlı Görsel Tanıma	40
3.1.9.2. YOLO Mimarisi	42
3.1.10. Tensorflow Makine Öğrenimi Kütüphanesi.....	43
3.1.11. Yazılım Geliştirme Programı	45
3.2. Yöntemler.....	46
3.2.1. İç Ortam Sıcaklığının Ölçülmesi	46
3.2.2. Gerçek Zamanlı Yüz Tanıma Sisteminin Geliştirilmesi.....	50
3.2.2.1. Veri Setlerinin Oluşturulması ve Sistemin Eğitilmesi	51
3.2.2.2. Yüz Tanıma Sisteminin Geliştirilmesi	54
3.2.2.3. Sıcaklık Verisinin Yüz Tanıma Sistemine Eklenmesi	55
3.2.2.4. Anlamlı Çıktıların Elde Edilmesi	56

3.2.3.	Dikey Kanat Kontrolü	62
3.2.4.	Gerçek Zamanlı İnsan Algılama Sisteminin Geliştirilmesi	64
3.2.4.1.	Nesne Algılama Algoritmasının Geliştirilmesi	65
3.2.4.2.	İnsan Algılama Algoritmasının Geliştirilmesi	67
3.2.4.3.	Algılanan İnsanların Konumunun Belirlenmesi	70
3.2.4.4.	Konum Tabanlı Dikey Kanat Yönlendirilmesi.....	74

DÖRDÜNCÜ BÖLÜM

4.	ARAŞTIRMA BULGULARI VE TARTIŞMA	83
4.1.	İç Ortam Sıcaklık Ölçümlerinin Doğrulanması	83
4.2.	Gerçek Zamanlı Yüz Tanıma Sistemi Testleri.....	85
4.3.	Gerçek Zamanlı İnsan Algılama Sistemi Testleri.....	89

BEŞİNCİ BÖLÜM

5.	SONUÇ VE ÖNERİLER.....	95
KAYNAKLAR.....		99

1. GİRİŞ

Akıllı ev konsepti, modern yaşam biçimlerinin dönüşümünde önemli rol oynayan ve teknolojinin yenilikçi potansiyelini ev yaşamına entegre eden kapsamlı bir yaklaşımdır. Akıllı ev sistemleri, ev otomasyonundan bir adım öteye giderek konfor, güvenlik, enerji verimliliği ve kişiselleştirilmiş bir deneyim açısından benzersiz avantajlar sunmaktadır. Ev sahiplerinin yaşam kalitesini artırmak için bu sistemler günümüzde giderek daha fazla ilgi görmektedir [1].

Akıllı Ev kavramının temelleri ilk olarak, 1960'ların başlarında kablolu evlerin geliştirilmesiyle atıldı. Bu evler, temel elektrik kontrolleriyle donatılmış olup, o dönemin teknolojisinin sınırları içerisinde kullanılan yeniliklerdi. Ancak, Akıllı Ev terimi resmi olarak ilk kez 1984 yılında American Association of House Builders (Amerikan Ev İnşaatçılar Derneği) tarafından kullanıldı. Bu dönemdeki teknolojik ilerlemeler, ev otomasyonu konseptinin daha geniş bir çerçevede ele alınmasını sağladı. Günümüzde Akıllı Ev kavramı yaygın bir şekilde kullanılsa da birçok akıllı cihaz gerçekte tam anlamıyla akıllı değildir. Bazı ürünler, pazarlama stratejisi olarak kullanılarak üstünlük sağlamak ve müşteri talebi oluşturmak amacıyla "akıllı" olarak adlandırılır. Bu da aslında, tüketicilerin akıllı ev sistemleri konusundaki talepleri ve beklentileri göz önünde bulundurularak yapılan bir yaklaşımdır. Riverside Concept Houses, bir evin gerçekten "akıllı" olarak kabul edilebilmesi için, ev içindeki aydınlatma, havalandırma, soğutma, ısıtma gibi elektrikle kontrol edilebilen sistemlerin bir uzaktan kumanda veya internet aracılığıyla yönetilebilmesi gerektiğini belirtiyor. Bu yaklaşım, ev sahiplerine daha fazla kontrol ve esneklik sunarak yaşam kalitesini artırmayı hedefliyor [2].

Araştırmacılar, "Akıllı Ev" terimini özellikle yapay zeka destekli öğrenme algoritmalarının entegre edildiği sistemler olarak tanımlamaktadır. Bu öğrenme algoritmaları, evin ısıtma, soğutma gibi temel işlevlerini gerçekleştirebilme kabiliyetine sahip olarak, evin çeşitli koşullarına ve kullanıcı tercihlerine duyarlı bir şekilde otomatik ayarlamalar yapabilir. Bu sayede, enerji verimliliğini en üst düzeye çıkarmakla kalmaz, aynı zamanda kullanıcıların yaşam kalitesini artırarak daha

konforlu ve sürdürülebilir bir yaşam tarzı sunar. Bu akıllı sistemler, ev sahiplerinin günlük rutinlerine entegre olurken, enerji tasarrufu ve konforun mükemmel bir denge içinde sağlanması hedeflenir. Özellikle değişen ihtiyaçlara anında cevap verebilme yeteneği, akıllı ev teknolojisinin gelecekteki önemli yönlerinden biri olarak öne çıkmaktadır [3].

Ev otomasyon sistemleri, kullanıcılarına konfor ve güvenlik sağlama amacıyla geliştirilen teknolojik çözümlerdir. Bu sistemler, yapay zeka, makine öğrenimi ve derin öğrenme gibi çağdaş teknolojik gelişmelerin etkisiyle ilk ortaya çıktığı yıllara kıyasla büyük bir evrim göstermiştir. Günümüzde, ev sahipleri için sadece bir lüks değil, aynı zamanda yaşam kalitesini artıran temel bir gereksinim haline gelmiştir. Yapay sinir ağları ve diğer ileri düzey algoritmaların kullanılmasıyla tasarlanan bu sistemler, evlerin ve işyerlerinin daha akıllı ve etkili bir şekilde yönetilmesini sağlar. Bu teknolojiler sayesinde, evlerdeki ışıklandırmadan klima sistemlerine, güvenlik kameralarından enerji tüketimi optimizasyonuna kadar bir dizi önemli işlem otomatik olarak gerçekleştirilebilir hale gelir. Bu evrim, akıllı ev sistemlerinin sadece bireylerin yaşam tarzını dönüştürmekle kalmayıp, aynı zamanda sürdürülebilirlik ve enerji tasarrufu gibi toplumsal faydaları da desteklediğini göstermektedir. Ev otomasyonunun geleceği, teknolojik inovasyonlar ve kullanıcı ihtiyaçlarına dayalı olarak daha da parlak ve verimli görünmektedir. Bu bağlamda, ev otomasyon sistemlerinin geliştirilmesi ve benimsenmesi, modern yaşamın gerektirdiği akıllı ve sürdürülebilir bir geleceğin temel taşlarından birini oluşturmaktadır [4].

Sonuç olarak, Akıllı Ev konsepti, teknolojinin ev yaşamına entegre edilmesiyle ev sahiplerine daha akıllı, esnek ve kişiselleştirilmiş bir yaşam tarzı sunmayı amaçlamaktadır. Gelişen teknoloji ile görüntü işleme ve derin öğrenme gibi yenilikçi teknolojilerin bu konseptte entegre edilebilirliği, evlerin daha sürdürülebilir, verimli ve kullanıcı dostu hale gelmesi açısından büyük potansiyel taşımaktadır. Bu teknolojiler sayesinde, ev otomasyon sistemleri ev sahiplerinin tercihlerine ve ihtiyaçlarına daha hassas bir şekilde yanıt verebilir, güvenlik önlemleri daha etkili bir biçimde uygulanabilir ve enerji verimliliği optimize edilebilir. Bu entegrasyon, Akıllı Ev

konseptinin ev sahiplerine sağladığı avantajları daha da artırarak gelecekteki ev yaşamının temelini oluşturabilir.

Dijital görüntü işleme, görüntülerin bilgisayar tabanlı yöntemlerle analiz edilmesini ve manipüle edilmesini amaçlayan bir alandır. Kökleri 1960'ların başlarına dayanırken, özellikle 1980'lerde ticari sistemlerin ilk kullanımıyla önemli bir evrim yaşamıştır. 1990'ların ortalarında daha karmaşık görüntü analizi teknikleri geliştirilirken, 2000'lerde yapay zeka ve derin öğrenme tekniklerinin yükselişi, nesne ve yüz tanıma gibi alanlarda büyük atılımlara neden olmuştur. Bugün, dijital görüntü işleme, akıllı cihazlar, tıbbi görüntüleme, güvenlik ve sanat gibi geniş bir yelpazede kullanılmaktadır. Bu süreçte artan işlem kapasitesi, gelişen algoritmalar ve yeni teknolojiler, dijital görüntü işlemenin hızlı evrimini sağlamıştır [5].

Derin öğrenme, insan beyninin nesne tanıma ve anlama süreçlerini taklit etmeye yönelik bir yaklaşımı temsil eder. Bu alanda amaç, bir nesneyi tanımak için gerekli olan özellikleri hiyerarşik bir yapıda ortaya çıkarmaktır. İnsan beyninin işleyişini yansıtmak üzere tasarlanmış olan derin öğrenme, birden çok katmandan oluşan doğrusal olmayan özelliklerin keşfedilmesine dayanır. Bu katmanlar, verinin daha yüksek seviyeli özelliklerle temsil edilmesini sağlar ve bu sayede karmaşık ilişkileri anlama yeteneği kazandırır. Bu bağlamda, derin öğrenme yaklaşımı, büyük veri setlerinden anlamlı desenlerin çıkarılmasını, nesnelerin, yüzlerin ve diğer karmaşık yapıların tanınmasını hedefler [6]. Tarih boyunca, bu yaklaşımın gelişimi büyük bir evrim geçirmiştir. İlk adımlar 1943 yılında McCulloch ve Pitts tarafından atılmıştır; düşünce süreçlerinin matematiksel ve sinir mantığı temelinde modellenmesi fikri ortaya atılmıştır. Ardından, 1958'de Rosenblatt iki katmanlı yapay sinir ağı tabanlı denetimli öğrenme algoritmasıyla desen tanıma alanında önemli bir ilerleme kaydetmiştir. 1965'te Ivakhnenko ve Lapa, karmaşık denklemlerle derin öğrenme algoritmalarını inceleyerek bu alandaki araştırmaları daha da derinleştirmişlerdir. 1988'de Fukushima'nın çok katmanlı yapay sinir ağı olan Neocognitron'u önermesi, el yazısı tanıma gibi alanlarda başarılı uygulamaların kapısını aralamıştır. Derin öğrenme, zaman içinde nesne tanıma, sınıflandırma ve diğer pek çok alanda kullanılabilirliğini kanıtlamıştır. Özellikle 2012'de Google'ın büyük ölçekli yapay

desen tanıma algoritmalarının insan seviyesinde performans sergilemesi ve 2014'te Facebook'un otomatik yüz tanıma sistemleri geliştirmesi, derin öğrenmenin potansiyelini iyiden iyiye ortaya koymuştur. Bu süreç, derin öğrenmenin insan benzeri öğrenme kapasitesini modelleme yolculuğunun sadece başlangıcıdır [7].

Görüntü işleme ve derin öğrenme, genellikle endüstriyel sektörlerde üretim proseslerinin yönetimi, kalite kontrol süreçlerinin optimize edilmesi ve veri analizi gibi alanlarda kullanılmaktadır. Bu teknikler, büyük veri setlerinin analiz edilmesi, karmaşık desenlerin tanınması ve öğrenme yeteneğine sahip yapay sinir ağlarının kullanılması yoluyla etkili sonuçlar elde etme amacını taşır. Özellikle kalite kontrol süreçlerinde hataların tespit edilmesi ve önemli kararların desteklenmesi gibi kritik görevlerde yaygın olarak kullanılmaktadır. Görüntü işleme ve derin öğrenme tekniklerinin bu endüstriyel uygulamaları, üretim süreçlerinin optimize edilmesi ve maliyet tasarrufu sağlanması açısından büyük önem taşımaktadır [8].

Ancak, görüntü işleme ve derin öğrenme tekniklerinin günümüzdeki hızlı ilerlemesi, sadece endüstriyel alanda değil, aynı zamanda günlük yaşamın daha bireysel ve kişisel alanlarında da büyük potansiyel taşıdığını göstermektedir. Bu bağlamda, son kullanıcıların hayatlarını daha akıllı, kişiselleştirilmiş ve optimize edilmiş bir şekilde yönetmelerini sağlamak amacıyla bu teknolojilerin akıllı ev sistemlerine entegre edilmesi büyük bir önem taşımaktadır.

1.1. Tezin Amacı

Bu tez çalışması, klima ve termosifon tasarımı ve üretimi yapan bir fabrikada gerçekleştirilerek, akıllı ev sistemlerine entegre edilecek görüntü işleme ve yapay zeka teknolojileriyle enerji tasarrufu sağlayan bir proje sunmayı amaçlamaktadır. Özellikle, bu projenin odak noktası, kişiselleştirme özelliklerine sahip akıllı bir klima sistemi geliştirmektir. Bu sistem hem yüz tanıma hem de insan algılama teknolojilerini kullanarak kullanıcı tercihlerine göre otomatik ayarlamalar yaparak ideal bir konfor ve enerji verimliliği dengesi sağlamayı hedeflemektedir.

1.2. Tezin Önemi

Bu çalışmanın önemi, akademik araştırmaların pratik uygulamalara dönüşme potansiyelini vurgulamaktadır. Günümüzde enerji verimliliği ve çevresel sürdürülebilirlik giderek daha büyük önem kazanmakta, bu bağlamda akıllı ev sistemlerinin etkisi de giderek artmaktadır. Bu çalışma, akıllı klima sistemleri üzerine yapılan yenilikçi araştırmaları vurgulayarak, endüstriyel uygulamalarını anlamamıza katkıda bulunmakta ve sektöre önemli katkılar sunma potansiyeline sahiptir. Ayrıca, sadece teorik bir çalışma olmaktan öte, ticari bir ürün haline dönüşebilecek potansiyele sahip olması, üniversite-sanayi iş birliği kapsamında firmaların akademik çalışmaların sonuçlarından doğrudan faydalanabileceğini ve böylece inovasyonu destekleyerek sektöre önemli katkılar sağlayabileceğini göstermektedir. Bu sayede, enerji tasarrufu ve çevresel sürdürülebilirlik gibi kritik konularda sektöre önemli katkılar sağlama potansiyeline sahiptir.

1.3. Tezin Katkısı

Bu çalışma, klima üretim sürecinde geleneksel yöntemlerin ötesine geçerek, son kullanıcı odaklı bir üründe kullanılmayan görüntü işleme ve yapay zeka tekniklerini klima sistemlerine entegre etmeyi amaçlamaktadır. Yüz tanıma ve insan algılama teknolojilerinin birleştirilmesiyle gerçekleştirilen kişiselleştirilmiş otomatik ayarlamalar, enerji verimliliği ve kullanıcı deneyiminde belirgin bir iyileşme sağlamaktadır. Özellikle, odada insan varlığı algılanmadığında klimanın otomatik olarak enerji tasarrufu moduna geçmesi gibi yenilikler, çevresel etkiyi azaltma ve enerji tasarrufu sağlama açısından önemlidir. Bu çalışma, akademik araştırmaların pratik uygulamalara dönüşme potansiyelini vurgulamanın yanı sıra, endüstrideki yeniliklere öncülük etme ve sektöre katkıda bulunma amacını taşımaktadır. Akıllı klima ve termosifon sistemlerinin bu yenilikçi özellikleri, gelecekteki ev teknolojilerinin gelişimine ivme kazandırabilir ve enerji verimliliği konusunda öncü bir rol oynayabilir.

2. LİTERATÜR ARAŞTIRMASI

Son yıllarda, örüntü tanıma çalışmalarındaki yoğun ilgi özellikle yüz tanıma sistemlerinin modern uygulama alanlarına odaklanmasına neden olmuştur. Bu sistemler, görüntü analizi alanında son derece başarılı uygulamaların geliştirilmesine olanak sağlamıştır. Yüz tanıma, temel olarak insanları yüz görüntülerinden tanıma sürecini ifade eder. Test edilen görüntüler, eğitilmiş görüntü grupları ile karşılaştırılarak test görüntülerinin eğitilmiş görüntülerle ne kadar uyumlu olduğu belirlenir. Yüz tanıma sistemlerinin uygulanmasında genellikle öznitelik çıkarımı yöntemi kullanılır, bu da yüz görüntülerinin özelliklerinin çıkarılmasını içerir. Bu süreç artık ön işleme ve geometrik normalleştirme ayrılmaz bir parçasıdır. Elde edilen özellikler daha sonra, benzer özelliklere sahip diğer görüntülerle eşleştirme amacıyla kullanılabilir hale gelir [9].

İnsan algılama da benzer bir süreci içerir; insan algılama sistemleri, insanları algılamak ve tanımak için görüntü analizi ve öznitelik çıkarımı tekniklerini kullanır. Bu sistemler, CNN gibi derin öğrenme modellerinin kullanımıyla daha da geliştirilmektedir. CNN'ler, özellikle görüntü tanıma ve işleme alanında başarılı sonuçlar veren yapay sinir ağlarıdır. Bu sinir ağları, biyolojik sinir ağlarından ilham alarak, önceki katmanlarda basit özellikleri algılar ve ardından daha karmaşık özellikleri öğrenmek için katmanlar arasında bilgiyi işler. İnsan algılama sistemleri, CNN'nin öğrendiği öznitelikleri kullanarak insanları algılamak ve tanımak için daha etkili bir şekilde çalışabilir. CNN'lerin kullanımı, insan algılama sistemlerinin hassasiyetini artırırken, aynı zamanda doğruluk oranlarını ve performansı da iyileştirebilir. Dolayısıyla, modern insan algılama sistemleri, CNN gibi derin öğrenme modellerini entegre ederek, görüntü analizi ve öznitelik çıkarımı alanında önemli bir ilerleme kaydetmektedir [10].

Bu bölümde, yüz tanıma ve insan algılama sistemlerinin geliştirilmesindeki zorluklar ve bu alandaki önceki çalışmalar ele alınmaktadır. Yüz tanıma sistemleri, görüntü analizi ve yapay zeka tekniklerinin birleşimiyle insan yüzlerini algılamayı ve tanımayı amaçlayan sistemlerdir. Benzer şekilde, insan algılama sistemleri, bir

ortamda insan varlığını tespit etmeyi ve tanımayı hedefleyen sistemlerdir. Her iki alanda da sistemlerin doğruluk oranlarını etkileyen faktörler arasında görüntü kalitesi, aydınlatma koşulları, insanların pozisyonları ve örtüleri gibi çeşitli faktörler bulunmaktadır. Önceki araştırmalar, bu zorlukların üstesinden gelmek için çeşitli yöntemler ve yaklaşımlar geliştirmiştir. Bu bölüm, yüz tanıma ve insan algılama sistemlerinin mevcut durumunu değerlendirmek ve bu alandaki gelişmeleri incelemek için mevcut literatürden edinilen bilgi birikimini özetlemektedir.

Girit tarafından yapılan çalışmada; araç sürücülerinin durumlarını tespit etmek için görüntü işleme teknolojisinin kullanımına odaklanılmıştır. Özellikle, sürücülerin göz bölgeleri incelenerek uyanıklık, uykulu veya uyku halinde olup olmadıkları belirlenmeye çalışılmıştır. İnsan yüzünün tespit edilmesini takiben, gözlerin belirlenmesi aşamasında karar verme algoritması uygulanmıştır. Bu amacı gerçekleştirmek için, gri skalalama, histogram eşitleme ve yeniden ölçeklendirme gibi işlemler görüntülerin yüz ve göz bölgelerinin daha net bir şekilde yorumlanmasını sağlayarak algoritmanın daha etkili çalışmasına katkıda bulunmuştur. Yüz ve göz bölgelerinin tespitinde, kaskad (cascade) şeklinde çalışan Viola-Jones yöntemi ve CART yüz tanıma yöntemi kullanılmıştır. Bu yöntemlerde, nesnenin boyutuna bağlı olarak değişen Haar dalgacıkları adı verilen siyah-beyaz pikseller yer almaktadır. Haar dalgacıkları sayesinde her piksel taranarak istenen nesnenin öznitelikleri çıkarılmaktadır. Sürücünün uyanıklık, uykululuk veya uyku durumları gibi farklı durumlarını tahmin etmek için yüz ve göz bölgelerinin tespitine ek olarak, öğrenen bir algoritma geliştirilmiştir. Bu algoritma, tespit edilen özelliklerin analizi ve değerlendirmesi yoluyla sürücünün durumunu belirlemeye odaklanmaktadır. Video kesitleri incelendiğinde, görüntü işleme teknikleriyle göz bölgeleri tanımlanır ve bu bölgeler yapay sinir ağlarına sağ ve sol göz sınıflandırıcıları olarak beslenir. Bu sinir ağları, gözleri açık, yarı açık veya kapalı olarak sınıflandırabilir. Tüm göz durumları videonun süresi boyunca bir araya getirilerek sürücünün uyanık veya uykulu olup olmadığı tahmin edilir. Önerilen yöntem, 30 saniyelik video kesitlerinde test edilmiştir ve sürücü durumu tespit yönteminin başarı oranı %99,1'e, göz durumu tespit yönteminin ise %94'e ulaşmaktadır [11].

Thabet ve Amor tarafından yapılan çalışmada; yüz tanımaya dayalı akıllı bir kapı zili sistemi oluşturmak hedeflemektedir. Bu çalışmanın temel aşaması, yüz algılama adımını içermekte olup, bu adım Haar benzeri filtrelerin kullanımıyla gerçekleştirilmektedir. Daha sonra, veritabanı kayıtlarıyla benzerlik arayışıyla ilerlenmektedir. Sistem, yüz tanıma sürecini üç ayrı ana bölüme ayırmaktadır: yüz algılama, özellik çıkarma ve yüz tanıma. İlk aşamada, görüntüler analiz edilerek insan yüzleri tespit edilir ve yüzlerin konumları belirlenir. Bu aşamada elde edilen sonuçlar, yüzleri karakterize eden yamalardır ve algoritmanın etkinliğini artırmak amacıyla giriş görüntüsüne yüz hizalama ve ölçeklendirme filtreleri uygulanmaktadır. İnsan yüzleri tespit edildikten sonra, görüntülerden yüz yamaları çıkarılır. Aydınlatma, yüz ifadeleri ve diğer çevresel faktörlerden kaynaklanan eksiklikleri en aza indirmek amacıyla, özellik çıkarımı yoluyla boyut azaltma ve gürültü azaltma işlemleri gerçekleştirilir. Bu işlemler sonucunda elde edilen yüz görüntüsü, sabit boyutlu bir vektöre dönüştürülerek saklanır. Eğitim verileri hazırlandıktan ve yüz vektörleri işlendikten sonra, depolanan verilerle giriş görüntüsü arasında eşleştirme algoritması uygulanır. Bu süreç, bir giriş görüntüsü geldiğinde, yüz algılamanın özellikleri belirleyeceği, ardından özellik çıkarmanın yalnızca yüzü çıkaracağı ve çıkarılan özellikleri veritabanındakilerle karşılaştıracağı şekilde ilerlemektedir. Bu bağlamda, yüz tanıma amacıyla OpenCV kütüphanesindeki Eigen yüz algoritması kullanılmaktadır. Gri ölçekli görüntüye dönüştürülen bir görüntüde, yüz tespitini gerçekleştirmek için Eigen yüz algoritması kullanılırken, veritabanında depolanan veri boyutunu azaltmak amacıyla yüz görüntüleri 8 bit yoğunluk değerine sahip iki boyutlu dizilere dönüştürülür. Tasarlanan sistemde kapı ziline basıldığında yüz tanıma algoritması otomatik olarak etkinleşmektedir. Yüz algılaması gerçekleştikten sonra, veritabanında kayıtlı olan yüzlerle karşılaştırma yapılmaktadır. Algılanan yüz, veritabanında bulunan kayıtlarla eşleşirse, kapı otomatik olarak açılmakta ve sistem yöneticisine Android ve internet arayüzü üzerinden bildirim gönderilmektedir. Algılanan yüz, veritabanında kayıtlı değilse, sistem yöneticisi tarafından kolaylıkla veri tabanına eklenme imkanına sahiptir. Geliştirilen algoritma sonuçları, düzgünce ışıklandırılmış bir ortamda elde edilmiş olup, ortalama tanıma doğruluğu %94,53 olarak belirlenmektedir. Bu sonuçlar, yüz tanıma teknolojisinin etkinliğini ve sistemin güvenilirliğini vurgulamaktadır [12].

Çapşek ve Karacı tarafından yapılan çalışmada; derin öğrenme tabanlı gerçek zamanlı bir sistem geliştirilmiş olup, yüz hareketleri ve göz kırpma durumları kullanılarak fare kullanımı sağlanmaktadır. Deneysel çalışma sonuçları, yüz hareketleri ile fare konumunun başarılı bir şekilde kontrol edilebildiğini ve göz kırpma durumlarının hızlı bir şekilde algılanarak farenin sol, sağ ve çift tıklama işlevlerini modelleyebildiğini göstermektedir. Geliştirilen sistemde, yüz hareketlerinin algılanması için OpenCV kütüphanesi kullanılmıştır. Yüz hareketlerinin tespiti sonucunda fare imlecinin ilgili hareketlere göre konumlandırılabilmesi amacıyla, Python tabanlı PyAutoGUI kütüphanesi tercih edilmiştir. Böylece, algılanan yüz hareketleriyle fare imleci koordinatları değiştirilebilmektedir. Benzer bir yaklaşım göz kırpma hareketleri için de kullanılmıştır. Göz kırpma hareketleri, makine öğrenmesi algoritmalarını içeren Dlib kütüphanesi ile modellenmiştir. Ardından, fare tıklama işlemleri ve gözlerin hangisinin kırıldığının belirlenmesi için Tensorflow ve Keras gibi kütüphanelerle bir CNN modeli oluşturulmuştur. Sonuç olarak, sağ göz kırıldığında fare sağ tıklama işlemi gerçekleştirirken, iki göz kırıldığında fare çift tıklama işlevini yerine getirmektedir. Bu sistem, farklı bireyler tarafından test edilmiş ve benzer sonuçlar elde edilmiş olmasıyla, engelli bireylerin eğitim ve günlük bilgisayar kullanımı gibi alanlarında önemli kolaylıklar sağlayabileceğini göstermektedir [13].

Eldem ve ark. tarafından yapılan çalışmada; ilerleyen teknoloji ile artan güvenlik gereksinimlerini karşılamak üzere temel bir uygulama geliştirmek amaçlanmaktadır. Bu sistemde öncelikle bireylerin yüz bölgelerinin fotoğrafları kamera ile yakalanarak bir veri tabanına kaydedilmekte ve bu veri tabanı oluşturulduktan sonra, bireyler gerçek zamanlı olarak kamera karşısına geçtiklerinde yüz bölgeleri işlenmektedir. Yüz bölgeleri renkli formdan gri resme dönüştürülüp daha sonra görüntünün daha net hale getirilebilmesi için parlaklık ve kontrast düzenlemeleri yapılmaktadır. Son aşamada, OpenCV kütüphanesi kullanılarak elde edilen daha net görüntüler, veri tabanında kayıtlı olan görüntülerle karşılaştırılarak bireyin tanınması sağlanmaktadır. Geliştirilen güvenlik sistemi için C# programlama dilinde bir arayüz tasarlanmıştır ve gerçekleştirilen testler, sistemin %79 oranında başarıyla çalıştığını göstermektedir [14].

Jacinta ve ark. tarafından yapılan çalışmada, yapay zeka ve görüntü işleme tekniklerinin kullanıldığı bir sistemle insan yüzlerindeki duygu değişimlerini tespit etmeye odaklanmaktadır. Özellikle, mutluluk, ıstırap, merak, umutsuzluk, öfke, nefret ve nötr duygu ifadelerini tanıma üzerine yoğunlaşmaktadır. Sistem, bir yüz görüntüsünü incelediğinde duygunun algılanabilmesi için çeşitli ilgi noktaları oluşturmaktadır. Ardından, bu ilgi noktaları bireyin nötr ifadesindeki ilgi noktalarıyla karşılaştırılmaktadır. Yüz ifadelerini tanımlamak için görüntüler, CNN tabanlı bir API olan Face++ kullanılarak işlenmektedir. Her bir yüz için 83 ilgi noktasının koordinatları belirlenmektedir. İlgi noktaları, gözler, kaşlar, yanaklar ve ağız gibi yüzün temel bölgelerinde belirlenmektedir. Bu bölgelerin algılanabilmesi için Viola-Jones algoritması kullanılmaktadır. Sistemin etkinliğini artırmak için KDEF veri kümesi ile eğitilmiş bir sınıflandırıcı kullanılmaktadır. KDEF veri kümesi, 35 erkek ve 35 kadın olmak üzere 70 bireyin yüz görüntülerini içerir ve her bir bireyin yüzü 6 farklı duygu ifadesini yansıtacak şekilde yakalanmıştır. Sonuç olarak, yüzdeki ilgi noktaları kullanılarak analiz edilen sistem, duygusal ifadesi olan ve nötr bir ifadeye sahip olan yüz görüntülerinin ilgi noktaları arasındaki korelasyona dayanmaktadır. Bu korelasyonlar nihai olarak kategorize edilir ve benzerlik grafikleri üzerinde öngörülmektedir. Çalışmanın nihai amacı, gerçekçi bir yüz duygu tanıma sistemine ulaşmaktır. Bu bağlamda, ortalama %57,7 doğruluk oranıyla mutluluk, ıstırap, merak, umutsuzluk, öfke, nefret ve küçümseme gibi altı farklı duygu, ayrıca ortalama %95,9 doğruluk oranıyla mutluluk ve merak gibi iki duygunun tanımlanabilme başarısı elde edilmiştir [15].

Das ve ark. tarafında tarafından yapılan çalışmada, TensorFlow ve OpenCV teknolojileri kullanılarak insanların yüzlerinde maske olup olmadığının tespitini sağlayan bir yöntemi önermektedir. COVID-19 salgınının küresel etkisi, dünya genelinde milyarlarca insanın yaşamını derinden etkilemiştir ve birçok sektör üzerinde olumsuz etkiler yaratmıştır. Salgının yayılmasını kontrol etmek zor ve karmaşık bir süreç olmuştur. WHO gibi sağlık otoriteleri, maske takmanın, enfeksiyon riskini azaltmak için önemli bir önlem olduğunu vurgulamaktadır. Bu proje, insanların yüzlerindeki maskelerin tespiti amacıyla OpenCV, Caffe tabanlı yüz tespit yöntemleri, Keras, TensorFlow ve MobileNetV2 teknolojilerini kullanmaktadır. Geliştirilen sistem eğitmek için iki farklı veri kümesi kullanılmaktadır. İlk veri kümesi toplamda 1376

görüntüden oluşmaktadır ve bu görüntülerin 690'ı maske takan kişiler geri kalan 686'sı ise maske takmayan kişilerden oluşmaktadır. Veri kümesinde bulunan maskeli kişilerin görüntülerinin hepsinde beyaz renkli maske bulunmaktadır. Bir diğer veri kümesinde ise toplamda 853 görüntü bulunmaktadır. Bu veri kümesindeki maskeli kişilerin görüntülerinde farklı tip ve renklerde maskeler kullanılmaktadır. Bu iki veri kümesi ile eğitilen sistem sayesinde fotoğraftan veya video akışından algılanan yüzlerin maskeli olup olmadığı kontrol edilebilmektedir. Geliştirilen sistem ile yüz; maske, saç veya el gibi kısmen görünür olması durumunda dahi etkili bir şekilde tespit edilebilmektedir. Bu tespit sırasında, maske veya elle kapatılmış yüz arasındaki ayrımı yapabilmek için burun, ağız, çene ve göz bölgelerindeki oklüzyon (kapanma) derecesini dikkate almaktadır. Bu sayede, yüzü tamamen kaplayan bir maske olsa dahi, sistem sadece "maskeli" olarak değerlendirmemektedir [16]. Adusumali ve ark. tarafından yapılan çalışmada da benzer bir yöntem önerilmektedir. Yapılan çalışma, diğer benzer çalışmaların aksine maske takmayan kişileri uyarmak amacı ile kişiye maske takması konusunda e-posta veya SMS göndermektedir. Geliştirilen sistemde, insanların yüzlerindeki maskelerin tespiti amacıyla OpenCV, Caffé tabanlı yüz tespit yöntemleri, Keras, TensorFlow ve MobileNetV2 teknolojilerini kullanmaktadır. Projede kullanılan veri kümesi 3.835 görüntü içermekte olup, bu görüntülerin 1.916'sında maskeli insanlar ve 1.919'unda maskesiz insanlar bulunmaktadır. İlk aşamada, Keras ve MobileNetV2 kullanılarak temel bir model oluşturulmaktadır. Bu model, etiketlenmiş veri kümesini eğitmek için kullanılmaktadır. Veri kümesi, görüntülerin %75'ini içeren bir eğitim seti ve kalan %25'ini içeren bir test seti olmak üzere iki kısma ayrılmaktadır. Eğitim sonrasında, insan yüzlerinde maske tespiti için kullanılabilecek bir model elde edilmektedir. Eğitilmiş model, maskeli veya maskesiz insan yüzlerini içeren görüntüler veya video akışları ile beslenebilmektedir. Video akışı durumunda, görüntü veya video karesi, öncelikle varsayılan yüz algılama modülüne iletilmektedir. Algılanan yüzler daha sonra eğitilmiş modele giriş olarak sunulur ve model, yüz bölgesinde maske olup olmadığını tespit etmektedir. Başka bir model, insan yüzleri ile eğitilmektedir. Bu modelin eğitimi için kullanılan görüntüler, kişinin adı ve e-posta adresini içeren etiketlerle sağlanmaktadır. Bu işlem OpenCV kullanılarak gerçekleştirilmektedir. Giriş bir görüntü olduğunda; sistem, algılanan kişiden adını ve e-posta adresini girmesini istemektedir. İlk modelin çıktısı, bu modele giriş olarak sunulmaktadır. Bu yüz, veritabanında kayıtlı kişilerle karşılaştırılır ve eşleşme durumunda yüzün üzerine kişinin adını içeren bir kutu çizer ve kutunun

altında “Maske Yok” ifadesi yer almaktadır. Bununla birlikte kişiye, maske takmadığına dair bir e-posta ve SMS gönderilmektedir. Algılanan kişinin maske takması durumunda, sistem yine kişinin adını içeren bir kutu çizer ve kutunun altında “Maskeli” ifadesi yer almaktadır [17].

Zuo ve de With tarafından yapılan çalışmada, akıllı ev sistemlerine entegre edilebilen bir yüz tanıma sistemi sunmaktadır. Bu sistem, gömülü kameralardan gelen canlı video akışlarını neredeyse gerçek zamanlı olarak işlemekte ve tanımlamayı gerçekleştirmektedir. Her görüntü karesi için, sistem görüntüdeki yüzleri algılar ve bu yüzlerin kaydını tutmaktadır. Ardından, sistem bu tanınan yüzleri veritabanı kayıtları ile kıyaslar ve en iyi eşleşmeyi sunmaktadır. Sistem, üç temel işlem adımı üzerine kurulmuştur. İlk adım, kamera görüntüsünde bir insan yüzünün olup olmadığını tespit eder ve bu, sistemin çalışmasının temelini oluşturur. İkinci adım, algılanan yüzleri normalleştirir ve göz, burun, ağız gibi önemli yüz özelliklerini çıkarır, bu da yüz tanıma sürecinin temelini oluşturur. Üçüncü adım, tanımlama işlemidir, yani yüzlerin veritabanı kayıtlarıyla eşleştirilmesi ve kişilerin tanımlanmasıdır. Yapılan testler, geliştirilen sistemin farklı çevrelerde ve karmaşık arka planlarda yüksek bir başarı oranına sahip olduğunu doğrulamaktadır. Ayrıca, tanıma işlemi için kullanılan 25 kişilik bir veritabanı oluşturulmuş ve her kişi için 4 ila 8 fotoğraf kaydedilmiştir. Yapılan testler sonucunda sistem, %95'lik bir tanıma başarısı elde etmektedir. Sistem ayrıca, zaman içinde değişen ışık koşulları ve farklı yüz görünüşleri gibi zorluklara uyum sağlamak için doğru tanımlanamayan kişileri eğitim setine eklemektedir. Bu sayede, sistem kendisini sürekli olarak geliştiren bir yapıya sahip olmaktadır. Bu yöntemle, sistem geçmiş deneyimlerinden öğrenerek zaman içerisinde performansını aşamalı bir şekilde arttırabilmektedir. Geliştirilen yüz algılama sistemi, akıllı ev uygulamalarına entegre edildiğinde, kullanıcılara akıllı ev deneyimlerini daha da kişiselleştirme ve otomatikleştirme olanağı sağlamaktadır. Özellikle televizyon veya müzik programlarını kişiselleştirmek gibi senaryolar için son derece etkili bir çözüm sunmaktadır [18].

Vaidya ve ark. tarafından gerçekleştirilen çalışma, yaşlı ve engelli insanlar için özel olarak tasarlanmış bir kapı izleme sistemi üzerine odaklanmaktadır. Bu sistem,

kullanıcı dostu bir arayüz ile donatılmıştır, böylece yaşlılar ve engelliler için kullanımı oldukça basit olmaktadır. Temel olarak Raspberry Pi ve Android işletim sistemine dayalı bir çözüm sunmaktadır. Kapı izleme sistemi, bir web kamerası ile Raspberry Pi arasında bir arayüz kullanılarak oluşturulmaktadır. Web kamera, Raspberry Pi'ye bir USB kablosu ile bağlanırken, kapı kilidi de Raspberry Pi ile entegre edilmektedir. Bu sayede kullanıcılar, Android tabanlı arayüz veya fiziksel bir anahtar kullanarak kapıyı manuel olarak açabilmektedir. Kapının önünde biri olduğunda sistem otomatik olarak çalışmaktadır. Kapı ziline basıldığında veya kamera anormal bir aktivite algıladığında, sistem için geliştirilen algoritmalar çalışarak kişinin görüntüsü yakalanmaktadır. Görüntü işleme için Python ve OpenCV kullanılarak arka plan çıkarma algoritması uygulanmaktadır. Yakalanan görüntülerdeki yüzler algılanır ve Haar özellikleri kullanılarak ayrıştırılmaktadır. Ayrıştırma işlemi sonrasında, algılanan yüzler ile veri tabanında kayıtlı yüzler karşılaştırılmaktadır. Yüz tanıma işlemi için PCA ve SVM algoritmaları kullanılmaktadır. Eğer tanıma işlemi başarılı olursa yani algılanan yüz veritabanına kayıtlı ise Raspberry Pi kapıyı otomatik olarak açar ve ev sahibine Android uygulama üzerinden kimin girdiğinin bilgisini vermektedir. Eğer yüz tanıma işlemi başarısız olursa, ev sahibine tanınmayan kişi ile ilgili bildirim gönderilir ve telefonda canlı video akışı başlamaktadır. Ev sahibi, tanınmayan kişi için uygulama üzerinden kapıyı açabilir veya uygulama arayüzü üzerindeki farklı bir buton ile polise haber verebilmektedir. Geliştirilen bu sistemin özellikle büyük şehirlerde evlerinde yalnız yaşayan yaşlı veya engelli bireyler için büyük fayda sağlayabileceği düşünülmektedir, çünkü bu kişilerin evlerine hırsızların saldırma olasılığı daha yüksektir. Bu sistem, güvenliklerini artırmak ve acil durumlara hızlı yanıt vermek isteyen kullanıcılar için ideal bir çözüm sunmaktadır [19].

Munir ve ark. tarafından gerçekleştirilen çalışmada, akıllı evler için bir güvenlik sistemi önerilmektedir. Vaidya ve ark. tarafından yapılan çalışmaya benzer olarak bir kapı izleme sistemi üzerine odaklanılmaktadır [19]. Bu sistemi diğer sistemlerden ayıran en büyük özelliği ise sistem, çevrimdışı çalışabilmektedir. Bunun yanı sıra önerilen sisteme sesli komut özelliği de entegre edilerek evdeki cihazların ses ile kontrol edilebilmesi sağlanmaktadır. Geliştirilen sistemde yüz tanıma işlemi için kapıya yerleştirilen bir kamera ve Raspberry Pi kullanılmaktadır. Öncelikli olarak sisteme tanıtılan yüzler Raspberry cihazında oluşturulan bir veritabanında

saklanmaktadır. Daha sonra kapıya yerleştirilen kamera ile gerçek zamanlı olarak video çekilmektedir. Kamerada algılanan kişinin yüzü Haar Cascade sınıflandırıcı ile tespit edilmektedir. Haar Cascade sınıflandırıcı sayesinde, kamera karşısındaki kişinin gözleri, ağzı, burnu ve kulakları tespit edilebilmektedir. Algılanan yüz veritabanındaki kayıtlı yüzler ile eşleşirse kapı açılmaktadır. Eğer algılanan yüz veritabanı ile eşleşmezse evin içine yerleştirilen buzzer'lar alarm vererek ev sahibini uyarmaktadır. Önerilen sistemde yüz tanıma işlemi için OpenCV kütüphanesi kullanılmaktadır. Tasarlanan algoritmada bir eşik değeri tanımlanmıştır ve gerçek zamanlı algılanan yüz ile veritabanında yer alan yüzler karşılaştırılır ve aralarındaki fark bulunmaktadır. Eğer bu fark tanımlanan eşik değerinden küçükse algılanan yüz ile veritabanında karşılaştırılan yüzün aynı olduğu kabul edilmektedir. Geliştirilen sistemde eşik değerleri üzerinde yapılan yüz tanıma testlerinde, 40 olarak belirlenen eşik değeriyle %90'lık bir başarı sağlanmaktadır. Yüz tanıma performansını artırmak için eşik değerine ek olarak diğer faktörler de göz önünde bulundurmak gerekmektedir. Veritabanında, her tanımlı kişi için kullanılan 20 görüntü yerine her kişi için 100 farklı görüntü kullanılarak, sistem geniş bir eşleşme yeteneği elde etmektedir. Ayrıca, gün boyunca kapıdaki ışık farklılıklarını telafi etmek için veritabanı, çeşitli pozlar ve ışıklandırmalar içeren görüntülerle eğitilmiş ve bu durum sistem hızını ve gecikmeyi oldukça azaltmaktadır. Tasarlanan sistemde sesle kontrol için V3 modülü kullanılmaktadır ve bu modül maksimum 80 komut saklayabilmektedir. Bu durum, aynı komutun farklı seslerle eğitilmesini kısıtlamaktadır. Sistem, standart aksan ve ortalama bir tonla eğitilerek iyileştirilmiştir. Ayrıca, çevresel gürültü ve bozulmalardan kaynaklanan arızaları önlemek için gürültülü ortamlarda eğitim yapılmıştır. Bu tasarlanan sistem, yeni nesil bir güvenlik sistemi geliştirme amacı taşımaktadır. İnternet bağımlılığı olmaksızın çalışabilmesi, internet erişiminin olmadığı bölgelerde bile kolaylıkla kullanılabilir olmasını sağlamaktadır [20].

Bhanse ve Jaybhave tarafından yapılan çalışmada, yüz tanımlama teknolojisinin otomotiv sektöründeki olası kullanım alanlarını incelemektedir. Yüz algılama ve tanımlama işlemi için Raspberry Pi ve OpenCV kullanılmaktadır. Geliştirilen sistemde, yüz tanımlama ile otomobilde torpido gözü ve benzeri bölümlerin servo motorla ile otomatik olarak açılıp kapanmasını sağlamaktadır. Yüz tanıma sürecinde ayrıca yüzün konum koordinatları da belirlenir. Haar Cascade

sınıflandırıcısı, bu sistemde yüz tanıma işlemi için ana bileşen olarak kullanılmaktadır. Tasarlanan sistemin çalışması; kamera sürekli olarak görüntü almaktadır. Kamera karşısına bir kişi geçtiğinde görüntü işleme işlemi başlar kişinin yüz bölgesi belirlenmektedir. Daha sonra belirlenen yüz Haar Cascade dosyasında kaydedilmiş yüzler ile kıyaslanmaktadır. Eğer sistem yüzü tanımlarsa bir sonraki aşama olan yüz koordinatlarını belirler. Yüz, kamera karşısında merkezlenmiş bir durumda ise Raspberry Pi cihazından Arduino'ya bir sinyal yollar ve Arduino da servo motorun çalışması için gerekli algoritmayı çalıştırmaktadır. Geliştirilen bu sistemdeki en büyük zorluk kameranın FPS hızıdır. Çünkü geliştirilen sistemde yüzün koordinatları da hesaplandığı için görüntü işleme işleminin hızlı ve doğru yapılması gerekmektedir. Bunun içinde FPS hızının optimum seviyede tutulması gerekmektedir. Optimum FPS değerine ulaşmak için birçok çözünürlükte sistem test edilmektedir. 1920 x 1080 çözünürlüğünde yapılan testlerde, sistem, kamera önünde herhangi bir yüz algılamadığında istikrarlı bir şekilde 30 FPS ile çalışmaktadır. Ancak, bir yüz algılandığında FPS değeri 15'e düşmekte ve bu durum, sistemin düşük performans göstermesine yol açmaktadır. Test sonuçlarına göre, 320 x 240 çözünürlüğünün tercih edilmesi gerektiği belirlenmektedir. Bu çözünürlük, yüz algılandığında 25 FPS hızında kararlı bir şekilde çalışabilen bir sistem sunmaktadır. Bu çalışma, otomotiv endüstrisinde kullanılabilecek yeni bir yüz tanıma sistemi tasarımının temellerini oluşturmaktadır [21].

Sharma ve ark. tarafından yapılan çalışmada, bir akıllı ayna tasarımı önerilmiştir. Raspberry Pi kullanılarak geliştirilen akıllı ayna, farklı sitelerden API anahtarları ile hava durumu, saat, tarih, haber başlıkları gibi güncel bilgileri ayna üzerinde gösterme imkânı sunmaktadır. Bu aynaya entegre edilen kamera ve OpenCV kütüphanesi sayesinde, kullanıcıların tanımlanması ve kişiselleştirilmiş deneyimler elde edilmesi mümkün olmaktadır. Ayrıca mikrofon entegrasyonu sayesinde kullanıcıların sesli komutlarla ayna üzerindeki bilgilerin görünürliğini düzenlemesine ve tercih ettikleri ayarlamaları kolaylıkla gerçekleştirmesine olanak tanımaktadır. Ek olarak, bu ayarlar kullanıcı bazında kaydedilerek, kullanıcı sistem tarafından tanındığında önceki tercihlerine uygun bir şekilde ayna üzerindeki içerikleri görebilmektedir. Bu özellikler sayesinde, kullanıcılar ayna karşısına geçtiklerinde

kişisel tercihlerine uygun ve tanıdık bir deneyim elde etmekte ve ayna üzerindeki içerikleri kişiselleştirebilmektedir [22].

Kadhim ve ark. tarafından yapılan çalışmada, Alzheimer hastalarının çevrelerindeki insanları tanımalarına yardımcı olmak için gerçek zamanlı bir yüz tanıma uygulaması önerilmektedir. Bu bağlamda, ESP32-CAM geliştirme kartlı bir gözlük ve Raspberry Pi tabanlı tek kartlı bir mikrobilgisayarın entegre edildiği taşınabilir bir sistem geliştirilmektedir. Bu sistem, kablosuz teknolojiler aracılığıyla gerçek zamanlı video karelerini yakalar ve Wi-Fi üzerinden Raspberry Pi'ye aktarmaktadır. Raspberry Pi, gelen videolardaki insan yüzlerini algılayıp tanımlamaktadır. Tanınan yüzlerin isimleri, kullanıcının kulaklarına Bluetooth üzerinden sesle etkinleştirilen geri bildirim olarak iletilmektedir. Yüz tanıma işlemi için, MobileNetV2 gibi düşük karmaşıklığa sahip ve düşük kaynak tüketen bir model kullanılmaktadır. Ayrıca, yüz tanıma doğruluğunu artırmak için daha gelişmiş algoritmalar da entegre edilmektedir. Bu amaçla, DenseNet modeli tercih edilmektedir. Yapılan deneyler, bu sistemle %99,46 eğitim doğruluğu ve %99,48 yüz tanıma doğruluğu elde edildiğini göstermektedir. Geliştirilen sistem, mobil cihazlara kıyasla daha düşük maliyetli ve daha taşınabilir bir çözüm sunarken, uygulama süresinin cihazın işlem gücüne bağlı olarak değişmesi gibi bazı zorlukları da beraberinde getirmektedir. Ancak yapılan deneyler, uygulama sürelerinin genellikle kabul edilebilir düzeyde olduğunu ortaya koymaktadır. Bütün prosesin tamamlanma süresi 1109 ms olarak belirlenmektedir. Bu, sistemin etkin bir şekilde çalıştığını ve gereksinimlerini karşıladığını göstermektedir. Sonuç olarak, geliştirilen bu teknolojik çözümün, Alzheimer hastalarının yaşam kalitesini artırması ve onları diğer insanlara daha az bağımlı hale getirmesi beklenmektedir. Geliştirilen uygulama, güvenilirlik, taşınabilirlik ve kullanım kolaylığı gibi avantajlar sunarak bu hedefe ulaşmada etkili bir araç olabilmektedir. Ayrıca, hastaların ve bakıcılarının yaşadığı günlük zorlukları azaltarak yaşam deneyimlerini iyileştirmeye yönelik bir adım olarak da değerlendirilebilmektedir [23].

Lin ve Li tarafından geliştirilen sistem, günümüzde manuel yoklamanın yerini alan elektronik yoklama sistemlerinin verimliliğini artırmayı hedeflemektedir.

Önerilen yüz tanıma tabanlı yoklama yöntemi, yoklamanın verimliliğini artırarak bu amaca hizmet etmektedir. Sistem, kamera terminali ve sunucu bilgi işlem terminali olmak üzere iki bölümden oluşmaktadır. Kamera terminali, sınıfta gerçek zamanlı öğrenci portre videoları toplar ve bu verileri sunucuya ileterek işlenmesini sağlamaktadır. Sunucunun temel işlevi, kameradan gelen insan görüntüsünün video bilgilerini parçalara ayırmak, her parçayı analiz etmek, bu parçaları insan yüzlerini tanımak için işlemek ve ardından tanıma sonuçlarını filtrelemek ve iyileştirmektir. Yapılan test sonuçlarına göre, tespit edilebilen yüz sayısının her zaman tanınabilen yüz sayısından daha fazla olduğu belirlenmiştir. Bu durumun temel sebebi, kameranın sınıfın ön tarafına yerleştirilmesidir. Bu durum, arka sıradaki öğrencilerin yüzlerinin tanınamamasına neden olan bulanık görüntülere yol açmaktadır. Bu problemi çözmek adına, daha net görüntüler elde etmek için kameranın hareketini kontrol etmek için bir platform geliştirilmiştir. Platformun ana amacı, kamerayı sınıfın her yerini görebilecek şekilde konumlandırarak, tüm alanları kapsamaktır. Daha sonra sınıf, dört parçaya ayrılarak işlenir ve bu şekilde tanıma doğruluğu artırılır. Tanıma oranı belirli bir seviyeye ulaşmadığında, sunucu her bir alanı daha ayrıntılı bir şekilde işlemek için ikiye bölerek, yüz tanıma doğruluğunu daha da artırabilmektedir. Tasarlanan sınıf yoklama sistemi, sınıfta gerçek zamanlı olarak işlev görür ve manuel yoklamanın neden olduğu zaman kaybını önlemektedir. Sistem, yoklamayı denetlemek için yüz tanıma teknolojisini kullanmaktadır. Öğrenciler sınıfta dikkat dağınıklığı yaşarlarsa veya uzun süre cep telefonlarıyla meşgul olurlarsa, tanınmama ihtimalleri vardır ve bu da sonradan devamsızlık olarak değerlendirilebilmektedir. Bu nedenle, sistem öğrenciler üzerinde etkili bir denetim mekanizması sağlar ve böylelikle sınıf içi öğrenim verimliliğini artırabilmektedir [24].

Suchitra ve ark. tarafından yapılan çalışmada, yüz görüntülerinden gerçek zamanlı duygu tanıma için bir yöntem önerilmektedir. Önerilen yöntemde üç aşamalı bir yaklaşım kullanılmaktadır. Haar Cascade ile yüz algılama işlemi, ASM ile özellik çıkarma yapılırken öfke, iğrenme, mutluluk, nötr ve şaşırma gibi beş duygunun sınıflandırılması için Adaboost sınıflandırıcısı kullanılmaktadır. Bu yöntemin amacı, yüz görüntülerinden öfke, iğrenme, mutluluk, sürpriz ve nötr gibi temel duyguları gerçek zamanlı olarak tanıyabilmektir. Bu amaçla veritabanında, 337 farklı denekten oluşan bir veri seti kullanılmaktadır. Deneklerin görüntüleri farklı aydınlatma

seviyelerine ve pozlara sahip olup 235 erkek ve 102 kadından oluşmaktadır. Öncelikle giriş görüntüsü web kamerası aracılığıyla yakalanmaktadır. Ardından, Viola-Jones yöntemi (Haar Cascade Sınıflandırıcı) ile yüz algılanır ve görüntü işlenmek üzere kaydedilmektedir. Görüntü ön işlemede, görüntü boyutlandırılır, gri tonlama yapılır ve gürültü giderici filtreler uygulanır. Özellik çıkarımı, ASM kullanılarak gerçekleştirilir ve sınıflandırma AdaBoost ile yapılmaktadır. Sonuç olarak elde edilen sistem, %94 genel doğruluk sağlamakta ve Raspberry Pi 2 gibi düşük ağırlıklı donanımlarda kullanılabilecek düzeyde performans sunmaktadır. Önerilen bu sistem, duygu tanınmanın farklı uygulamalardaki önemli rolü göz önünde bulundurularak, toplum için oldukça yararlı bir araç olarak değerlendirilebilmektedir. Gelecekte yapılacak çalışmalarda, tanıma doğruluğunu artırmak için çeşitli algoritmaların uygulanması ile geliştirilebilmektedir [25].

Fernandez ve ark. tarafından yapılan çalışmada, mobil cihazlar için tasarlanmış akıllı bir fotoğraf paylaşım uygulaması önerilmektedir. Bu uygulama, güçlü ve verimli bir yüz tanıma sistemine dayanmaktadır. Sistem, elde edilen biyometrik bilgileri telefon rehberiyle ilişkilendirerek video veya kaydedilmiş fotoğraflardan kişileri tanımlamaya olanak sağlamaktadır. Temel olarak, uygulama kişilerin biyometrik bilgilerini telefon rehberiyle eşleştirir ve bu bilgileri otomatik olarak etiketleyerek görüntüleri kişilere e-posta yolu ile gönderebilmektedir. Yüz algılama için Haar Cascade ve LBP tabanlı yöntemler kullanılarak gerçekleştirilen işlem, LBP'nin daha hızlı ancak önden olmayan yüzlerde algılama sorunları yaşama potansiyeli nedeniyle Haar Cascade detektörüyle dengelenmektedir. Uygulamanın temel amacı, kullanıcıların fotoğrafları videoları kişilerle kolayca paylaşmasını sağlamaktır. Kayıt ve tanımlama olmak üzere iki ana aşamadan oluşmaktadır. Kayıt aşamasında, LBP şablonları yüz görüntülerinden çıkarılır ve telefon rehberiyle ilişkilendirilir. Tanımlama aşamasında, seçilen fotoğraftaki yüzler tespit edilir ve önceden kaydedilmiş kişiler tanımlanmaktadır. Uygulama, ayrıca bilinmeyen kişilerin etiketlenmesine ve biyometrik profillerin saklanmasına olanak tanımaktadır. Geliştirilen bu uygulama, yüz tanıma sisteminin mobil cihazlarda önemini vurgulamaktadır. Ayrıca, kullanıcıların fotoğraf paylaşımını daha etkili ve özgün hale getirerek dijital iletişim deneyimini geliştirmektedir [26].

Karuppasamy ve ark. tarafından yapılan çalışmada, geleneksel ATM sistemlerine alternatif olarak akıllı bir otomatik vezne makinesi için yenilikçi bir doğrulama yöntemi önerilmektedir. Bu yöntem, müşterilerin kimlik doğrulamasını geleneksel kart ve PIN kodu kullanımı yerine yüz tanıma teknolojisiyle gerçekleştirmelerini sağlamaktadır. Müşterinin kimliğini doğrulamak için, ATM'ye yerleştirilen kamerayla çekilen bir resim ile veri tabanındaki resimler karşılaştırılır ve gerçek müşteri belirlenirken, sistem güncel resimlerle modelini güncellemektedir. Doğrulama sürecinde, resim işleme adımları arasında OpenCV ve Haar Cascade Sınıflandırıcı kullanılırken, yüz tanıma işlemi LBP algoritması kullanılarak gerçekleştirilmektedir. Yüz algılama ve işleme adımları Python programlama dili ile gerçekleştirilir ve çeşitli aydınlatma koşullarında çekilen resimlerin iyileştirilmesi için Görüntü İyileştirme prosedürü uygulanmaktadır. Son olarak, netleştirilen resimler JPEG formatında kaydedilir ve yüz tanıma işlemi Fisher yüz hesaplaması kullanılarak tamamlanmaktadır. Bu yöntemle, kullanıcılar eğer sistem tarafından tanımlanırsa hesaplarına kart kullanmadan giriş yapabilir ve gerekli işlemlerini gerçekleştirebilmektedirler. Yeni geliştirilen bu yöntem, geleneksel biyometrik sistemlere kıyasla ATM güvenliğini artırmak ve soygunları önlemek için daha güvenilir bir seçenek sunmaktadır. Önerilen sistem, geleneksel ATM sistemlerinin dezavantajlarını azaltarak kart üretim maliyetlerinden tasarruf sağlar ve kullanıcıların kart kaybetme veya dolandırıcılık tehditleriyle karşılaşma riskini azaltmaktadır [27].

Nagari ve ark. tarafından önerilen akıllı ev otomasyon sistemi, Raspberry Pi ve merkezi bir HUB'a bağlı bir dizi IoT cihazını içermektedir. Bu cihazlar, evin içindeki insan hareketlerini ve varlığını izlemek ve çeşitli eylemleri gerçekleştirmek için sensörler, kameralar ve aktüatörler gibi çeşitli bileşenleri barındırmaktadır. Sistemde kullanılan YOLO nesne algılama algoritması, insanların varlığını ve konumunu tespit etmek için temel bir araç olarak kullanılmaktadır. YOLO'nun IoT cihazlarıyla entegrasyonu sayesinde, sistem insan varlığına bağlı olarak otomatik eylemler gerçekleştirebilmektedir. Sistemin prototipi, ampul ve fan gibi cihazlar kullanılarak çeşitli senaryolarda test edilmiştir. Örneğin, kameranın YOLO algoritması sayesinde bir insan algılandığında, oda parlaklığı kontrol edilir ve düşükse, otomatik olarak ışıklar açılabilir. Benzer şekilde, odada bir insan algılandığında sıcaklık sensörü, odanın sıcaklığını ölçer ve gerektiğinde fanı otomatik olarak devreye sokmaktadır.

İnsan algılanmadığında ise ışıklar kapanır ve fanın hızı düşürülerek enerji tasarrufu sağlanmaktadır. Gelecekte, sistemin yeteneklerini artırmak için daha karmaşık nesneleri tanımlama ve farklı aktiviteleri algılama kapasitesini geliştirecek yeni nesne algılama algoritmalarını entegre etme potansiyeli bulunmaktadır. Ayrıca, kullanıcı davranışlarını analiz etmek ve öngörmek için makine öğrenimi tekniklerinin sisteme entegrasyonu, otomasyonun daha kişiselleştirilmiş ve verimli hale getirilmesine olanak sağlayabilmektedir [28].

Shu ve Hu tarafından yapılan çalışmada, yapay zeka ile insan duruşunu tanımlama yeteneğine sahip olan bir sistem geliştirilmiştir. Bu sistem; oturma, ayakta durma ve düşme gibi çeşitli duruşları doğru bir şekilde sınıflandırabilmektedir. Yapılan deneylerde, önerilen yaklaşımın YOLOv5 modeli ile desteklendiği ve insan duruşunu yüksek doğrulukla tanımladığı görülmektedir. Yapılan karşılaştırmalı analizlerde, YOLOv5'in öncül modelleri olan YOLOv3 ve YOLOv4'e kıyasla daha yüksek doğruluk ve daha hızlı işlem süreleri sunduğu belirlenmiştir. Bu nedenle, YOLOv5 modeli seçilmiş ve önerilen yaklaşımın temelini oluşturmaktadır. Bu gelişme, akıllı ev teknolojileri ve sağlık sektörü için önemli bir ilerlemedir. Özellikle yaşlı bakımında, düşmelerin otomatik olarak tespit edilmesi ve hızlı müdahale için alarm sistemleri gibi özellikler, yaşam kalitesini artırmak ve güvenliği sağlamak için kritik bir rol oynayabilmektedir. Bu sistem, yaşlı bireylerin güvenliğini artırmak için akıllı ev sistemlerine başarılı bir şekilde entegre edilebilmektedir. Ayrıca, spor ve egzersiz uygulamalarında doğru duruşun izlenmesi veya ofis ortamlarında ergonomik çalışma pozisyonlarının takibi gibi farklı alanlarda da kullanılabilecek geniş bir potansiyeli vardır [29].

Hwang ve Wahyutama tarafında geliştirilen sistemde yapay zeka destekli bir çöp kutusu geliştirilmektedir. Geliştirilen sistem, Raspberry Pi 4 üzerinde çalışan ve metal, kağıt, plastik gibi geri dönüştürülebilir çöpleri tanımlayabilen bir YOLO nesne algılama modelini kullanarak bu çöpleri otomatik olarak ayıran ve toplayan bir çöp kutusu olarak tasarlanmaktadır. Sistemin temelinde, bir web kamerası, geri dönüştürülebilir malzemelerin video görüntülerini yakalar ve YOLO nesne algılama modeli aracılığıyla bu malzemeleri sınıflandırmak için giriş verisi olarak kullanır.

Nesneler tespit edildikten sonra, bir servo motor, tespit edilen malzemelere göre çöp kutusu kapağını uygun bölmeye hizalamaktadır. Ayrıca, her bölmenin üzerine, çöp kutusunun doluluk durumunu belirlemek için ultrasonik sensör ve çöp kutusunun konumunu belirlemek için bir GPS modülü yerleştirilmektedir. Bu veriler, Arduino Uno tarafından toplanır ve ardından ESP8266 Wi-Fi modülü aracılığıyla Firebase veri tabanına gönderilmektedir. Firebase veri tabanından alınan veriler, bir mobil uygulama aracılığıyla gerçek zamanlı olarak görüntülenebilmektedir. Sistem geliştirilirken, Raspberry Pi'nin küçük boyutu ve taşınabilirliği gibi avantajları göz önünde bulundurularak YOLOv4-tiny modeli tercih edilmiştir. Bu modelin, testler sonucunda % 75 doğruluk elde ettiği belirlenmiştir. Ancak, geliştirilen sistem bir bilgisayara yüklendiğinde ve tam boyutlu YOLOv4 modeliyle çalıştırıldığında, model daha az sayıda veri ile eğitilmiş olmasına rağmen % 91 oranında doğruluk elde etmiştir. Raspberry Pi'nin sınırlı işlem gücünden dolayı YOLOv4-tiny modeli tercih edilmektedir. Fakat Raspberry Pi'nin küçük boyutu, taşınabilirliği ve ekonomik yapısı gibi avantajları ile doğruluk oranının % 16 daha az olmasına rağmen saha testlerinden elde edilen sonuçların uygun olması, Raspberry Pi'nin ideal bir seçenek olmasını sağlamaktadır. Bu sistemle, çevresel kirliliğin azaltılması, geri dönüştürülebilir malzemelerin verimli bir şekilde toplanması, çöp kutularının doluluk durumlarının gerçek zamanlı olarak izlenmesi ve böylelikle çevre dostu bir yaşam alanının sağlanması gibi önemli faydalar elde edilmektedir [30].

Sharrar ve Abas tarafından yapılan projede, akıllı bir klima sistemi tasarımı önerilmektedir. Bu sistem, Python programlama dili ile görüntü işleme, derin öğrenme ve yapay zeka alanlarında kullanılan kütüphaneler olan OpenCV, TensorFlow ve Keras ile geliştirilmektedir. Temel olarak, odadaki insanların faaliyetlerini izlemek ve termal gereksinimlerini belirlemek için bir bilgisayarla görüş algoritması kullanılmaktadır. Bu süreçte, çeşitli sensörler aracılığıyla oda koşulları da ölçülmektedir. Toplanan veriler, IoT aracılığıyla klima kontrolörüne iletilerek oda sıcaklığının otomatik olarak düzenlenmesi sağlanmaktadır. Ana hedef, odanın termal koşullarını sakinlerin ihtiyaçlarına uygun şekilde optimize ederek elektrik israfını azaltmak ve enerji maliyetlerini düşürmektir. Bu doğrultuda, YOLO ve OpenPose gibi derin öğrenme algoritmaları, insanların giyim durumları ve pozisyonları gibi faktörleri dikkate alarak doğru termal ayarlamalar yapmak için kullanılmaktadır.

YOLO algoritması, farklı kıyafetlerdeki insanları tanıyarak, sistemin insan algılaması yaparken aynı zamanda giyim durumunu sınıflandırabilme yeteneğini kazanmasını sağlamaktadır. OpenPoses kütüphanesi ise insan iskelet sistemini tanımlayarak, algılanan kişinin ayakta mı yoksa oturma pozisyonunda mı olduğunu belirleyebilmektedir. Geliştirilen bu projede kullanılan Raspberry Pi gibi sınırlı işlem gücüne sahip cihazlar, sistem performansında kısıtlamalara neden olabilmektedir. Bu durumda, daha güçlü donanım kullanılması, sistemin performansını artırabilir ve daha verimli bir çalışma sağlayabilmektedir. Enerji verimliliğini artırarak elektrik israfını azaltmayı ve kullanıcıların termal konforunu optimize etmeyi amaçlayan bu sistem, önemli katkılar sağlama potansiyeline sahiptir [31].

Huang ve Hao tarafından yapılan çalışmada, CNN tabanlı görsel tanıma özellikli akıllı klima sistemi önerilmektedir. Önerilen bu sistem, gelişmiş bir görüntü tabanlı algılama teknolojisi ve yapay zeka algoritmaları ile donatılmıştır. Kullanıcıların hareketlerini, durumlarını ve çevresel koşulları sürekli olarak izleyerek klimaların çalışma parametrelerini optimize edebilme kabiliyetine sahiptir. Akıllı bina uygulamaları için YOLO evrimsel sinir ağı teknolojisinin kullanıldığı bir görüntü tanıma yöntemi sunulmaktadır. Bu yöntem ile iç mekânda bulunan insanları ve konumlarını %98'e varan bir doğrulukla tespit edilebilmektedir. Böylece, klimalar sadece mevcut alanlarda bulunan insanlara uygun soğutma hizmeti sunabilir, bu da enerji tasarrufu sağlamaktadır. Ayrıca, sistem geleneksel PIR sensörleri ve CO₂ sensörlerinin zorlandığı durumlarda bile yüksek hassasiyetli algılama sağlayabilmektedir. Hayvanlardan kaynaklanan yanlış algılamaları da filtreleyerek daha optimal bir çalışma sağlamaktadır. PIR sensörleri, ses algılayıcıları ve CO₂ sensörlerinin aksine sabit veya sessiz kişilerin tespit edilmesinin yanı sıra, sistem aynı zamanda kullanıcıların konumlarına göre mesafe ve yön tahmini yapabilmektedir. Bu özellikler, klimaların kullanıcıların varlığı ve konumu gibi dinamik faktörlere duyarlı bir şekilde çalışmasını sağlayarak mekânın enerji verimliliğini optimize etmektedir. Bu yaklaşım, sadece enerji tasarrufu sağlamakla kalmaz, aynı zamanda kullanıcı deneyimini ve çevresel sürdürülebilirliği de artırmaktadır. Sonuç olarak, bu çalışma, iç mekân iklim kontrolünde farklı bir yaklaşım sunarak enerji verimliliğini artırmak ve sürdürülebilir bir gelecek için önemli bir kilometre taşı olmaktadır [32].

3. MATERYAL VE YÖNTEMLER

Bu bölümde, akıllı klima sistemi için kullanılan materyallerin seçimi, hazırlığı ve kurulum adımları incelenmektedir. Ayrıca, yüz tanıma ve insan algılama algoritmalarının seçimi ve bu algoritmaların nasıl uygulandığı bu bölümde açıklanmaktadır. Materyal ve yöntemlerin doğru bir şekilde seçilmesi ve uygulanması, araştırmanın güvenilirliği ve sonuçlarının doğruluğu için hayati önem taşımaktadır. Bu bölüm, projenin teknik yönlerini netleştirerek araştırmanın nasıl yapıldığına dair kapsamlı bir bakış sunmaktadır.

3.1. Materyaller

Bu bölümde geliştirilen projede kullanılan materyaller incelenmektedir. Bu materyaller arasında yazılım araçları, donanım bileşenleri ve veri kaynakları bulunmaktadır. Her bir materyal, projenin amaçlarına uygun şekilde seçilmiş ve kullanılmıştır. Ayrıca, bu materyallerin seçim süreci ve kullanımıyla ilgili detaylar da bu bölümde sunulmaktadır.

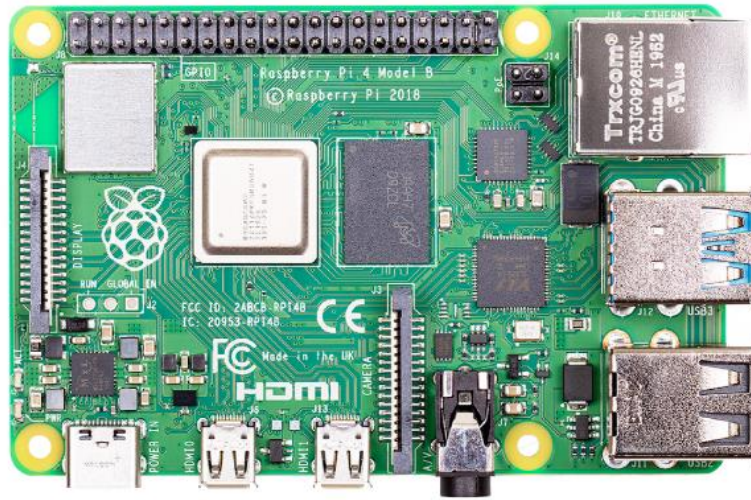
3.1.1. Raspberry Pi

Raspberry Pi, düşük maliyetli, yüksek performanslı ve genel amaçlı bir bilgisayar platformudur. 2012 yılından bu yana, Raspberry Pi Vakfı tarafından geliştirilmekte olan bu platformlar, ARM mimarisi üzerine kurulmuş ve genellikle Linux işletim sistemi ile çalışmaktadır. Raspberry Pi, eğitimden endüstriyel uygulamalara kadar geniş bir yelpazede kullanılmakta ve herkesin erişebileceği bir bilgisayar deneyimi sunmaktadır. Python, C/C++, Java gibi yaygın kullanılan dillerle uyumlu olması, Raspberry Pi'yi özellikle eğitim ve endüstriyel projeler için tercih edilen bir platform haline getirmektedir. Ayrıca, düşük güç tüketimi, küçük boyutu ve geniş bağlantı seçenekleri, projelerde esneklik sağlamak ve çeşitli uygulamalara olanak tanımaktadır [33].

Bu tez çalışmasında, Raspberry Pi 4'ün 8GB modeli tercih edilmiştir ve geliştirilen sistem için Linux tabanlı Raspian işletim sistemi kullanılmaktadır. Bu yeterli performanslı ve geniş bellek kapasiteli Raspberry Pi modeli, tez çalışmasında kullanılan görüntü işleme ve yapay zeka algoritmalarının çalıştırılmasına olanak tanımaktadır. Raspberry Pi 4'ün teknik özellikleri Tablo 3.1'de ve genel görünümü Şekil 3.1'de yer almaktadır.

Tablo 3. 1. Raspberry Pi 4 Teknik Özellikleri [33]

Özellik	Raspberry Pi 4 (8GB Model)
İşlemci	Broadcom BCM2711, dört çekirdekli Cortex-A72 (64-bit)
RAM	8GB LPDDR4 SDRAM
Bağlantı	2 x USB 3.0, 2 x USB 2.0, 2 x Micro HDMI, Ethernet
Depolama	MicroSD kart yuvası
İşletim Sistemi	Genellikle Linux tabanlı (Raspian)
GPIO	40 pinli Genel Amaçlı Giriş/Çıkış (GPIO)
Video Çıkışı	4K çözünürlükte 60Hz (2 x Micro HDMI)
Güç Girişi	5V USB-C port
Kablosuz Bağlantı	2.4GHz ve 5GHz 802.11b/g/n/ac Wi-Fi, Bluetooth 5.0
Boyutlar	85.6mm x 56.5mm x 17mm



Şekil 3. 1. Raspberry Pi 4 8GB [33]

Raspberry Pi 4'ün 8GB modelinin tercih edilmesinin arkasında çeşitli nedenler bulunmaktadır. Bu tez çalışmasında, son kullanıcı odaklı yapay zekalı bir sistem geliştirilmiş olup, donanım maliyeti ve kullanılabilirlik önemli faktörler olarak ön plana çıkmaktadır. Raspberry Pi 4, düşük maliyeti ve yeterli performansı ile bu tür sistemlerin prototiplerinin geliştirilmesi için ideal bir platform sunmaktadır. Geliştirilen sistemlerin üretim ve satış aşamasında, Raspberry Pi benzeri sınırlı işlem kapasitesine sahip, uygun maliyetli donanımlar kullanılacaktır. Alternatif olarak değerlendirilebilecek cihazlar arasında NVIDIA Jetson Nano ve Intel NUC bulunmaktadır.

NVIDIA Jetson Nano, yüksek GPU performansı ile öne çıkarken, maliyeti ve enerji tüketimi Raspberry Pi 4'e kıyasla daha yüksektir, bu da son kullanıcı odaklı ürünler için maliyet etkinliğini düşürmektedir [34]. Intel NUC ise yüksek işlem gücü ve geniş özellik yelpazesi sunmakla birlikte, maliyeti ve karmaşıklığı nedeniyle son kullanıcıya yönelik ekonomik bir çözüm olmaktan uzaktır [35]. Ayrıca, Raspberry Pi'nin geniş kullanıcı topluluğu, kapsamlı dokümantasyon desteği ve kolay erişilebilirliği, geliştirme sürecini önemli ölçüde kolaylaştırmakta ve teknik destek bulmayı mümkün kılmaktadır. Bu bağlamda, Raspberry Pi 4'ün tercih edilmesi, düşük maliyet, yeterli performans, enerji verimliliği ve kullanıcı dostu özelliklerinin birleşimi nedeniyle son kullanıcı odaklı yapay zekalı sistemlerin prototip geliştirme süreci için en uygun seçenek olarak öne çıkmaktadır.

3.1.2. Kamera

Işığa duyarlı dijital hücrelere, sensörler veya optik sensörler denir. Kamera sensörleri, görsel veriyi toplamak için optik objektiflerden gelen ışığı alır ve bunu elektronik sinyallere dönüştürmek için tasarlanmıştır. Bu sensörler, genellikle fotoğraf makinesi veya dijital kameralarda bulunur ve çeşitli boyutlarda ve çözünürlüklerde sunulmaktadır. Sensörler, içinde bulunan fotodiyot adı verilen ışığa duyarlı hücrelerden oluşur. Fotodiyotlar, ışığın etkisi altında elektronları serbest bırakır ve bu elektronlar daha sonra elektrik sinyallerine dönüştürülür. Sensörün çözünürlüğü, içindeki piksel fotodiyot sayısına bağlıdır. Örneğin, 10 megapiksel bir kamera sensörü,

içinde yaklaşık 10 milyon fotodiyot barındırır. Bu, sensörün daha yüksek çözünürlüğe sahip olduğu anlamına gelmektedir. Çünkü daha fazla piksel daha fazla ayrıntıyı ve görüntünün daha yüksek kalitede olmasını sağlamaktadır [36].

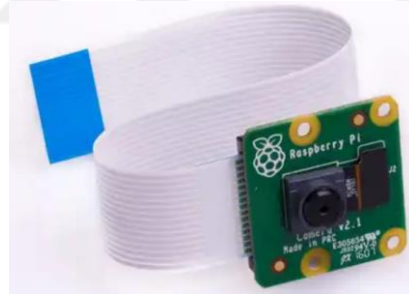
Neredeyse tüm dijital kamera ve ölçüm sistemlerinde kullanılan iki temel sensör türü vardır. Bunlardan biri CCD sensörü, diğeri ise CMOS sensörüdür. CCD sensörleri, görüntüyü elektronik sinyallere dönüştürmek için bir dizi yük bağlantılı kapasitör kullanırken, CMOS sensörleri ise aktif piksellerin bir matrisini içerir ve her piksel kendi amplifikatörüne ve diğer devre bileşenlerine sahiptir. CMOS sensörlerin avantajları arasında daha düşük güç tüketimi, daha küçük boyutlar ve daha az ısı üretimi bulunmaktadır. Bu nedenle, özellikle cep telefonları, tabletler ve diğer taşınabilir cihazlarda tercih edilmektedir. Ayrıca, CMOS sensörlerin üretim maliyetleri düşüktür ve seri üretimde kolaylık sağlar, bu da onları yaygın olarak kullanılan bir seçenek haline getirir. Ancak, CCD sensörlerine kıyasla CMOS sensörlerinin bazı dezavantajları da vardır. Örneğin, CMOS sensörlerin görüntü kalitesi ve dinamik aralığı genellikle CCD sensörlerine kıyasla daha düşüktür. Bununla birlikte, teknolojinin sürekli ilerlemesiyle birlikte, CMOS sensörlerinin performansı giderek artmaktadır [37, 38].

Yapılan kapsamlı literatür araştırması sonucunda bu tez çalışmasında CMOS sensörlü kameraların tercih edilmesindeki temel nedenler arasında düşük güç tüketimi, kompakt boyutları ve ekonomik avantajları ön plana çıkmaktadır. Bu bağlamda özellikle, Raspberry Pi kamera modülü v2'nin kullanımı, akıllı klima sisteminin gereksinimlerine ideal bir çözüm sunmaktadır. Raspberry Pi kamera modülü v2, sunduğu yüksek çözünürlük ve hassasiyetle çeşitli ışık koşullarında bile tutarlı performans sergileyebilme yeteneğine sahiptir. Bu modülün, Raspberry Pi'ye kolayca entegre edilebilmesi, uyum ve kolay bağlantı imkanları sayesinde sistem kurulumunu büyük ölçüde kolaylaştırmaktadır. Kompakt tasarımı, sınırlı alanlarda da rahatlıkla kullanılabilmesine olanak tanırken, yüksek kaliteli görüntü işleme yetenekleriyle istenilen performansı sağlamaktadır. Ayrıca, düşük güç tüketimi ve maliyet etkinliği, sistemin enerji verimliliğini artırırken, bütçe dostu bir çözüm sunmaktadır. Kamera modülünün sağladığı bu avantajlar, akıllı klima sisteminin performansını ve

güvenilirliğini artırmakta, kullanıcı deneyimini iyileştirmektedir. Bu özellikleriyle Raspberry Pi kamera modülü v2, geliştirilen sistem için uyumlu ve güvenilir bir çözüm sunmaktadır. Literatürde de benzer uygulamaların bu tür kameraları tercih etmesi, yapılan seçimin doğruluğunu desteklemektedir. Raspberry Pi kamera modülü v2'nin teknik özellikleri Tablo 3.2'de ve genel görünümü Şekil 3.2'de yer almaktadır.

Tablo 3. 2. Raspberry Pi Kamera v2 Teknik Özellikleri [33]

Özellik	Raspberry Pi Kamera v2
Çözünürlük	8 Megapiksel
Sensör Tipi	Sony IMX219 (CMOS)
Görüş Açısı	62,2°
Lens Çapı	3.04 mm
Video Çözünürlüğü	1080p (Full HD)
Boyutlar	25 mm x 23 mm x 9 mm
Bağlantı Arayüzü	CSI-2 (Camera Serial Interface 2)



Şekil 3. 2. Raspberry Pi Kamera v2 [33]

3.1.3. Sıcaklık Sensörü

Termistörler, sıcaklık ölçümünde kullanılan özel dirençli sensörlerdir ve termo-direnç etkisine dayanmaktadır. Termistörler, diğer sıcaklık sensörlerine kıyasla bir dizi avantaja sahiptir. Öncelikle, küçük boyutları sayesinde çeşitli uygulamalarda kolayca kullanılabilirler. Ayrıca, yüksek hassasiyetleri sayesinde hassas sıcaklık ölçümleri sağlamaktadırlar. Büyük nominal direnç değerleri, geniş sıcaklık aralığında çalışabilirliklerini artırırken, düşük maliyetleri de tercih edilme sebeplerindendir [39].

Genel olarak, termistörlerin sıcaklık-direnç ilişkisi, direnç sıcaklık katsayısı olarak bilinen bir özellik ile karakterize edilir. Bu katsayı pozitif veya negatif olabilir; Pozitif sıcaklık katsayılı termistörlerde direnç, sıcaklık artışıyla artarken, negatif katsayılı termistörlerde azalır. Pozitif katsayılı termistörler; ısıtma sistemlerinde, sıcaklık kontrol cihazlarında ve termal anahtarlarda kullanılırken, negatif katsayılı termistörler; elektronik cihazlarda, iklimlendirme sistemlerinde, otomotiv endüstrisinde, ev aletlerinde ve daha birçok alanda sıcaklık ölçümleri için yaygın olarak kullanılmaktadırlar [40].

Klima sistemlerinde; iç ortam sıcaklığı, dış ortam sıcaklığı, evaporatör sıcaklığı, deşarj sıcaklığı gibi bir dizi parametrenin hassas bir şekilde ölçülmesi önemlidir. Bu ölçümler, klima sistemlerinin doğru çalışması ve kullanıcı konforunun sağlanması için kritiktir. Bu nedenle, sıcaklık ölçümünde kullanılan sensörlerin doğruluğu ve güvenilirliği büyük önem taşımaktadır. Bu parametrelerin ölçümünde NTC termistörler kullanılmaktadır. Geliştirilen akıllı klima sisteminde de var olan klima ürünleriyle uyumlu olması amacıyla NTC termistörler tercih edilmektedir. NTC termistörler, geniş bir sıcaklık aralığında hassas ölçümler yapabilme yetenekleri ve uygun maliyetleriyle dikkat çekmektedir. Bu termistörler, klima sistemlerinin iç ve dış ortam sıcaklıklarını doğru bir şekilde izleyerek, enerji verimliliğini artırmaya ve kullanıcı konforunu optimize etmeye yardımcı olmaktadır. Projede kullanılan NTC termistörün teknik özellikleri Tablo 3.3'te ve genel görünümü Şekil 3.3'te yer almaktadır.

Tablo 3. 3. NTC Termistör Teknik Özellikleri [41]

Parametre	Değer
Oda Sıcaklığında (25 °C) Direnç Değeri	10k Ω
Sıcaklık Ölçüm Aralığı	-40 °C ile 100 °C
Sıcaklık Ölçüm Hassasiyeti (0 – 40 °C)	$\pm 0,5$ °C
β Değeri (Sıcaklık Katsayısı)	3950



Şekil 3. 3. NTC Termistör [41]

3.1.4. MCP3008 ADC Entegresi

Raspberry Pi üzerinde yer alan GPIO pinleri sadece dijital okuma ve yazma yapabilmektedir. Bundan dolayı pinler sadece 1 veya 0 olarak ayarlanabilmektedir. Raspberry Pi üzerinden analog sinyal okunmak veya yazmak istenildiğinde harici bir ADC entegresi kullanılması gerekmektedir [33]. Bu tez çalışmasında, Raspberry Pi ile NTC termistörü üzerinden sıcaklık verisinin ölçülebilmesi için MCP3008 ADC entegresi kullanılmaktadır.

MCP3008, Raspberry Pi gibi sadece dijital sinyal işleyebilen cihazların kullanım alanını genişletmek için önemli bir role sahiptir. Microchip Technology tarafından üretilen bir ADC entegresidir. Bu entegre, SPI arayüzü ile kontrol edilebilir ve 10-bit çözünürlükle sekiz adet analog giriş kanalına sahiptir. MCP3008'in çıkışı, 0 ila 1023 arasında bir sayısal değer üreterek analog giriş voltajını dijital veriye dönüştürebilmektedir. Bu entegre, düşük güç tüketimi, yüksek doğruluk ve geniş besleme voltajı aralığı gibi avantajlara sahiptir [42]. Sıcaklık, nem, ışık gibi analog değerlerin izlenmesi gereken uygulamalarda kullanılan bu entegre, Raspberry Pi'nin yeteneklerini genişleterek daha esnek ve verimli sistemlerin oluşturulmasına olanak tanıyabilmektedir. MCP3008 ADC entegresinin teknik özellikleri Tablo 3.4'te ve genel görünümü Şekil 3.4'te yer almaktadır.

Tablo 3. 4. MCP3008 ADC Entegresi Teknik Özellikleri [42]

Özellik	MCP3008 ADC Entegresi
ADC Çözünürlüğü	10 bit
Giriş Kanal Sayısı	8
Çalışma Voltaj Aralığı	2,7-5,5 VDC
5VDC ile Çalışırken Akım Tüketimi	Maksimum 500 μ A



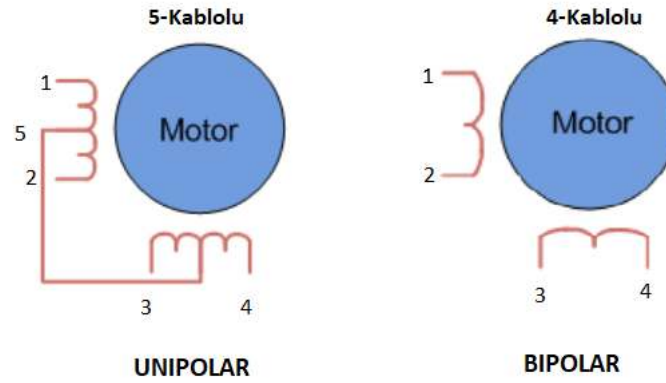
Şekil 3. 4. MCP3008 ADC Entegresi [42]

3.1.5. Step Motor

Endüstriyel otomasyon, robotik, elektronik cihazlar ve üç boyutlu yazıcılar gibi birçok alanda sıkça kullanılan step motorlar, dijital darbe sinyallerini doğrudan mekanik enerjiye dönüştürebilen özel motorlardır. Step motorlarının dönüşü, sürücü biriminden gelen PWM darbe sinyalleriyle gerçekleştirilmektedir. Her bir darbe, motorun belirli bir adımını oluşturur ve böylece motorun belirli bir açıda veya mesafede dönmesini sağlamaktadır. Aynı zamanda, step motorun hızı da PWM sinyal frekansının kontrolüyle düzenlenebilmektedir. Bu özellikleri sayesinde step motorlar, hassas pozisyon kontrolü gerektiren uygulamalarda ideal bir seçenek haline gelmektedir [43].

Step motorlar, sürücü topolojilerine, motor yapılarına ve adımlama modellerine göre sınıflandırılabilir. Özellikle, step motorlarının sürücü topolojisi,

motor seçiminde kritik bir faktördür. Bu bağlamda, unipolar (tek kutuplu) ve bipolar (iki kutuplu) olmak üzere iki ana topoloji vardır. Unipolar step motorlar, her biri ayrı bir bağlantıya sahip olan ve sabit manyetik kutupları bulunan bir rotor etrafında düzenlenmiş bobinlerden oluşmaktadır. Her bir bobin, sırasıyla etkinleştirilerek manyetik alanlar oluşturur ve rotoru hareket ettirir. Bu tip motorlar, basit bir kontrol devresi ile sürülebilirler, ancak genellikle daha düşük tork ve performans sunmaktadırlar. Bipolar step motorlarda ise, iki bobin ucunda bir bağlantı bulunur. Bu bobinler, sırayla etkinleştirilerek rotorun etrafında dönmesini sağlayan manyetik alanlar oluşturmaktadır. Bipolar motorların sürülmesi, bobinler arasındaki manyetik alanın ters yönde değiştirilmesiyle gerçekleştirilmektedir. Bu, daha karmaşık bir kontrol devresi gerektirir ancak daha yüksek tork ve performans sağlamaktadır [44]. Şekil 3.5'te unipolar ve bipolar step motor topolojilerinin şematik gösterimleri yer almaktadır.



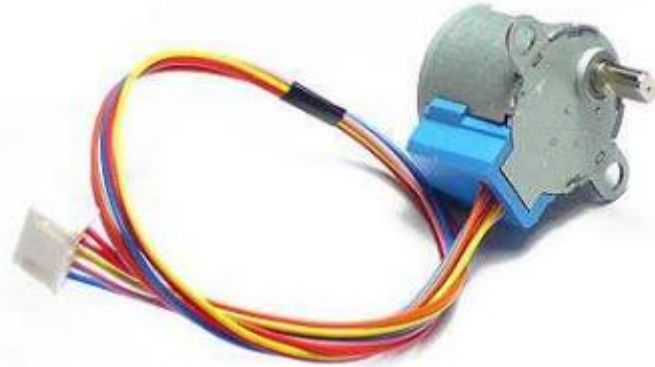
Şekil 3. 5. Step Motor Topolojileri [45]

Klimaların iç ünitelerinde, hava akışını yönlendirmek için yatay ve dikey kanatlar bulunur. Bu kanatların hareketi, unipolar step motorlar kullanılarak gerçekleştirilmektedir. Bu motorlar, basit sürücü devreleriyle kolayca kontrol edilebilir ve hava akışını yönlendiren kanatları hareket ettirmek için yeterli tork sağlayabilir. Üretim maliyeti avantajı ve sınırlı alanlara monte edilebilirlik gibi özellikleri sayesinde düşük maliyetli ve kompakt yapıdadırlar. Bu avantajlar, klimaların tasarımında ve üretiminde esneklik sağlamaktadır. Böylece daha verimli ve ekonomik çözümler elde edilebilmektedir. Geliştirilen akıllı klima sistemlerinde de bu

avantajlar göz önünde bulundurularak unipolar step motorlar tercih edilmiştir. Projede kullanılan 28BYC-48 ürün numaralı unipolar step motorun teknik özellikleri Tablo 3.5'te ve genel görünümü Şekil 3.6'da yer almaktadır.

Tablo 3. 5. Step Motor Teknik Özellikleri [46]

Özellik	28 BYC-48 Unipolar Step Motor
Çalışma Voltajı	5 VDC
Faz Sayısı	4
Hız Değişim Oranı	1/64
Adım Açısı	5,625°/64
Çalışma Torku	34,3 mNm
Sıkışma Torku	117,6mNm



Şekil 3. 6. Step Motor [46]

3.1.6. Step Motor Sürücüsü

Step motorların verimli çalışması için stator sargılarına uygun bir besleme gerilimi sağlanmalıdır. Motorun hareketi için gerekli olan faz sinyalleri, kontrol devresi programı tarafından belirlenen sırayla gönderilir. Bu sinyaller, motorun adım adım doğru şekilde hareket etmesini sağlamaktadır [47]. Geliştirilen sistemde, step motorlarını kontrol etmek için ULN2003A entegreli sürücü kartı kullanılmaktadır. Bu

entegre, step motorunun doğru şekilde sürülmesini sağlamak için faz sinyallerini güçlendirir ve motorun istenilen adımları atmasını sağlamaktadır. Ayrıca, sürücü entegresi, motorun hızını ve yönlendirilmesini kontrol etmek için çeşitli ayarlamalar yapılmasını mümkün kılan bir kontrol mekanizması sunmaktadır. ULN2003A entegreli step motor sürücü kartının teknik özellikleri Tablo 3.6'da ve genel görünümü Şekil 3.7'de yer almaktadır.

Tablo 3. 6. ULN2003A Sürücü Entegresi Teknik Özellikleri [48]

Özellik	ULN2003A Step Motor Sürücü Entegresi
Kontrol Lojik Voltajı	0-5 VDC
Çıkış Voltajı	5-15 VDC
Çıkış Akımı	500 mA (600 mA maksimum)



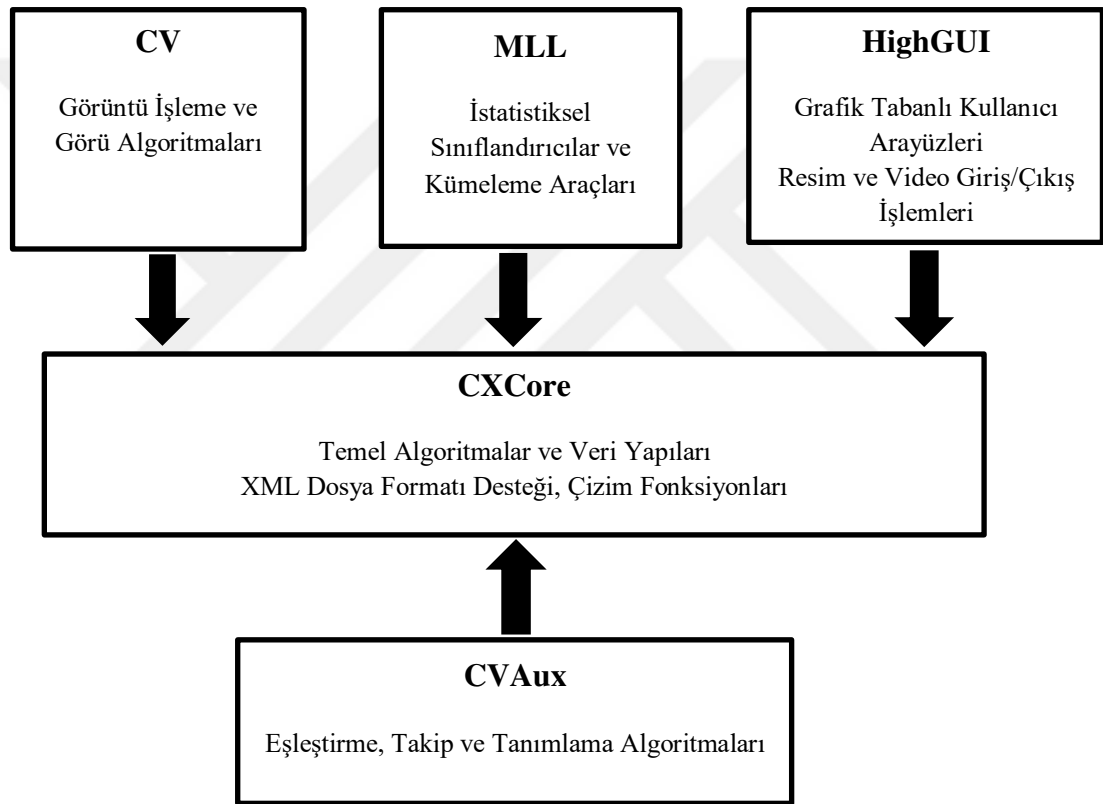
Şekil 3. 7. ULN2003A Step Motor Sürücü Kartı [48]

3.1.7. OpenCV Görüntü İşleme Kütüphanesi

OpenCV, Intel firması tarafından 1999 yılında geliştirilmeye başlanan ve o zamandan beri sürekli olarak yenilenen bir görüntü işleme kütüphanesidir. Bu kütüphane, resimlerden veya videolardan anlamlı bilgileri çıkarma ve işleme yeteneğine sahiptir [49]. Çeşitli programlama dilleriyle, özellikle de C, C++, Python ve Java ile yazılmıştır. OpenCV; Windows, Linux, MacOS, iOS ve Android gibi çeşitli işletim sistemleriyle uyumlu çalışabilmektedir. Bu kütüphane BSD lisansı ile lisanslanmıştır. Bu lisans türü, kodun kullanımı konusunda geniş bir özgürlük

sunmaktadır. BSD Lisansı altında, kullanıcılar OpenCV ile geliştirdikleri kodları istedikleri gibi kullanabilirler. Ayrıca, bu lisans altında OpenCV'nin hem akademik hem de ticari kullanımı ücretsizdir [50]. OpenCV, birçok farklı platformda kullanılabilirliği ve geniş özellik yelpazesıyla, görüntü işleme alanında tercih edilen bir kütüphanedir.

OpenCV kütüphanesi beş ana bileşenden oluşmaktadır. Bu bileşenler Şekil 3.8'de gösterilmektedir [51].



Şekil 3. 8. OpenCV Temel Bileşenleri [51]

CV bileşeni, temel resim işleme fonksiyonlarını ve yüksek seviyeli algoritmaları içeren beş ana bileşenden biridir. Computer Vision (Bilgisayar Görüşü) kavramının baş harflerinden oluşan bir kısaltmadır ve görüntüleri anlamlandırmak, işlemek ve yorumlamak için kullanılan bir alanı ifade etmektedir [51].

MLL, Machine Learning Library (Makine Öğrenimi Kütüphanesi) kelimesinin baş harflerinden oluşan bir kısaltmadır ve makine öğrenimi alanında kullanılan önemli bir bileşendir. Bu kütüphane, makine öğrenimi modelleri oluşturmak, eğitmek ve değerlendirmek için kullanılan çeşitli istatistiksel ve matematiksel fonksiyonları içermektedir. Veri analizi, sınıflandırma, regresyon, kümeleme ve boyut indirgeme gibi makine öğrenimi işlemlerini gerçekleştirmek için kullanılabilir. Ayrıca, MLL, büyük veri setlerinden bilgi çıkarmak ve tahminler yapmak için kullanıcı dostu araçlar sunmaktadır [51].

HighGUI, OpenCV'nin önemli bir bileşenidir ve grafik arayüzleri oluşturmak için kullanılır. Aynı zamanda, resim ve videoları işlemek için temel giriş/çıkış (I/O) işlevlerini içerir. Bu bileşen, kullanıcıların görüntü verilerini kolayca kaydetmelerini, yüklemelerini ve silmelerini sağlayan işlemlere de sahiptir. HighGUI, görüntü işleme projelerinde kullanıcı dostu bir deneyim sunmak için tasarlanmıştır [51].

CXCore bileşeni, OpenCV'nin önemli bir parçası olan ve Ip1Image, cvPoint, cvSize, cvMat, cvHistogram gibi temel veri yapılarını içerir. Bu bileşen, çeşitli veri yapılarını sağlayarak, görüntü işleme ve analiz uygulamalarında kullanılan verilerin depolanması, yönetilmesi ve işlenmesi için temel altyapıyı sunmaktadır. Ip1Image yapısal olarak, görüntü verilerini depolamak için kullanılır. cvPoint ve cvSize gibi yapılar, koordinatları ve boyutları temsil eder ve nesnelerin konumlarını ve boyutlarını belirlemek için sıklıkla kullanılır. cvMat görüntü işleme algoritmalarının işlem yapması için kullanılan veri matrislerini temsil eder. cvHistogram yapıları, görüntülerin renk dağılımını analiz etmek için kullanılır ve nesnelerin renk özelliklerini çıkarmak için önemlidir. Ayrıca, CXCore bileşeni XML verileriyle de etkileşimli olacak şekilde tasarlanmıştır, bu da veri depolama ve işleme süreçlerinde XML formatını kullanma olanağı sağlamaktadır. Özellikle eğitilmiş model oluşturulmak istendiğinde, modelin; yapısal bilgileri ve öğrenilmiş parametreleri bu dosya biçimi ile saklanmaktadır. Son olarak CvAux bileşeni, OpenCV kütüphanesinin içinde bulunan çeşitli deneysel algoritmaları içermektedir. Bu algoritmalar arasında şablon eşleştirme, şekil eşleştirme, nesnelerin ana hatlarını bulma, yüz tanıma, ağız hareketlerini izleme, vücut hareketlerini tanıma ve kamera kalibrasyonu gibi birçok

özellik bulunmaktadır. Bu özellikler, görüntü işleme ve analiz uygulamalarında kullanılarak geniş bir yelpazede işlevsellik sunmaktadır [51].

Geliştirilen akıllı klima sisteminde, görüntü işleme fonksiyonları için OpenCV kütüphanesi kullanılmaktadır. OpenCV'nin tercih edilme nedeni, sahip olduğu kapsamlı ve yüksek performanslı algoritmalar sayesinde görüntülerin anlamlandırılması, işlenmesi ve yorumlanmasında üstün bir yetkinlik sağlamasıdır. Ayrıca, OpenCV'nin sunduğu geniş veri yapıları ve kullanıcı dostu arayüzler, geliştirilen sistemin esnekliğini ve verimliliğini artırmaktadır. MLL bileşeni sayesinde, makine öğrenimi modellerinin oluşturulması ve değerlendirilmesi kolaylaşmakta, HighGUI bileşeni ile grafik arayüzlerin etkin bir şekilde yönetilmesi mümkün hale gelmektedir [51]. Bu teknolojik üstünlükler ve avantajlar, OpenCV'yi, akıllı klima sistemi gibi kompleks ve dinamik uygulamalar için ideal bir seçim kılmaktadır. Bu tercih, sistemin çeşitli algılama ve tanıma işlemlerini gerçekleştirebilme yeteneğini güçlendirmektedir. Ayrıca, OpenCV'nin açık kaynaklı yapısı, sistem üzerinde yapılan geliştirmeleri ve adapte edilmeleri kolaylaştırarak, akıllı klima sisteminin gelecekteki ihtiyaçlara daha rahat bir şekilde cevap verebilmesini sağlamaktadır.

3.1.8. Haar Cascade Yöntemi

Yüz algılama, bilgisayarla görme alanında önemli bir araştırma konusudur. Özellikle biyometrik teknolojinin hızla gelişmesiyle birlikte, yüz algılama teknolojisi çeşitli alanlarda hem ticari hem de akademik olarak büyük bir ilgi görmektedir [52]. Bu alanda, Viola ve diğerleri tarafından önerilen Haar Cascade algoritması, özellikle dikkat çekicidir. Bu algoritmanın üstünlüklerini tam anlamıyla kavrayabilmek için, literatürde yapılan çeşitli karşılaştırmalı çalışmaların derinlemesine incelenmesi gerekmektedir.

Dalal ve Triggs tarafından geliştirilen HOG algoritması ile karşılaştırıldığında, Haar Cascade algoritması, daha düşük hesaplama maliyeti sunarak gerçek zamanlı uygulamalarda belirgin avantaj sağlamaktadır [53]. CNN tabanlı yüz tespit

algoritmaları ise genellikle daha yüksek doğruluk oranları sunmasına rağmen, bu algoritmaların eğitim ve işlem sürelerinin uzunluğu, Haar Cascade algoritmasına kıyasla belirgin bir dezavantaj oluşturmaktadır. Ayrıca, CNN tabanlı algoritmalar, genellikle büyük miktarda etiketli veri gerektirmekte ve eğitim sürecinde karmaşık donanımlara ihtiyaç duymaktadır, bu da geliştirme sürecini zorlaştırmaktadır. Buna karşılık, Haar Cascade algoritması, sınırlı miktarda eğitim verisiyle dahi etkili sonuçlar verebilmekte ve daha basit donanımlarda hızlı ve verimli çalışabilmesiyle öne çıkmaktadır. Ek olarak, Haar Cascade algoritmasının güçlü özellik çıkarım yetenekleri ve entegre kademeli sınıflandırıcı yapısı, yüksek doğruluk oranlarına ulaşılmasını mümkün kılmaktadır [54]. Bu avantajlar, Haar Cascade algoritmasını, sınırlı kaynaklarla çalışan sistemler için ideal bir seçim haline getirmektedir.

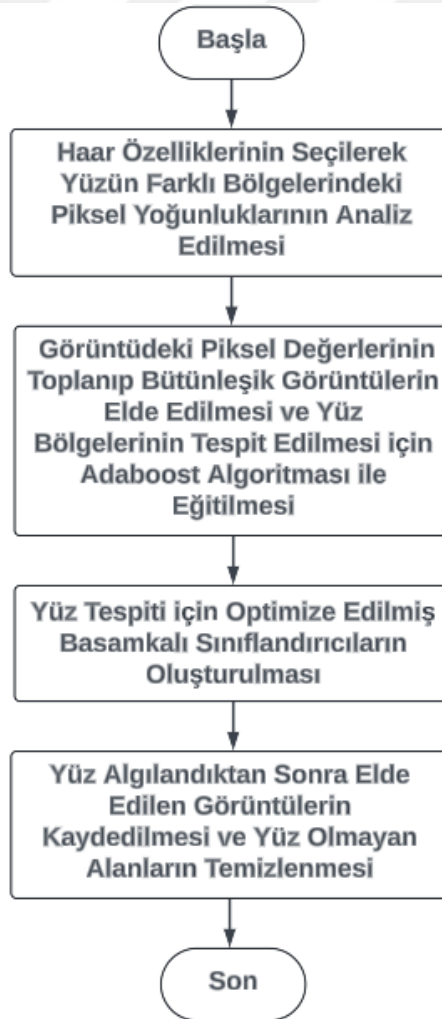
Viola-Jones algoritması, yüz tespiti için yaygın olarak kullanılan güçlü bir yöntemdir ve dört temel adımdan oluşur. İlk adım, Haar özelliklerinin seçilmesidir. Haar özellikleri, yüzün farklı bölgelerindeki piksel yoğunluğunu analiz eden basit dikdörtgen desenlerdir. Bu özellikler, yüzün karakteristik yapısını tanımlamaya yardımcı olur. İntegral görüntü tekniği, bu özelliklerin hesaplanmasını hızlandırmak için kullanılır ve herhangi bir dikdörtgen bölgenin piksel toplamını hızlı bir şekilde hesaplamayı mümkün kılarak algoritmanın performansını önemli ölçüde artırır [55].

İkinci adım, AdaBoost algoritması kullanılarak Haar özelliklerinin eğitilmesidir. AdaBoost, birçok zayıf sınıflandırıcıyı birleştirerek güçlü bir sınıflandırıcı oluşturur. Bu süreçte, her bir Haar özelliği bir zayıf sınıflandırıcı olarak kabul edilir ve AdaBoost, hangi özelliklerin yüz tespiti için en yararlı olduğunu belirler. Her özellik için bir ağırlık değeri hesaplanarak yüz ve yüz olmayan bölgeler doğru bir şekilde ayırt edilir, bu sayede yüz tespiti doğruluğu artırılır [55].

Üçüncü adımda, basamaklı sınıflandırıcılar oluşturulur. Basamaklı sınıflandırma, hızlı ve etkili yüz tespiti için kritik bir rol oynar. İlk aşamalarda, hızlı fakat az doğruluklu sınıflandırıcılar kullanılarak yüz olmayan bölgeler hızla elenir. Daha sonraki aşamalarda ise daha doğruluklu fakat yavaş sınıflandırıcılar kullanılarak

kalan bölgeler detaylı bir şekilde incelenir. Bu hiyerarşik yapı, işlem süresini kısaltırken doğru yüz tespiti yapılmasını sağlar [55].

Son adımda, yüz algılama işleminden sonra elde edilen yüz görüntüleri daha fazla işlem için kaydedilir ve yüz olmayan alanlar temizlenir. Bu aşamada, yüz tespiti yapılan bölgeler çıkarılarak yüz olmayan bölgelerden ayrılır. Algılanan yüzler, daha sonra tanıma ve doğrulama gibi ek işlemler için kullanılabilir. Bu dört adımın birleşimi, Viola-Jones algoritmasını özellikle gerçek zamanlı uygulamalar için ideal hale getirir ve çeşitli yüz varyasyonlarına karşı dayanıklılık sağlar. [55]. Viola-Jones algoritmasının yüz tespiti sürecindeki temel adımlarının gösterildiği akış şeması Şekil 3.9'da yer almaktadır.



Şekil 3. 9. Haar Cascade Algoritmasının Yüz Algılama Adımları

Haar algoritması, insan yüzlerini tespit etmek için önemli bir role sahiptir. Bu algoritma, eğitilmiş verilerden öğrenerek, insan yüzlerinin karakteristik özelliklerini tanımlamaktadır. Bu özellikler arasında iki göz arasındaki mesafe, alın ve burun arasındaki uzaklık gibi belirli anatomik ölçümler bulunmaktadır. Algoritmanın eğitimi için kullanılan veriler, pozitif ve negatif görüntüler olmak üzere iki kategoride bulunur. Pozitif görüntüler, insan yüzlerini içerirken, negatif görüntüler insan yüzlerini içermez. Algoritma, bu verileri kullanarak, insan yüzlerinin karakteristik özelliklerini belirler ve daha sonra bu özellikleri kamera önündeki görüntülerde tespit etmek için uygulamaktadır [56].

Haar özellikleri, belirli bir eşik değerinin üzerinde olduğunda, bir yüzün veya yüzlerin bir görüntü içinde yer aldığı sonucuna varmamızı sağlamaktadır. Bu özellikler, görüntü içindeki potansiyel yüzleri etkin bir şekilde filtrelemek için kullanılmaktadır. Bu filtreleme işlemi için integral görüntü tekniği kullanılmaktadır. İntegral görüntü, her pikselin değerini hızlı bir şekilde hesaplamak için kullanılan bir tekniktir. Haar özellikleri, bu integral görüntüye dayanarak hesaplanmaktadır. İntegral görüntü değeri, piksellerin toplu birikim değerini temsil eder ve soldan sağa doğru hesaplanır. Haar özellik değeri, genellikle belirli piksel grupları arasındaki fark olarak hesaplanır ve ardından bu farklar bir eşik değeriyle karşılaştırılarak yüz algılama işlemi gerçekleştirilmektedir. Bu eşik değeri, nesnenin yüz olup olmadığını belirlemek için kullanılan önemli bir parametredir [57].

3.1.9. YOLO Derin Öğrenme Algoritması

Günümüzde, derin öğrenme yöntemlerinin popüleritesi hızla artmaktadır. Yapay Zeka, bilgisayarların veya bilgisayar kontrolündeki makinelerin, insan benzeri düşünme yeteneğine sahip olmasını ifade etmektedir. Derin Öğrenme, yapay sinir ağları ve diğer makine öğrenme algoritmalarını içeren bir alanı kapsar ve genellikle bir veya daha fazla gizli katman içerir. Bu alanda, bir dizi farklı algoritma kullanılmaktadır. Bu algoritmalar içerisinde YOLO algoritması, yüksek işleme hızına sahip olmasıyla öne çıkar ve bu sayede daha net sonuçlar elde edilmesine olanak tanımaktadır [58].

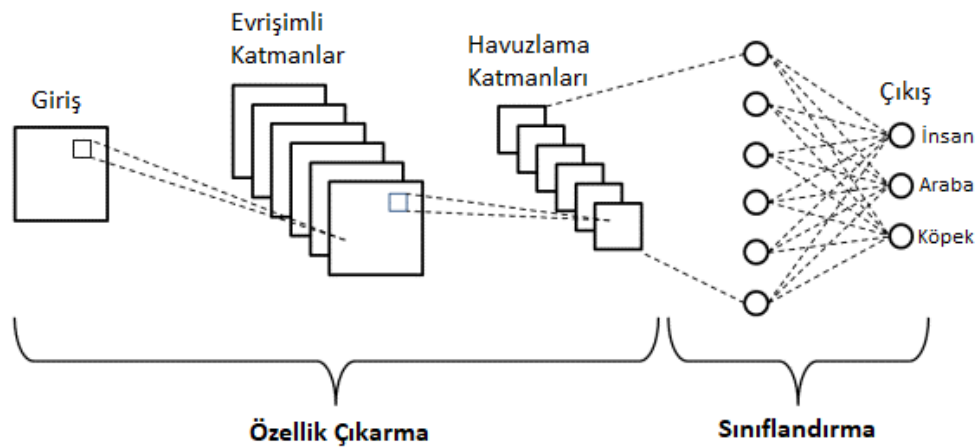
3.1.9.1. CNN Tabanlı Görsel Tanıma

Nesne tanıma, bilgisayarla görmenin temel bir alanıdır ve görüntülerde bulunan nesnelerin türlerini belirleyerek, konumlarını ve yönelimlerini tanımlamayı amaçlamaktadır. Bu nesneler arasında insan vücudu, yüzler, araçlar, hayvanlar vb. yer alabilir. Son yıllarda, özellikle insan yüzü ve vücut algılama, bu alanda büyük bir ilgi odağı olmuştur. Nesne tanıma süreci, her nesne türünün kendine özgü özelliklere sahip olduğu gerçeğine dayanmaktadır. Örneğin, daireler genellikle dairesel bir şekle sahiptir ve insanlar daireleri ararken merkezlerinden belirli bir uzaklıktaki nesneleri incelerler. Benzer şekilde, karelerin dört köşesi dik ve kenar uzunlukları eşittir. İnsanlar kareleri ararken bu özellikleri göz önünde bulundurmaktadırlar. İnsan vücudu tanıma sürecinde ise başlar, kollar, bacaklar, gövde, ten rengi ve kol mesafesi gibi özel özellikler büyük önem taşımaktadır. Geleneksel makine öğrenimi yöntemleri, önce bu özellikleri tanımlar ve daha sonra SVM gibi tekniklerle nesneleri sınıflandırmaktadır. Ancak, son yıllarda derin öğrenme, bu geleneksel yaklaşımları aşmıştır. Derin öğrenme, özel özellikleri önceden belirlemeden nesneleri doğrudan tespit edebilir. Sinir ağları kavramı, insan beyninden ilham alarak ortaya çıkmıştır ve uzun yıllar boyunca çeşitli sinir ağı mimarileri geliştirilmiştir. 1998'de araştırmacılar, CNN'i tanıtarak elle yazılan rakamları sınıflandırmak için yeni bir yaklaşım sunmuşlardır. CNN'ler, derin sinir ağlarının bir türü olarak, geleneksel makine öğrenimi yöntemlerine kıyasla nesne sınıflandırmasının doğruluğunu önemli ölçüde artırmaktadır [59].

Son yıllarda, derin öğrenme tekniklerinden biri olan CNN alanındaki gelişmeler, görüntü işleme alanında büyük ilerlemelere yol açmıştır. Bu sinir ağları, insan görsel korteksinde gözlemlenen işleme hiyerarşisinden ilham almaktadır ve eğitim verilerinden önemli özellikleri otomatik olarak çıkarabilmektedir. CNN'in bu özelliği, özellik çıkarma ve sınıflandırma süreçlerinde devrim yaratan bir nitelik taşımaktadır [60].

Şekil 3.10'da gösterildiği gibi CNN, bir girdi katmanı, bir çıktı katmanı ve birden fazla gizli katmandan oluşan karmaşık bir sinir ağı yapısına sahiptir. Bu

katmanlardaki birçok parametre eğitimle belirlenmektedir. CNN mimarisi, evrişimli katmanlar, havuzlama katmanları ve tam bağlantı katmanları olmak üzere üç ana katman türünden oluşmaktadır. Evrişimli katmanlar, matematiksel hesaplamaların çoğunu gerçekleştirdiği için CNN'nin temel yapı taşıdır. Bu katmanlar, önceki katmandan gelen veriler üzerinde belirli bir matematiksel işlem uygulayarak görüntüdeki özellikleri çıkarmakla görevlidir. Bu özellikler, giriş görüntülerinin belirli desenlerini veya özelliklerini temsil etmektedir. Ardından, havuzlama katmanları, özellik haritasının boyutunu azaltmak ve önemli bilgileri özetlemek için kullanılır. Bu sayede, ağıın parametre sayısı azaltılarak hesaplama gücü gereksinimi azaltılır. Böylece ağıın daha hızlı çalışması sağlanmaktadır. Tam bağlantı katmanları ise, nesne tanıma kararlarını oluşturmak için sınıflandırma puanlarını hesaplar. Örneğin, insan algılama durumunda, insan vücudu puanları, diğer nesnelerin puanlarından daha yüksek veya belirli bir eşik değerini aşıyorsa, CNN tabanlı görsel tanıma algoritması giriş görüntülerindeki insan vücutlarını tespit eder ve ilgili çıktıyı üretmektedir. CNN, akıllıca birleştirilmiş doğrusal ve doğrusal olmayan katmanlarının uyumlu bir kombinasyonudur. CNN'in eğitimi, geleneksel sinir ağlarını eğitmekle benzerdir; geri yayılım algoritmaları kullanılarak tüm ağ katmanlarındaki parametre değerleri ayarlanmaktadır. Eğitim tamamlandığında, bu parametre değerleri nesne sınıflandırması için nihai hale getirilmektedir [32].



Şekil 3. 10. CNN Genel Mimarisi [61]

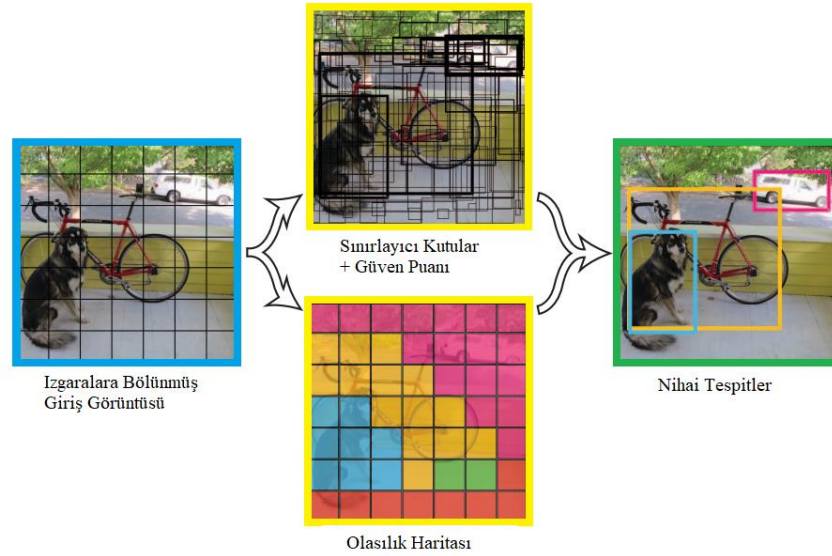
3.1.9.2. YOLO Mimarisi

Nesne tespiti, bir görüntüde veya videoda bulunan nesnelerin konumunu belirlemek ve her birini sınıflandırmak için kullanılan bir tekniktir. İlk olarak, tespit edilen nesne bir sınırlayıcı kutunun içine alınır. Ardından, bu kutu içindeki nesnenin hangi sınıfa ait olduğu tahmin edilmeye çalışılır. Son yıllarda, nesne tespiti için YOLO modelleri oldukça etkili sonuçlar vermektedir. YOLO, derin öğrenme tabanlı CNN kullanan tek aşamalı bir nesne tespit yöntemidir. YOLO, belirli bir nesnenin sınırlayıcı kutularını ve ilgili sınıf olasılıklarını uzamsal olarak ayıran bir regresyon problemine dayanmaktadır. Diğer nesne tespit modellerine kıyasla, YOLO nesneleri hızlı bir şekilde gerçek zamanlı olarak sınıflandırabilir ve aynı zamanda daha yüksek bir hassasiyet sunabilmektedir. Bu tek evrimsel sinir ağı, aynı anda birden fazla nesne üzerinde tahminler yapabilir ve her bir nesne için ayrı ayrı sınıf olasılıkları üretebilmektedir [10].

YOLO algoritması, diğer yöntemlerle karşılaştırıldığında nesneleri daha hızlı tahmin edebilmektedir. Model boyutunu kolayca değiştirerek hız ve doğruluk arasında denge sağlayabilir ve diğer sistemlere göre yaklaşık dört kat daha hızlı çalışmaktadır. Ayrıca, yeniden eğitim gerektirmeyen esnek yapısıyla öne çıkar. YOLO, diğer CNN tabanlı sistemlerin aksine, nesneleri algılarken sınıflandırıcıları yeniden kullanmak yerine tek bir CNN kullanır. Bu yaklaşım, özellikle nesnelerin sınıflandırılması ve konumlandırılması için daha etkili sonuçlar sunmaktadır [58].

YOLO algoritması, görüntü işleme hızında oldukça etkileyici bir performans sergiler, yaklaşık 40-90 FPS hızında işlem yapabilmektedir. Bu, diğer yöntemlerle karşılaştırıldığında önemli bir hız avantajı sağlar ve gerçek zamanlı video işleme uygulamaları için ideal bir çözüm oluşturmaktadır. Örneğin, video akışını birkaç milisaniye gecikmeyle işleyebilir, bu da hızlı ve etkin bir algılama süreci sunar. Ayrıca, R-CNN ve Faster R-CNN gibi diğer nesne algılama yöntemlerine kıyasla, YOLO algoritması büyük bir hız artışı sağlar; R-CNN'e kıyasla 1000 kat, Faster R-CNN'e kıyasla ise 100 kat daha hızlıdır. Bu özellikleriyle, YOLO algoritması gerçek zamanlı nesne algılama uygulamalarında üstün performans sağlamaktadır [62].

YOLO sinir ağı mimarisi, giriş görüntüsünü birçok ızgara hücresine böler ve her bir hücrenin yalnızca bir nesneyi algılayabileceği ve belirli sayıda sınırlayıcı kutuya sahip olduğu bir yaklaşımı benimsemektedir. Her sınırlayıcı kutu, algılanan nesneye ait bir güven puanı ile ilişkilendirilir. Bu da algılamanın ne kadar güvenilir olduğunu göstermektedir. YOLO, aynı nesnenin birden fazla kez tespit edilmesini önlemek amacıyla en düşük güven puanına sahip tahminleri elemek için maksimum olmayan bastırma yöntemini kullanmaktadır. Bu yaklaşım, nesnelerin belirlenmesinde ve sınıflandırılmasında kullanılan sınırlayıcı kutuların koordinatlarını tahmin etmek için evrişimli sinir ağlarını temel almaktadır. YOLO, evrişimli katmanları ön eğitime tabi tutar, bu da görüntülerin temel özelliklerini öğrenmesini sağlamaktadır. Ardından, insan algılama eğitimi gerçekleştirilmektedir. Bu da nesnelerin belirlenmesi ve sınıflandırılmasını içerir. Bu sayede, YOLO nesne algılama konusunda son derece etkili bir yöntem haline gelmektedir [32]. YOLO algoritmasının nesne tespiti sırasında çalışma prosesleri Şekil 3.11’da gösterilmektedir.



Şekil 3. 11. YOLO Algoritması Nesne Algılama Süreci [10]

3.1.10. Tensorflow Makine Öğrenimi Kütüphanesi

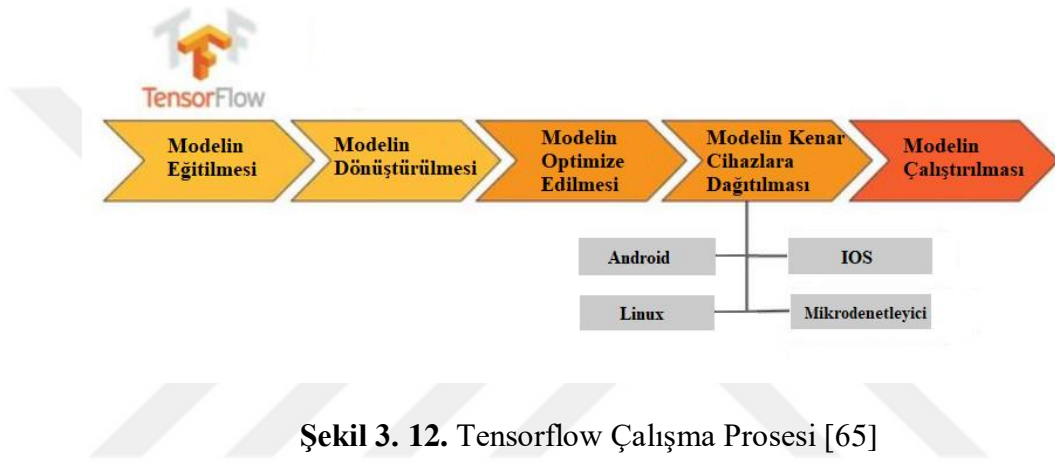
TensorFlow, Google tarafından geliştirilen açık kaynaklı bir makine öğrenimi ve derin öğrenme kütüphanesidir. Temelde, büyük veri setleri üzerinde karmaşık

matematiksel işlemleri gerçekleştirmek için kullanılan bir araçtır. TensorFlow'un temelini, hesaplama grafikleri oluşturmaktadır. Bu grafikler, verilerin akışını nasıl kontrol edeceğimizi ve hangi işlemlerin gerçekleştirileceğini belirlemektedir. TensorFlow, bu grafikler üzerinde çalışan düğümlerden oluşmaktadır. Her düğüm; girdi alır, bir işlem gerçekleştirir ve çıktı üretir. Bu, kullanıcıların makine öğrenimi modelleri oluştururken grafikleri görselleştirmesini ve bu grafiklerdeki her bir düğümün ne yaptığını anlamasını sağlamaktadır. TensorFlow, birçok önceden oluşturulmuş katman ve işlevler içeren zengin bir kütüphane sunmaktadır. Bu, kullanıcıların kendi modellerini baştan oluşturmak yerine, bu hazır yapıları kullanarak daha hızlı ve etkili bir şekilde çalışmalarına olanak tanımaktadır. TensorFlow, ayrıca model eğitimi, değerlendirmesi ve dağıtımı için kullanıcı dostu araçlar sağlamaktadır. Bu sayede, kullanıcılar karmaşık makine öğrenimi projelerini daha kolay yönetebilir ve gerçek dünyadaki problemlere çözüm bulabilmektedirler [63].

Google'ın TensorFlow'u, makine öğrenimi uygulamaları için kullanılan bir çerçeve olup, özellikle veri akışı programlamasıyla işlev görmektedir. Bu çerçeve, matematiksel işlemleri temsil eden düğümlerden ve çok boyutlu veri yapılarını temsil eden Tensörlerden oluşur. TensorFlow'un sunduğu nesne algılama API'si, görüntülerdeki nesneleri tanımlamak ve canlı video akışındaki nesneleri hızlı ve doğru bir şekilde tespit etmek için kullanılmaktadır. Bu API, genellikle saniyede 20 ila 30 kare hızında çalışabilir, böylece hızlı ve gerçek zamanlı nesne algılama sağlamaktadır. Ayrıca, TensorFlow'un esnekliği ve verimliliği sayesinde, sürücüsüz araçlar gibi canlı video akışıyla ilgili karmaşık uygulamalarda kullanılabilir. Ek olarak, TensorFlow'un nesne algılama API'si, MobileNet gibi hafif sinir ağlarıyla birlikte kullanıldığında, düşük hesaplama maliyetine sahip ve verimli bir şekilde nesne algılama yapabilen bir veri kümesi oluşturmaktadır. Bu özellikleriyle TensorFlow, çeşitli uygulama alanlarında kullanılan güçlü bir nesne algılama çözümü sunmaktadır [64].

TensorFlow'un çalışma akışı, belirli adımların dikkatlice takip edilmesini gerektiren bir süreçtir. İlk adım, modelin eğitilmesidir. Bu adımda, genellikle büyük bir veri seti kullanılarak modelin parametreleri ayarlanır ve istenilen sonuçları elde etmek için optimize edilmektedir. Ardından eğitilmiş model, TensorFlow Lite gibi

daha hafif bir formata dönüştürülür ve optimize edilir. Dönüştürülen model, kenar cihazlarda kullanılmak üzere dağıtılır. Bu adımda, model Android, Linux, iOS veya mikrodenetleyiciler gibi farklı platformlara uygun hale getirilmektedir. Son olarak, modelin kenar cihazlarda kullanılmasıyla gerçek zamanlı çıkarımlar yapılmaktadır. Bu adımda, model yerel olarak çalıştırılarak, kullanıcının verileri üzerinde tahminler yapılmakta ve sonuçlar elde edilmektedir. Bu adımların dikkatlice takip edilmesi, TensorFlow'un etkin ve güçlü bir şekilde kullanılmasını sağlamaktadır. Şekil 3.12'de Tensorflow'un çalışma prosesi yer almaktadır [65].



Şekil 3. 12. Tensorflow Çalışma Prosesi [65]

3.1.11. Yazılım Geliştirme Programı

Proje kapsamında, geliştirilen algoritmalar için Thonny IDE tercih edilmiştir. Thonny, kullanıcı dostu bir yazılım geliştirme ortamı olarak öne çıkar. Basit arayüzü ve kullanımıyla dikkat çeken Thonny, programlama dilleri arasında Python'u destekler. Temel kod düzenleme, derleme ve hata ayıklama işlevlerinin yanı sıra, eklentilerle genişletilebilir ve kişiselleştirilebilir. Kullanıcılar, kodlarını verimli bir şekilde yazabilir ve yönetebilirken, Thonny'nin hızlı performansı ve düşük kaynak tüketimi de tercih sebepleri arasındadır. Thonny'nin sunduğu kullanım kolaylığı ve esneklik, çeşitli yazılım projeleri için ideal bir çözüm sunmaktadır. [66].

Geliştirilen projede, Python dilinde yazılmış ve açık kaynak kodlu olan OpenCV, Haar Cascade, YOLO ve Tensorflow gibi algoritmaların sisteme kolayca

entegre edilebilmesi için yazılım mimarisi Python dilinde oluşturulmuştur. Thonny IDE programının hem Python dilini desteklemesi hem de sunduğu kullanıcı dostu arayüz ve hızlı işlem özellikleriyle bu projenin kodlarını derlemek için ideal bir araç olmaktadır.

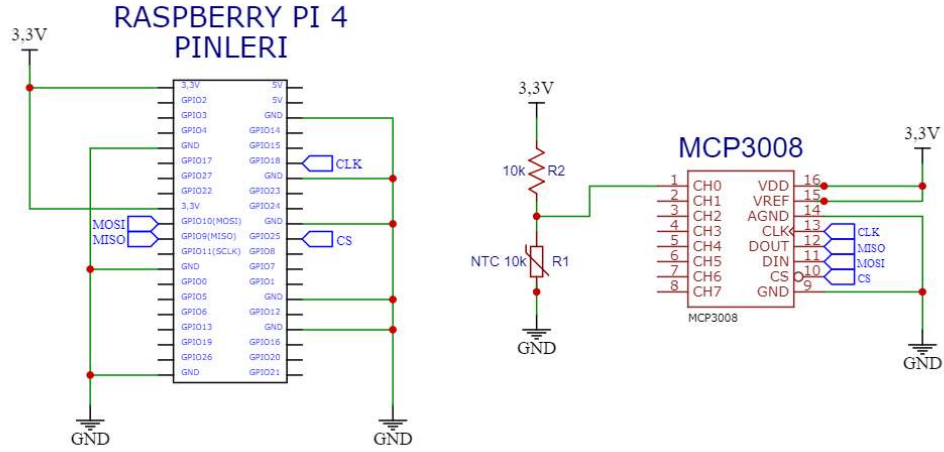
3.2. Yöntemler

Bu bölümde geliştirilen akıllı klima sisteminin temel bileşenleri ve algoritma mimarisi detaylı bir şekilde incelenmektedir. Özellikle, yüz tanıma ve insan algılama sistemleri, geliştirilen projenin merkezinde yer almakta olup, bu sistemlerin entegrasyon, tasarım ve uygulanma süreçleri ayrıntılı olarak açıklanmaktadır.

3.2.1. İç Ortam Sıcaklığının Ölçülmesi

Bu bölümde, iç ortam sıcaklığının ölçümü için kullanılan NTC termistörü ve MCP3008 ADC entegresi aracılığıyla Raspberry Pi ile entegre edilme yöntemi ve kullanımı detaylı bir şekilde ele alınmaktadır.

Raspberry Pi kartındaki GPIO pinleri yalnızca dijital veri okuma ve yazma yeteneğine sahiptir. Bundan dolayı analog veri okumak için MCP3008 ADC entegresi kullanılmaktadır. Sıcaklık ölçümü için geliştirilen yazılımın doğrulanabilmesi için devre bir breadboard üzerinde kurulmuş ve test edilmiştir. Kurulan devrenin şematik gösterimi Şekil 3.13'te yer almaktadır.



Şekil 3. 13. MCP3008 ve NTC Bağlantı Şeması

Şekil 3.13'te görüldüğü gibi, MCP3008 entegresinin SPI arayüz bağlantıları Raspberry Pi ile gerçekleştirilmiştir. NTC termistör, sıcaklık değişimine göre direnç değeri değişen bir devre elemanıdır. Sıcaklık verisinin elde edilebilmesi için öncelikle termistörün direnç değişimi ölçülmelidir. Bu amaçla, direnç değerinin ölçülerek voltaj değerine dönüştürülmesi ve bu voltaj değişiminin MCP3008 ADC entegresi üzerinden okunması gerekmektedir. Bu işlemi gerçekleştirebilmek için bir voltaj bölücü devresi tasarlanmış ve uygulanmıştır. Voltaj bölücü devresinden voltaj değişim verisinin elde edileceği kısım MCP3008 entegresinin ilk kanalı olan CH0 pinine bağlanmıştır. Voltaj bölücü devresinde NTC termistöre seri olacak şekilde 10kΩ değerinde direnç bağlanmıştır. NTC termistör direnci 'R' olarak tanımlanıp, CH0 pinine bağlı olan bölgedeki voltajı ölçebilmek için 3.1 numaralı denklem kullanılmaktadır.

$$V_o = \frac{R}{(R+10000)} \times V_{cc} \quad (3.1)$$

MCP3008 entegresinin CH0 kanalından okunan analog değeri bulabilmek için 3.2 numaralı denklem kullanılmaktadır.

$$ADC_{Değeri} = V_o \times \frac{1023}{V_{aref}} \quad (3.2)$$

$$ADC_{Değeri} = \frac{R}{R+10000} \times V_{cc} \times \frac{1023}{V_{aref}} \quad (3.3)$$

Şekil 3.11’de gösterildiği gibi lojik ve analog referans voltajlarının her ikisi de 3,3V’dur. Dolayısıyla 3.3 numaralı denklemde V_{cc} ve V_{aref} değişkenleri birbirlerine eşittir.

$$ADC_{Değeri} = \frac{R}{R+10000} \times 1023 \quad (3.4)$$

$$R = \frac{10000}{\frac{1023}{ADC_{Değeri}} - 1} \quad (3.5)$$

3.4 numaralı denklemde hareketle, hedeflenen termistör direnç değerini belirleyen 3.5 numaralı denklem türetilmektedir. Bu sayede, MCP3008 ADC entegresi ile NTC termistörün analog değerinden doğrudan direnç değeri ölçülebilmektedir.

NTC termistörler ile sıcaklık ölçümü için genellikle Steinhart-Hart denklemi veya β (B) parametresi denklemi gibi matematiksel modeller kullanılmaktadır [67].

$$\frac{1}{T} = a + b \times \ln R + c \times (\ln R)^3 \quad (3.6)$$

3.6 numaralı denklemde yer alan Steinhart-Hart denklemi, termistör direnç değerini sıcaklığa dönüştürmek için kullanılmaktadır. Ancak bu denklemde çok sayıda bilinmeyen değişken bulunur ve bundan dolayı hesaplama karmaşıklığı artmaktadır. Bu nedenle sıcaklık ölçümü için genellikle Steinhart-Hart denkleminden türetilen β (B) parametresi denklemi tercih edilmektedir [67].

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \times \ln \frac{R}{R_0} \quad (3.7)$$

3.7 numaralı denklemde yer alan β (B) parametresi denklemi, sıcaklık ile direnç değerleri arasındaki ilişkiyi basitleştirerek, sıcaklık değerini ölçmek için daha pratik bir yöntem sunmaktadır. β (B) parametresi denkleminde yer alan T bilinmeyi, Kelvin cinsinden ölçülmek istenilen sıcaklığı temsil etmektedir. T_0 bilinmeyi oda sıcaklığını temsil etmektedir. Genellikle bu değer 25 °C yani 298,15 °K olarak kabul edilmektedir. B bilinmeyi NTC termistörün sıcaklık katsayısı değeridir [67].

Tablo 3.3'te gösterildiği gibi geliştirilen projede kullanılan termistörün sıcaklık katsayısı değeri 3950'dir. R_0 bilinmeyi, termistörün oda sıcaklığındaki direncini ifade etmektedir [67]. Tablo 3.3'te gösterildiği gibi geliştirilen projede kullanılan termistörün oda sıcaklığındaki direnci 10 k Ω olarak üretici firma tarafından belirlenmiştir. β (B) parametresi denkleminde yer alan R bilinmeyi, termistörün farklı sıcaklıklardaki direnç değerini temsil etmektedir [67]. Termistörün farklı sıcaklıklardaki direnç değeri, 3.5 numaralı denklem kullanılarak hesaplanmaktadır.

Geliştirilen proje kapsamında iç ortam sıcaklığının NTC termistör ile ölçülebilmesi için 3.5 ve 3.7 numaralı matematiksel denklemler Python dilinde yazılan algoritmaya entegre edilmiştir. Şekil 3.14'te iç ortam sıcaklığının ölçülmesi için geliştirilen Python dilindeki algoritmanın kod örneği yer almaktadır.

```
def Temperature():
    thermistorAnalogValue.clear() # Clear previous measurement
    for i in range(sampleNumber):
        thermistorAnalogValue.append(channel.value)

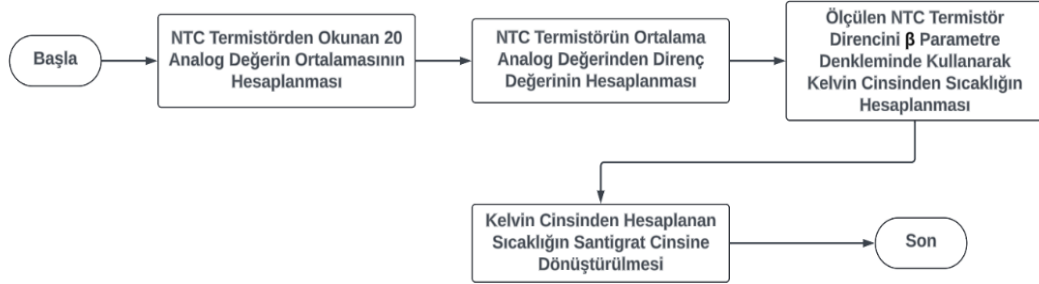
    averageThermistor = sum(thermistorAnalogValue) / sampleNumber
    thermistorResistance = R1 / ((1023.0 / averageThermistor) - 1.0)

    tempKelvin = (1.0 / tempRef) + (math.log(thermistorResistance / R0) / beta)
    tempKelvin = 1 / tempKelvin
    tempCelsius = tempKelvin - 273.15
    print("Sıcaklık: {:.2f}°C".format(tempCelsius))
```

Şekil 3. 14. İç Ortam Sıcaklığı Ölçüm Algoritması

Geliştirilen sıcaklık ölçümü algoritmasının çalışma sürecini anlatan akış şeması, Şekil 3.15'te yer almaktadır. Bu akış şeması, algoritmanın iç ortam sıcaklığını

ölçme işlemini adım adım görselleştirerek, algoritmanın mantığını ve işleyişini daha açık bir şekilde sunmayı amaçlamaktadır.



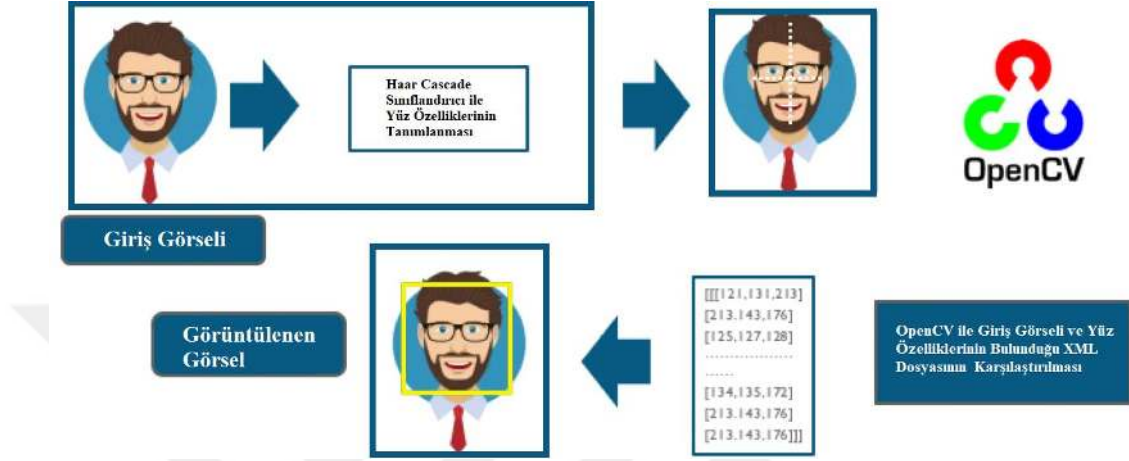
Şekil 3. 15. İç Ortam Sıcaklığı Ölçüm Algoritmasının Akış Şeması

Sıcaklık ölçümünde daha hassas sonuçlar elde etmek amacıyla, termistör direnç değeri hesaplanırken tek bir analog ölçüm yerine 20 kez analog ölçüm alınmıştır. Ardından, bu alınan 20 analog ölçümün ortalaması alınarak, sıcaklığın daha hassas bir şekilde ölçülmesi hedeflenmektedir. Bu yöntem, ölçümün güvenilirliğini artırarak sistem performansını optimize etmeyi amaçlamaktadır. Geliştirilen sistemde tasarlanan bu ölçüm yöntemi, iç ortam sıcaklığının hassas bir şekilde belirlenmesini sağlamaktadır. Python dilinde yazılan algoritmanın kullanımı, ölçüm sürecinin verimli ve doğru bir şekilde gerçekleştirilmesini sağlamaktadır. Bu çalışma, iç ortam sıcaklığının güvenilir bir şekilde izlenmesi için etkili bir yöntem sunmaktadır.

3.2.2. Gerçek Zamanlı Yüz Tanıma Sisteminin Geliştirilmesi

Yüz tanıma sistemi, yüz özelliklerine dayanarak bireyleri gerçek zamanlı olarak tanımlamamıza olanak sağlayan bir araçtır. Bu teknoloji oldukça önemlidir ve geniş bir uygulama yelpazesinde kullanılmaktadır. Yüz tanıma sistemleri, erişim kontrolü, görüntü eşleştirme, kimlik doğrulama ve benzeri işlevler için açık veya sınırlı alanlarda insanları tanımlamak için kullanılabilir [52]. Geliştirdiğim sistem, Raspberry Pi platformu üzerinde OpenCV ve Haar Cascade makina öğrenimi algoritması kullanılarak gerçek zamanlı yüz tanıma yapabilen bir uygulama sunmaktadır. Sistem, ilk olarak kendi fotoğraflarımdan oluşan veri seti ile eğitilmiş ve

daha sonra gerçek zamanlı tanıma için optimize edilmiştir. Bu sistem, yüz tanıma teknolojisinin günlük hayatımızda daha yaygın ve kullanıcı dostu bir şekilde kullanılmasına katkıda bulunmaktadır. Yüzün, Haar Cascade ve OpenCV kullanılarak algılanma sürecinin görseli Şekil 3.16'da yer almaktadır.



Şekil 3. 16. Haar Cascade Sınıflandırıcı ile Yüz Algılanması [68]

3.2.2.1. Veri Setlerinin Oluşturulması ve Sistemin Eğitilmesi

Sistemin eğitim aşamasında, kendi fotoğraflarıyla yüz tanıma modelimi geliştirmek için Haar Cascade algoritması kullanılmıştır. Öncelikle Raspberry Pi üzerindeki kamera modülü aracılığıyla fotoğraf çekilebilmesi için bir algoritma geliştirilmiştir. Bu algoritma ile farklı ışık koşullarında ve açılarında 50 farklı fotoğraf çekilmiş ve bu fotoğraflar eğitim veri seti olarak kullanılmıştır. Daha sonra, Haar Cascade algoritmasıyla yüz tanıma modeli, bu fotoğraflardan elde edilen özellikler ve parametreler kullanılarak eğitilmektedir. Farklı ışık koşullarında çekilen fotoğrafların kullanılması, modelin genel tanıma doğruluğunu artırarak optimizasyonunu sağlamaktadır. Şekil 3.17'de veri setlerinin oluşturulması için hazırlanan kod örneği yer almaktadır.

```

import os
import cv2
from picamera import PiCamera
from picamera.array import PiRGBArray

name = 'Atahan'
priorityNumber = 1
requestTemp = 19

# Klasör adını oluştur
folder_name = "{}- {} {}".format(priorityNumber, name, requestTemp)
folder_path = os.path.join("dataset", folder_name)
os.makedirs(folder_path, exist_ok=True) # Klasörü oluştur veya varsa geç

cam = PiCamera()
cam.resolution = (512, 304)
cam.framerate = 10
rawCapture = PiRGBArray(cam, size=(512, 304))

img_counter = 0

while True:
    for frame in cam.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array
        cv2.imshow("Press Space to take a photo", image)
        rawCapture.truncate(0)

        k = cv2.waitKey(1)
        rawCapture.truncate(0)
        if k%256 == 27: # ESC pressed
            break
        elif k%256 == 32:
            # SPACE pressed
            img_name = os.path.join(folder_path, "image_{}.jpg".format(img_counter))
            cv2.imwrite(img_name, image)
            print("{} written!".format(img_name))
            img_counter += 1

        if k%256 == 27:
            print("Escape hit, closing...")
            break

cv2.destroyAllWindows()

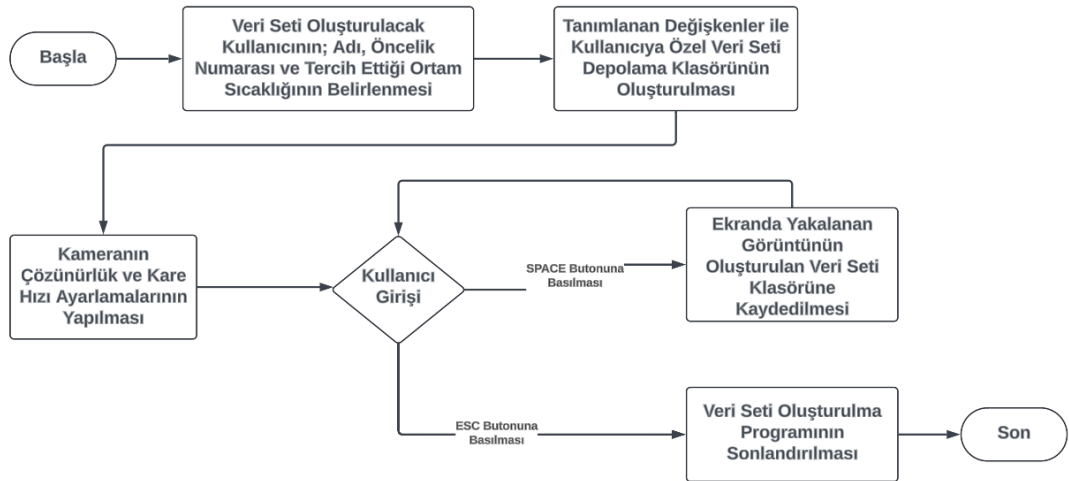
```

Şekil 3. 17. Veri Setleri Hazırlığı Yazılım Kodu

Haar Cascade algoritması, yüz tanıma modelini eğitmek için gerekli olan özellikleri ve parametreleri belirlemek için XML dosyası kullanmaktadır. Bu XML dosyası, belirlenen özelliklerin ve sınıflandırıcıların tanımlanmasını içerirken, aynı zamanda modelin nasıl çalışacağını tanımlayan yapısal bilgileri de içermektedir. Bu dosya, modelin nasıl çalışacağını ve yüz tanıma işlemi için gerekli olan özelliklerin ve parametrelerin belirlenmesini sağlamaktadır.

Şekil 3.17’de yer alan algoritma Python dilinde oluşturulmuştur. Raspberry Pi üzerinde bir PiCamera kullanarak fotoğraf çekmek ve çekilen bu fotoğrafların belirli bir klasörde saklanarak veri seti oluşturmak amaçlanmaktadır. İlk olarak gerekli

kütüphaneler algoritmaya eklenmektedir. Kullanıcının adı, öncelik numarası ve istenilen oda sıcaklık bilgileri değişken olarak tanımlanmaktadır. Daha sonra tanımlanan değişkenler kullanılarak dinamik bir klasör adı oluşturulmaktadır. Oluşturulacak dinamik klasör kullanıcıya özel olup ilgili kullanıcının görüntülerinin saklanması için kullanılmaktadır. Klasör oluşturulduktan sonra kamera konfigürasyonu yapılarak kameranın çözünürlüğü ve kare hızı ayarlanmaktadır. Daha sonra sonsuz bir döngü içinde kamera sürekli olarak görüntü yakalamakta ve her bir karede yakalana görüntü “cv2.imshow” fonksiyonu ile ekranda gösterilmektedir. Klavyeden “SPACE” butonuna basıldığında ekranda yakalanan görüntü, oluşturulan klasör içerisine kaydedilmekte ve “img_counter” değişkeni kullanılarak kaydedilen görselin dosya adı oluşturulmaktadır. “ESC” butonuna basıldığında döngü içinden çıkılarak program sonlandırılmaktadır. Şekil 3.18’de veri seti oluşturmak için geliştirilen algoritmanın akış şeması yer almaktadır.



Şekil 3. 18. Veri Seti Oluşturma Algoritması Akış Şeması

Eğitim aşaması, yüz tanıma modelinin temelini oluştururken, çeşitli ışık koşulları ve farklı açılardan çekilen fotoğrafların kullanımı, modelin genel performansını artırmada kritik bir rol oynamaktadır. Veri seti oluşturulurken çekilen fotoğraflardaki ışık açısı ile yüz tanımlama işlemi yapılırken karşılaşılan mevcut ışık açısının farklılığı, sistemin yüzü tanımlamada zorlandığını göstermiştir. Bu durum, veri setindeki görüntüler ile gerçek zamanlı yüzleri eşleştirme konusunda sorunlara

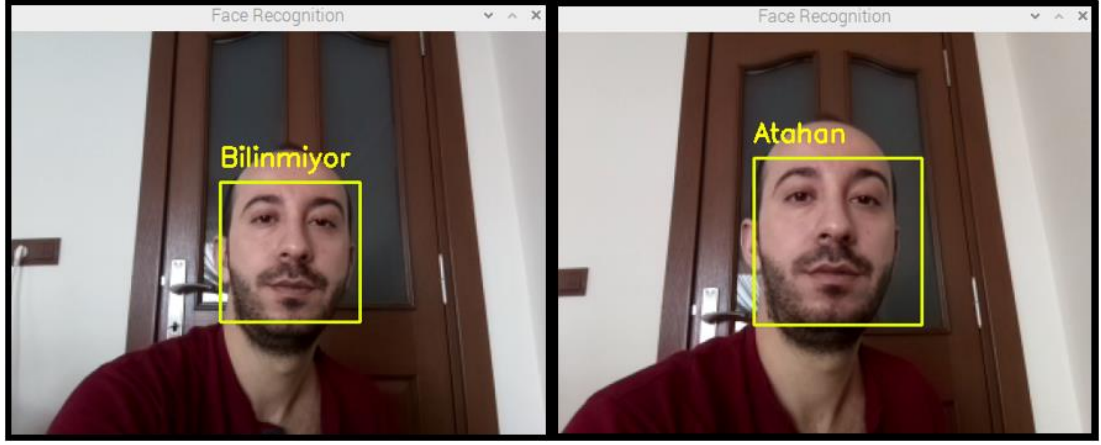
yol açmaktadır. Bu zorluğun üstesinden gelmek amacıyla, farklı ışık açılarında çekilmiş fotoğraflar kullanılarak daha kapsamlı bir veri seti oluşturulmuştur.

Haar Cascade algoritmasıyla eğitilen model, belirlenen özellikler ve parametreler doğrultusunda optimize edilmiştir. Eğitim sürecinde, veri setlerinin ön işlenmesi büyük bir önem taşımış ve bu adımda görüntülerin normalize edilmesi, yüzlerin doğru etiketlenmesi gibi işlemler gerçekleştirilmiştir. Modelin farklı ışık koşulları altında dahi yüksek doğrulukla çalışabilmesi için, çeşitli ışıklandırma senaryolarında veri toplama süreci dikkatlice planlanmış ve uygulanmıştır. Sonuç olarak, eğitim aşaması, sistemin performansını ve güvenilirliğini artırmak için kritik bir adım olup, modelin farklı ışık koşulları altında dahi yüksek doğrulukla çalışabilmesini sağlamıştır.

3.2.2.2. Yüz Tanıma Sisteminin Geliştirilmesi

Eğitim süreci tamamlandıktan sonra, sistem gerçek zamanlı yüz tanıma yapmaya hazır hale gelmiştir. Bu aşamada, Raspberry Pi üzerindeki kamera modülü aracılığıyla yakalanan görüntüler, eğitilmiş olan yüz tanıma modeline giriş verisi olarak gönderilmektedir. Kamera modülü, çeşitli ortamlarda ve ışık koşullarında yüzleri yakalayabilir, bu da sistemin çeşitli ortamlarda etkin bir şekilde çalışmasını sağlamaktadır. Yakalanan görüntüler, piksellerin yoğunluğu ve yerleşimi gibi özellikler açısından analiz edilmekte ve yüz tanıma modeli tarafından işlenmektedir.

Eğitim sürecinde kullanılan fotoğraflar, gerçek zamanlı tanıma aşamasında temel veri setini oluşturmaktadır. Bu aşamada, eğitilen modelin tanıma yetenekleri, kameradan yakalanan görüntüler üzerinde test edilir. Eğer bir yüz, eğitim veri setinde yer alan fotoğraflardan biriyle eşleşirse, sistem bu yüzü tanır ve o fotoğrafın kaydedildiği dosya adını etiket olarak kullanarak yüzü tanımlamaktadır. Ancak, eğer yakalanan yüz eğitim veri setindeki fotoğraflarla eşleşmezse, sistem bu yüzü "Bilinmiyor" olarak etiketlemektedir. Şekil 3.19'da sistem tarafından tanımlanmayan bir yüzün oluşturulan veri seti ile eğitilmesinin ardından tanımlanması yer almaktadır.



Şekil 3. 19. Eğitimden Önce ve Sonra Algılanan Yüzlerin Etiketlenmesi

Geliştirilen sistemimin gerçek zamanlı yüz tanıma işlevi, eğitim sürecinin başarıyla tamamlanmasıyla mümkün hale gelmektedir. Raspberry Pi üzerine entegre edilen kamera modülü, çeşitli ortamlarda ve farklı ışık koşullarında yüzleri yakalayarak sistemin genel etkinliğini artırmaktadır. Eğitilen model, kullanıcıların yüzlerini tanımlama ve kişiselleştirilmiş hizmetler sunma yeteneğine sahip olmaktadır.

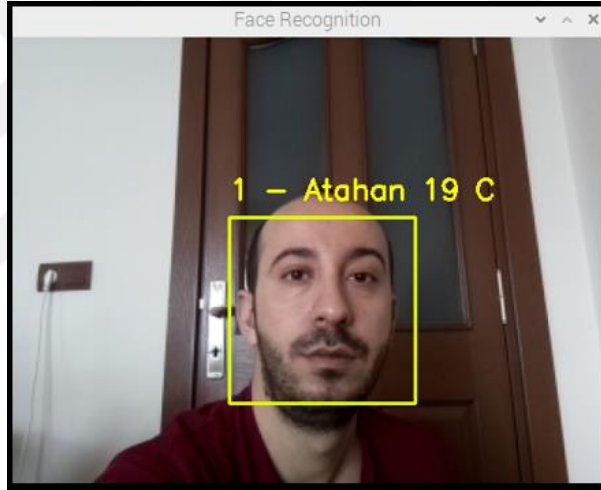
3.2.2.3. Sıcaklık Verisinin Yüz Tanıma Sistemine Eklenmesi

Sistemin eğitim aşamasında, kullanıcıların fotoğraflarıyla birlikte kişisel veri setleri oluşturulmakta ve her bir kullanıcının veri seti ayrı klasörlerde depolanmaktadır. Bu süreçte, kullanıcıların tercih ettikleri ortam sıcaklık verisi de sisteme entegre edilmektedir. Her kullanıcının belirlediği ortam sıcaklık verisi, ilgili kişinin veri seti klasör adı ile ilişkilendirilir. Eğitim aşaması tamamlandıktan sonra, sistem gerçek zamanlı yüz tanıma sürecine hazır hale gelmektedir.

Gerçek zamanlı yüz tanıma sistemi, kamera tarafından algılanan her yüzü, bu kişisel veri setleriyle karşılaştırarak tanımaya çalışmaktadır. Algılanan yüz, veri setlerinden biriyle eşleşirse, sistem yüzü tanır ve kişiyi ilgili veri setinin klasör adı ile etiketlemektedir. Bu etiketle birlikte, ilgili kişinin belirlediği ortam sıcaklık tercihi de hatırlanmaktadır.

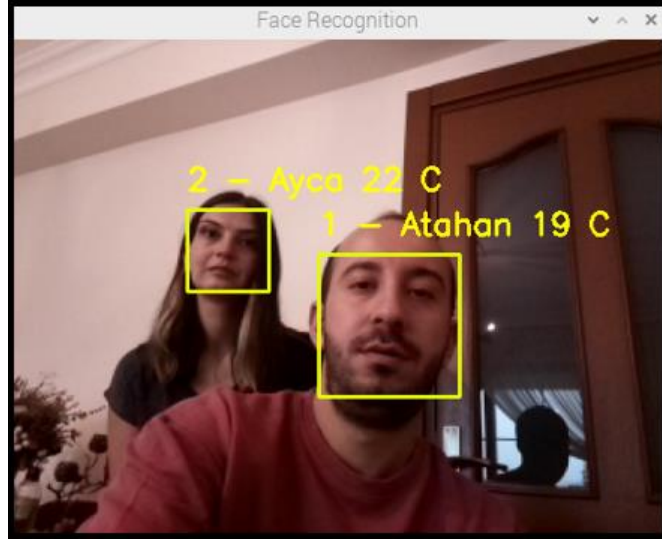
3.2.2.4. Anlamalı Çıktıların Elde Edilmesi

Şekil 3.20’de gösterildiği gibi kamera karşısında tanımlanan kişilerin etiket isimleri, belirledikleri sıcaklıkları ve etiket isimlerinin önünde öncelik numaraları yer almaktadır. Öncelik numaraları, birden fazla kullanıcının tanımlanması durumunda öncelik sıralaması gibi bir mekanizmayı devreye almak için kullanılmaktadır. Örneğin, sistemde tanımlanan iki veya daha fazla kişi varsa ve kamera karşısına birden fazla kişi aynı anda geçerse, öncelik sırası devreye girmektedir. Bu öncelik sırası, eğitim aşamasında belirlenmekte ve her bir kullanıcıya öncelik numarası atanmaktadır. Öncelik numarası, sistemdeki karmaşıklığı azaltmak ve kullanıcı tercihlerine öncelik vermek amacıyla belirlenmektedir.



Şekil 3. 20. Algılanan Yüzün Etiketlenmesi

Kullanıcıların öncelik numaralarına göre sıralandığı bu sistemde, kamera tarafından algılanan ve tanınan kişilerin öncelik numaraları, hangi kullanıcının tercihlerinin öncelikli olduğunu belirlemek için kullanılmaktadır. Bu sayede, sistem klimaya hangi kullanıcının sıcaklık tercihini ileteceğine karar verirken öncelik numaralarını dikkate almakta ve uygun şekilde ortam sıcaklığını ayarlayabilmektedir. Bu yaklaşım, sistemin çalışma optimizasyonunu artırırken karmaşıklığı önlemekte ve her kullanıcının tercihlerine uygun bir ortam sağlamaktadır. Şekil 3.21’de aynı anda iki kişi algılanması durumunda sistemin nasıl çıktı verdiği gösterilmektedir.



Şekil 3. 21. Aynı Anda Algılanan İki Yüzün Etiketlenmesi

Şekil 3.20 ve Şekil 3.21'de, sistem tarafından tanımlanan yüzlerin etiketlenme süreci gösterilmektedir. Tanımlanan her kişi için öncelik numarası, isim ve istediği ortam sıcaklık değeri etiket olarak atanmaktadır. Ancak, bu etiketler String veri tipindedir ve sistemin bu bilgileri kullanabilmesi için ayrıştırılıp Integer veri tipine dönüştürülmesi gerekmektedir. Etiket içindeki String veri tipindeki karakterler, boşluklar aracılığıyla ayrıştırılarak bir liste haline getirilmektedir. Listenin ilk elemanı öncelik numarasını, sondan ikinci elemanı ise istenilen sıcaklık değerini içermektedir. Şekil 3.22'de etiket içerisinde yer alan karakterler detaylı olarak sunulmaktadır.

1. Öncelik Numarası	2. -	3. Kullanıcı İsmi	4. Ortam Sıcaklığı	5.C
---------------------	------	-------------------	--------------------	-----

Şekil 3. 22. Etiket İçerisindeki Karakterler ile Oluşturulan Liste

Bu ayrıştırma işlemi, sistemin yüz tanımlama sürecini tamamladıktan sonra gerekli fonksiyonları yerine getirebilmesi için kritik öneme sahiptir. Ayrıştırma işleminin nasıl yapıldığına dair kod örneği Şekil 3.23'te yer almaktadır.

```
#Function for get priority number and request temp
def extract_priority_and_temperature(name):
    parts = name.split(" ")
    priority_number = int(parts[0])
    requestTemperature = int(parts[-2])
    return priority_number, requestTemperature
```

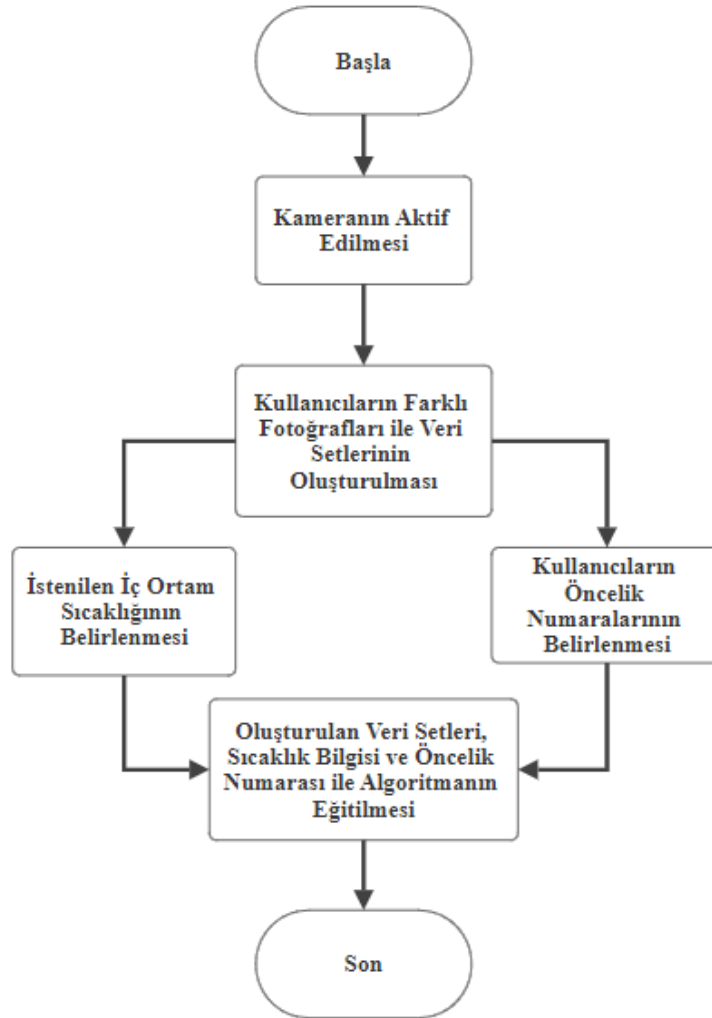
Şekil 3. 23. Tanımlanan Yüz Etiketinden Gerekli Verilerin Ayrıştırılması

Geliştirilen Yüz tanıma işlemi sırasında algılanan yüzün kimliği belirlendikten sonra sistem, öncelik numarasına göre ilgili kullanıcının tercih ettiği ortam sıcaklık bilgisini de dikkate almaktadır. Bu sıcaklık bilgisi, klima kontrolü için kullanılmakta ve klimaya ortamı belirlenen sıcaklığa ulaştırması için sinyal gönderilmektedir. Ayrıca, NTC sensörü aracılığıyla ölçülen ortam sıcaklığı bilgisi, klimanın ortamı belirlenen sıcaklığa ulaştırabilmesi için geri bildirim sağlamaktadır. Bu şekilde, sistem otomatik olarak kullanıcıların kişisel tercihlerine göre ortam sıcaklığını düzenleyebilmekte ve istenilen konfor seviyesini sağlayabilmektedir.

Sistem herhangi bir kullanıcıyı tanımlayamadığında, yüz çevresine bir sınırlayıcı çerçeve çizilmekte ve 'Bilinmiyor' etiketi verilmektedir. Bu durumda, varsayılan bir ortam sıcaklığı kullanılarak klimaya komut gönderilmektedir. Sistem, "Bilinmiyor" etiketi ile tanımlanan yüzler için 23°C varsayılan ortam sıcaklığı ayarını kullanmaktadır. Bu yaklaşım, kullanıcı tanımlanamasa bile ortam konfor seviyesinin korunmasını ve klimanın etkin bir şekilde çalışmasını sağlamaktadır.

Varsayılan sıcaklık ayarı, sistemin kullanıcı dostu ve esnek olmasını desteklemektedir. Kullanıcı tanımlanmadığında ortamın belirlenen konfor seviyesinde tutulması, ani sıcaklık değişimlerinin önüne geçmekte ve enerji verimliliğini arttırmaktadır. Aynı zamanda, tanımlanamayan kullanıcılar için de optimum bir konfor seviyesinin sağlanması, sistemin genel performansını ve kullanıcı memnuniyetini yükselteceği düşünülmektedir. Bu çözüm, ev içi uygulamalarda kullanıcı tanımlama işleminin başarısız olduğu durumlarda bile sistemin etkinliğini korumasına olanak tanımaktadır. Ayrıca, 23°C derece gibi ortalama bir sıcaklık değeri seçilmesi, genel kullanıcı memnuniyetini artırırken, enerji tüketimini de dengede tutar.

Sonuç olarak, geliştirilen yüz tanıma sistemi iki ana aşamadan oluşmaktadır. İlk aşama, veri setinin oluşturulması ve sistemin eğitilmesidir. Bu aşamada, öncelikle kullanıcılar tarafından sağlanan fotoğraflar toplanır ve her fotoğraf, ilgili kişinin kimlik bilgileri ile etiketlenmektedir. Fotoğrafların çeşitli açılardan ve farklı ışık koşullarında çekilmesi, sistemin daha geniş bir veri yelpazesi üzerinde eğitilebilmesini sağlamaktadır. Bu süreçte, her kullanıcının tercih ettiği ortam sıcaklık bilgisi ve öncelik numarası da veri setine entegre edilmektedir. Toplanan ve etiketlenen bu veriler, yüz tanıma algoritmasının eğitimi için kullanılmaktadır. Eğitim aşamasında, algoritma, verilen veri seti üzerinden yüzleri tanımayı ve ilişkilendirilen sıcaklık bilgileri ile öncelik numaralarını öğrenmektedir. Bu süreç, sistemin doğru ve güvenilir bir şekilde çalışması için kritik öneme sahiptir. Şekil 3.24'te, sistemin eğitilme sürecini ve bu süreçte izlenen adımları gösteren akış diyagramı yer almaktadır.



Şekil 3. 24. Sistemin Eğitilme Akış Diyagramı

İkinci aşama, eğitilen modelin pratikte kullanılması ve gerçek zamanlı yüz tanıma işleminin gerçekleştirilmesidir. Bu süreçte, kamera tarafından yakalanan görüntüler, önceden eğitilen model ile karşılaştırılarak tanınan yüzlerin belirlenmesi sağlanmaktadır. Tanınan yüzler, öncelik numarasına göre ilgili kullanıcıların belirlediği ortam sıcaklık tercihi ile ilişkilendirilmekte ve sisteme kaydedilmektedir. Böylece, sistem her kullanıcının bireysel sıcaklık tercihini öğrenmiş olur ve bu bilgiler, kullanıcının tanınması durumunda otomatik olarak uygulanmaktadır. Ardından sistem, tanınan kişilerin ortam sıcaklık tercihlerine göre klimaya sinyal gönderir.

Bu sinyal, klimanın istenilen ortam sıcaklığını ayarlaması için gerekli komutları içerir ve klima, bu komutlara göre otomatik olarak çalışır. Bu süreçte, NTC sensörü tarafından iç ortam sıcaklığı sürekli olarak ölçülmekte ve klimaya geri bildirim verilmektedir. Bu sayede, klimanın istenilen ortam sıcaklığına ulaşp ulaşmadığı kontrol edilebilmekte ve gerektiğinde otomatik ayarlamalar yapılabilmektedir.

Geliştirilen sistem ayrıca, herhangi bir yüz tanımlanmadığında otomatik olarak iç ortam sıcaklığını varsayılan değer olan 23°C'ye ayarlamaktadır. Bu varsayılan ayar, enerji verimliliğini maksimize ederken aynı zamanda kullanıcı konforunu da korumaktadır. Kullanıcıların genel konfor seviyesi göz önünde bulundurularak belirlenen bu varsayılan sıcaklık, enerji tasarrufu sağlarken genel kullanıcı memnuniyetini de artırmaktadır.

Bu otomatik ayarlama mekanizması, hem sistemin kullanımını daha ekonomik hale getirmekte hem de kullanıcı müdahalesi gerektirmeden optimum konfor sağlamaktadır. Şekil 3.25'te gerçek zamanlı yüz tanımlama sisteminin akış diyagramı yer almaktadır.

3.2.3. Dikey Kanat Kontrolü

Klima sistemlerinde hava akışını yönlendirmek için dikey ve yatay kanatlar kullanılmaktadır. Bu kanatların pozisyonu, genellikle step motorlar aracılığıyla veya manuel olarak ayarlanmaktadır. Eğer dikey kanat step motor ile ayarlanabiliyorsa, klimanın kumandası üzerinden verilen komutlar, bu kanatların kademeli bir şekilde hareket etmesini sağlamaktadır. Dikey kanatlar genellikle 60°'lik bir açı aralığında hareket ettirilebilmektedir. Bu konfigürasyon, kullanıcıların hava akışını istedikleri yöne doğru yönlendirmelerine olanak tanımaktadır.

Bu kanatların pozisyonlarının doğru bir şekilde ayarlanması, klimanın performansı ve kullanıcı konforu açısından büyük önem taşır. Doğru yönlendirilmiş hava akışı, soğutma veya ısıtma işleminin daha etkin bir şekilde gerçekleştirilmesini sağlamaktadır. Bu nedenle, kanatların hareket mekanizması ve kontrol algoritmaları, optimum hava dağılımını ve kullanıcı memnuniyetini sağlamak üzere dikkatlice tasarlanmalıdır.

Geliştirilen projede, yapay zeka algoritmaları kullanılarak dikey kanadın otomatik olarak ayarlanması hedeflenmektedir. Bu otomatik hareket, geleneksel klima sistemlerinde olduğu gibi step motor aracılığıyla gerçekleştirilmektedir. Yapay zeka algoritması, algılanan insanın konumuna bağlı olarak dikey kanadın 60°'lik açı aralığında hareket etmesini sağlayacak şekilde tasarlanmıştır. Bu yaklaşım, ortamda bulunan bireylerin yerleşim pozisyonlarını algılayarak, hava akışını optimal bir şekilde yönlendirmeyi amaçlamaktadır.

Şekil 3.26'da step motorun 60°'lik açı aralığında saat yönünde veya saat yönünün tersinde hareket edebilmesini sağlayan kod örneği yer almaktadır. Bu kod örneği, step motor kontrolü için gerekli olan komutların nasıl verileceğini ve motorun istenilen açılara hassasiyetle nasıl hareket ettirileceğini detaylandırmaktadır. Bu sistem, özellikle geniş alanlarda veya çoklu kullanıcı senaryolarında, bireylerin konfor düzeyini optimize etmek için oldukça etkili bir çözüm sunmaktadır.

```

import RPi.GPIO as GPIO
import time

# Define Motor Pins
IN1 = 17
IN2 = 18
IN3 = 27
IN4 = 22

# Initialize Pins
def init():
    GPIO.setmode(GPIO.BCM)
    for pin in (IN1, IN2, IN3, IN4):
        GPIO.setup(pin, GPIO.OUT)

# Set Step Sequence
def set_step(w1, w2, w3, w4):
    GPIO.output(IN1, w1)
    GPIO.output(IN2, w2)
    GPIO.output(IN3, w3)
    GPIO.output(IN4, w4)

# Set Step Sequence for Rotate Clockwise and Counter Clockwise
def move_motor(steps, direction):
    cw_seq = [[1,0,0,1],[0,0,0,1],[0,0,1,1],[0,0,1,0],[0,1,1,0],[0,1,0,0],[1,1,0,0],[1,0,0,0]]
    ccw_seq = [[1,0,0,0],[1,1,0,0],[0,1,0,0],[0,1,1,0],[0,0,1,0],[0,0,1,1],[0,0,0,1],[1,0,0,1]]
    sequence = cw_seq if direction == 'right' else ccw_seq
    init()
    try:
        for _ in range(steps):
            for step in sequence:
                set_step(*step)
                time.sleep(0.002)
    except KeyboardInterrupt:
        GPIO.cleanup()

def main():
    try:
        while True:
            # Rotate 60 degrees clockwise
            print("Saat yönünde Hareket!")
            move_motor(85, 'right') # 512 steps for 360 degrees and 85 steps for 60 degrees
            time.sleep(2)

            # Rotate 60 degrees counterclockwise
            print("Saat Yönünün Tersinde Hareket!")
            move_motor(85, 'left') # 512 steps for 360 degrees and 85 steps for 60 degrees
            time.sleep(2)

    except KeyboardInterrupt:
        GPIO.cleanup()

if __name__ == "__main__":
    main()

```

Şekil 3. 26. Step Motor Hareket Testi

Şekil 3.26 yer alan kod örneğinde step motor yarım adım modunda ayarlanmıştır. Böylelikle açısal hareketlerin daha hassas bir şekilde yapılması amaçlanmaktadır. Half step modu, step motorunun adımını ikiye bölen bir sürüş modudur, her bir adımda motor iki ayrı konum arasında geçiş yapar. Bu mod, daha yüksek çözünürlük sağlayarak motorun daha hassas hareketler yapmasını ve ince konum ayarlamaları gerçekleştirmesini sağlamaktadır. Ayrıca, titreşimi azaltarak daha sessiz bir çalışma sağlamaktadır [44].

3.2.4. Gerçek Zamanlı İnsan Algılama Sisteminin Geliştirilmesi

Yapay zeka destekli insan algılama sistemi geliştirmek, modern teknolojinin birleşimi ve bilgisayar biliminin en son yeniliklerinin birleşimini gerektirmektedir. Bu bağlamda bu sistemi, Raspberry Pi gibi kompakt bir platforma entegre edebilmek için Tensorflow makine öğrenme algoritması ve YOLO nesne algılama modeli kullanılmaktadır. Tensorflow ve YOLO bu sistemin temelini oluşturan kritik unsurlardır. TensorFlow'un sunduğu veri eğitimi, transfer eğitimi ve nesne izleme gibi destekler, sistem tasarımında esneklik sağlamak ve çeşitli nesne algılama modellerinin entegrasyonunu mümkün kılmaktadır [63]. TensorFlow'un bu çok yönlülüğü, sistemi geliştirme ve özelleştirme sürecinde önemli bir avantaj sunmaktadır. Ayrıca, YOLO'nun hızlı işleme süreleri ve yüksek doğruluk avantajları, sistemin gerçek zamanlı insan algılama görevlerini başarıyla yerine getirmesine olanak tanımaktadır [10]. Bu kombinasyon, taşınabilir bir cihaz üzerinde güçlü bir insan algılama sistemini mümkün kılarken kullanıcılara güvenilir bir algılama deneyimi sunması amaçlanmaktadır.

Geliştirilen gerçek zamanlı insan algılama sisteminde YOLOv4 Tiny modeli tercih edilmiştir. YOLOv4 Tiny modeli, son derece hızlı ve hafif yapısıyla öne çıkmaktadır. Bu özellik, sınırlı işlem gücüne sahip cihazlarda bile etkili bir nesne algılama performansı sağlamaktadır. Ayrıca, YOLOv4 Tiny modeli, düşük güç tüketimi ve düşük bellek gereksinimleri ile dikkat çekmektedir. Bu özellikler, Raspberry Pi gibi gömülü sistemlerde ve mobil cihazlarda enerji verimliliği sağlamak ve sistemlerin uzun süre kesintisiz çalışmalarına olanak tanımaktadır [69].

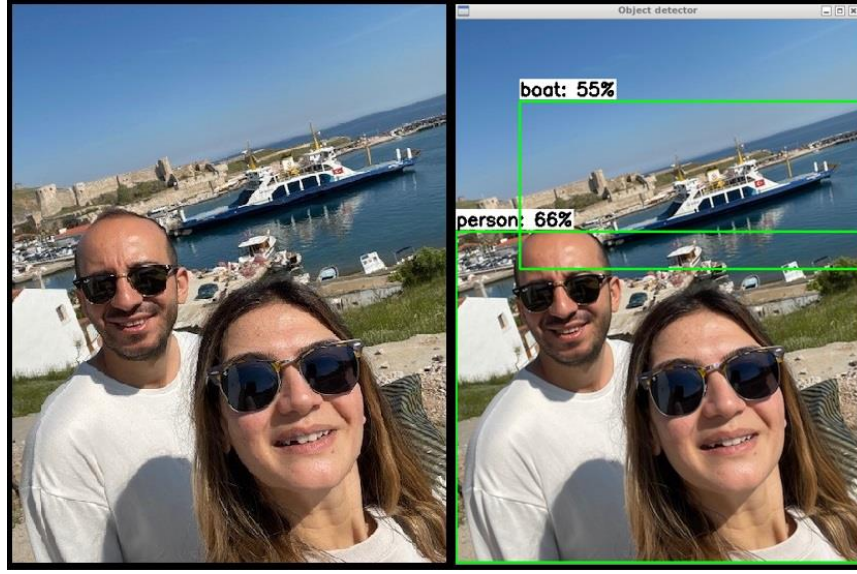
YOLOv4 Tiny'nin hızı ve doğruluğu, gerçek zamanlı nesne algılama uygulamalarında yüksek performanslı ve güvenilir sonuçlar elde etme imkânı sunmaktadır [69]. Tüm bu faktörler, YOLOv4 Tiny'nin, sınırlı işlem gücüne sahip cihazlarda bile iyi bir performans ile çalışabilmesi ve yüksek doğruluklu bir nesne algılama çözümleri sunabilmesi, bu modeli ideal bir seçenek haline getirmektedir.

3.2.4.1.Nesne Algılama Algoritmasının Geliştirilmesi

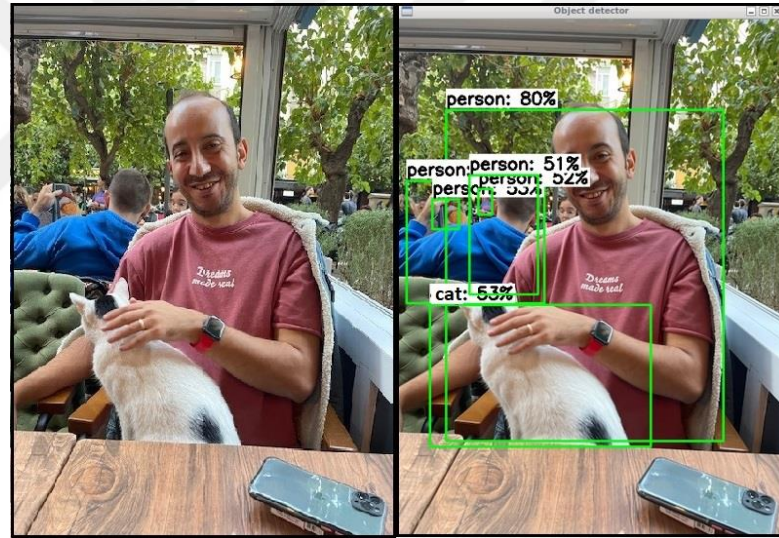
Raspberry Pi üzerinden nesne tespiti yapmak için TensorFlow ve OpenCV kütüphanelerini kullanan bir uygulama geliştirilmiştir. Bu uygulamada, YOLOv4 Tiny modeli nesne tespiti için kullanılmaktadır. YOLOv4 Tiny modeli, Microsoft COCO veri seti ile eğitilmiştir. COCO veri seti, görüntü tabanlı makine öğrenimi algoritmalarının eğitimi ve değerlendirmesi için kullanılan önemli bir veri setidir. Bu veri seti, 80 farklı nesne kategorisini kapsayan ve toplamda 1.5 milyon etiketlenmiş görüntü içermektedir. Bu görüntülerde insanlar, hayvanlar, araçlar, mobilyalar ve diğer çeşitli nesneler yer almaktadır. COCO, 328,000'den fazla eğitim görüntüsü, 40,000 doğrulama görüntüsü ve 20,000 test görüntüsünden oluşan bir veri setine sahiptir [70]. Bu görüntüler, çeşitli bağlamlarda ve arka planlarda çekilmiş gerçek dünya sahnelerini yansıtmakta ve zorlu nesne algılama ve sınıflandırma senaryolarını içermektedir. COCO'nun sunduğu geniş kapsamlı veriler, görüntü işleme ve derin öğrenme modellerinin eğitilmesi ve değerlendirilmesi için değerli bir kaynak sağlamaktadır [71].

Nesne algılama algoritmasının performansını değerlendirmek için öncelikle kaydedilmiş görüntülerde bulunan nesneler üzerinde algoritma test edilmiştir. Bu aşamada, algoritmanın çeşitli senaryolarda nasıl tepki verdiği ve nesneleri ne kadar doğru bir şekilde tanımladığı detaylı bir şekilde incelenmiştir. Daha sonra, algoritmanın gerçek zamanlı uygulamalardaki performansı canlı kamera görüntüleri üzerinde değerlendirilmiştir.

Canlı kamera görüntülerindeki nesnelerin tanınma doğruluğu incelenmiştir. Geliştirilen algoritmanın günlük kullanım senaryolarında ne kadar başarılı olduğu değerlendirilmiştir. Geliştirilen YOLO modeli nesne algılama algoritmasına farklı görseller eklendiğinde, elde edilen çıktılar Şekil 3.24 ve Şekil 3.25'te yer almaktadır.

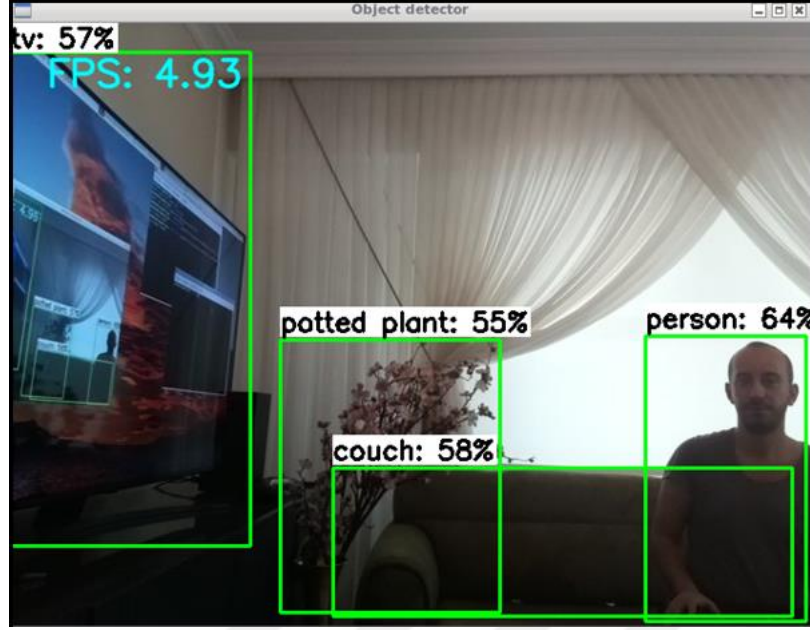


Şekil 3. 27. Fotoğraflardaki Farklı Nesnelerin Algılanması 1



Şekil 3. 28. Fotoğraflardaki Farklı Nesnelerin Algılanması 2

Geliştirilen nesne algılama algoritması çeşitli görsellerle başarılı bir şekilde test edildikten sonra, algoritma gerçek zamanlı kamera görüntüsünde de performansı değerlendirilmiştir. Elde edilen sonuçlar, Şekil 3.26'da yer almaktadır. Gerçek zamanlı testler, algoritmanın pratikte ne kadar etkili olduğunu belirlemek için gerçekleştirilmiştir. Şekil 3.26'daki çıktılar, nesne algılama algoritmasının canlı ortamda nesneleri doğru bir şekilde tanıma yeteneğini göstermektedir.



Şekil 3. 29. Gerçek Zamanlı Görüntülerdeki Nesnelerin Algılanması

3.2.4.2. İnsan Algılama Algoritmasının Geliştirilmesi

Geliştirilen nesne algılama algoritmasından kullanılan YOLO modeli COCO veri seti ile eğitilmiştir. COCO veri setinde birçok farklı nesne sınıfı algılanabilmektedir. Geliştirdiğim sistemde ise sadece insanların algılanması amaçlanmaktadır. COCO veri setinde insanları tanımlamak için “person” etiketi kullanılmaktadır. Bu nedenle algoritmanın insanları algılaması için öncelikle veri setinde yer alan “person” etiketinin indeks numarasının tespit edilmesi gerekmektedir. Daha sonra her bir algılanan nesnenin algılama sonucu incelenmektedir.

Sonuç olarak algılanan nesneye verilen etiket indeks numarası “person” etiketinin indeks numarası ile karşılaştırılmaktadır. Eğer algılanan nesne "person" etiketini temsil eden indeks numarasına sahip değilse, algoritma bu nesneyi görmezden gelerek ekranda tanımlamasını yapmamaktadır. Eğer algılanan nesne “person” etiketinin indeks numarasına sahipse algılanan nesnelerin etrafına sınırlayıcı bir çerçeve çizilerek etiketlenmektedir. Bu sayede geliştirilen algoritmanın sadece insanları algılayıp çıktı vermesi sağlanmaktadır. Şekil 3.30'da geliştirilen algoritmanın doğru nesneleri tespit etme ve işleme sürecini gösteren bir kod örneği yer almaktadır.

Bu örnek, algoritmanın sadece insanları algılayıp çıktı olarak sunmasını sağlamaktadır. Geliştirilen insan algılama algoritmasının çalışma sürecini gösteren akış şeması Şekil 3.31’de yer almaktadır.

```
# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]

# Find person label index
person_index = labels.index("person")

while True:

    # Retrieve detection results
    boxes = interpreter.get_tensor(output_details[boxes_idx]['index'])[0] # Bounding box coordinates of detected objects
    classes = interpreter.get_tensor(output_details[classes_idx]['index'])[0] # Class index of detected objects
    scores = interpreter.get_tensor(output_details[scores_idx]['index'])[0] # Confidence of detected objects

    # Loop over all detections and draw detection box if confidence is above minimum threshold
    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

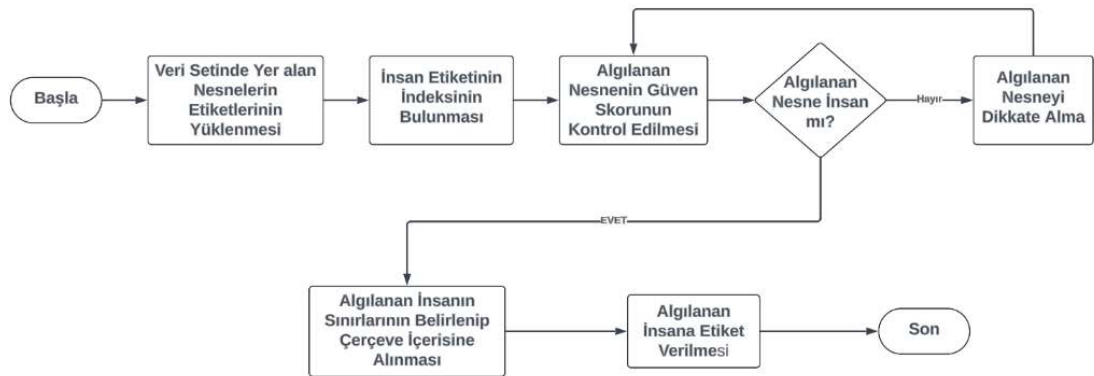
            # If detected object is not a person, skip
            if int(classes[i]) != person_index:
                continue

            # Get bounding box coordinates and draw box
            ymin = int(max(1, (boxes[i][0] + imH)))
            xmin = int(max(1, (boxes[i][1] + imW)))
            ymax = int(min(imH, (boxes[i][2] + imH)))
            xmax = int(min(imW, (boxes[i][3] + imW)))

            cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

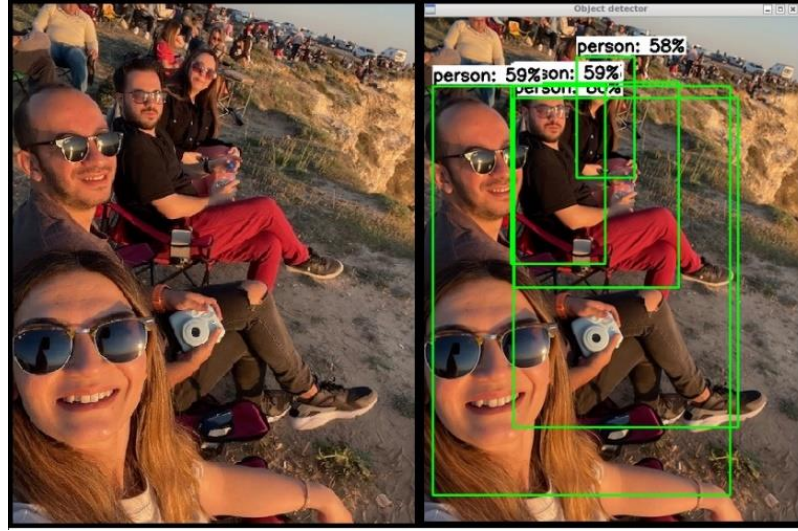
            # Draw label
            object_name = labels[int(classes[i])]
            label = "%s: %d%%" % (object_name, int(scores[i]*100))
            labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
            label_ymin = max(ymin, labelSize[1] + 10)
            cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED)
            cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
```

Şekil 3. 30. Algılanan Nesnelerden İnsanların Ayırıştırılması



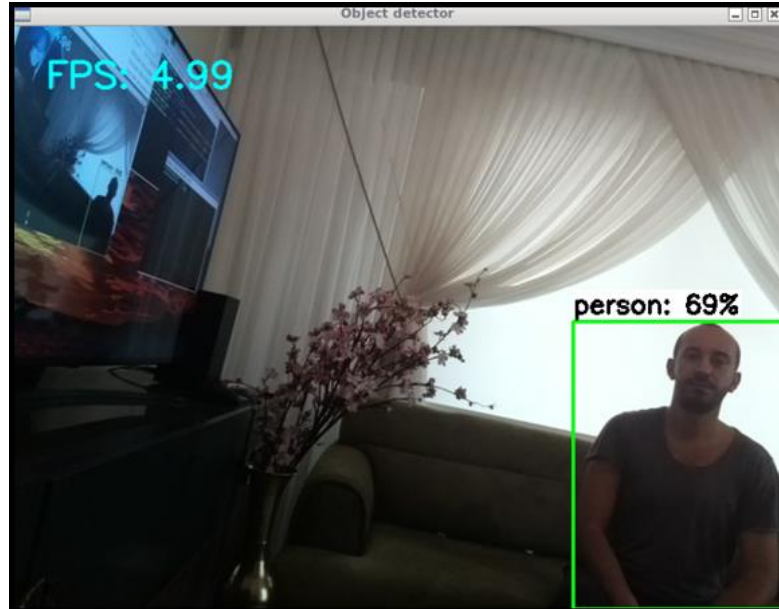
Şekil 3. 31. İnsan Algılama Algoritması Akış Şeması

Geliştirilen nesne algılama algoritmasının sadece insanları algılaması için gerekli olan güncellemeler yapıldıktan sonra algoritma öncelikle kaydedilmiş görüntüler ile test edilmiştir. Şekil 3.32’de güncellenen algoritmanın çıktıları yer almaktadır.



Şekil 3. 32. Fotoğrafta Yer Alan İnsanların Tespit Edilmesi

Geliştirilen insan algılama algoritması, kaydedilmiş görüntüler ile test edildikten sonra gerçek zamanlı görüntüler üzerinde test edilmiştir. Bu testler, algoritmanın dinamik bir ortamda nasıl davrandığını ve farklı koşullar altında ne kadar etkili olduğunu değerlendirmek için yapılmıştır. Gerçek zamanlı bir görüntüde algoritmanın verdiği çıktı Şekil 3.33’te yer almaktadır.



Şekil 3. 33. Gerçek Zamanlı İnsan Tespiti

3.2.4.3. Algılanan İnsanların Konumunun Belirlenmesi

Yapılan güncellemeyle COCO veri setiyle geliştirilen nesne algılama algoritması, yalnızca insanları algılamak üzere özelleştirilmiş, diğer nesneleri algılamamak üzere optimize edilmiştir. Bu değişiklik, algoritmanın yalnızca belirlenmiş bir sınıf olan insanları tespit etmesini sağlamaktadır. Böylelikle istenmeyen nesnelerin görmezden gelinmesi mümkün olmaktadır. Özellikle akıllı klima ürününe entegre edilmesi planlanan bu sistem, algılanan insana göre dikey hava kanatlarını otomatik olarak ayarlamaktadır. Bunu yapılabilmesi için algılanan insanın konumunun da teyit edilmesi gerekmektedir. Şekil 3.33'te gösterildiği gibi algoritma, kamera karşısında bir insan tespit ettiğinde, algıladığı insanın etrafına bir sınırlayıcı çerçeve yerleştirmektedir. Bu çerçevenin koordinatları, algılanan nesnenin konumu ile ilgili bilgi vermektedir. Bu koordinatlar kullanılarak nesnenin merkezi hesaplanmaktadır.

Nesnenin merkezi hesaplanırken, sınırlayıcı çerçevenin sol üst köşesinin koordinatları (Algoritmada bu koordinat x_{min} ve y_{min} değişkeni ile belirtilmiştir.) ve sağ alt köşesinin koordinatları (Algoritmada bu koordinat x_{max} ve y_{max} değişkeni ile belirtilmiştir.) referans alınmaktadır. Bu koordinatlar arasındaki yatay ve dikey mesafelerin ortalamaları alınarak nesnenin merkezi tespit edilebilmektedir. Yatayda nesnenin merkezi için 3.8 numaralı formül, dikeyde ise 3.9 numaralı formül kullanılmaktadır. Bu hesaplamalar sayesinde algılanan nesnenin tam merkezi bulunarak konumlandırılması sağlanmaktadır.

$$\frac{(x_{min} + x_{max})}{2} \quad (3.8)$$

$$\frac{(y_{min} + y_{max})}{2} \quad (3.9)$$

Nesnenin merkezi, ekranın merkeziyle karşılaştırılarak nesnenin ekranın sağında mı yoksa solunda mı olduğu tespit edilmektedir. Öncelikle algoritmada

kameranın görüntüsünün yansıtılacağı alanın çözünürlüğü belirlenmektedir. Daha sonra belirlenen çözünürlüğün genişlik ve yükseklik değerleri kullanılarak ekranın merkezi hesaplanmaktadır. Ekranın merkezi, ekran genişliğinin yarısı ve ekran yüksekliğinin yarısı olarak hesaplanmaktadır. Nesnenin merkezi, ekranın merkeziyle karşılaştırılarak nesnenin ekranın hangi tarafında olduğu belirlenmektedir.

Nesnenin konumu belirlendikten sonra, eğer nesne ekranın solunda ise çerçeve kırmızı renkte, sağında ise çerçeve yeşil renkte çizilmektedir. Bu, sistemin test edilme sürecinde nesnenin konumun tespitinin hızlı ve kolay bir şekilde doğrulanabilmesine olanak tanınmaktadır. Şekil 3.34'te, algılanan insanların konumlarının tespit edilmesi ve ekranın hangi tarafında olduğunun belirlenmesini sağlayan kod örneği yer almaktadır. Bu kod, nesnelerin ekran üzerindeki konumlarını analiz ederek, çerçeve renklerini dinamik olarak belirlemektedir. Ayrıca, geliştirilen bu konum algılama algoritmasının çalışma sürecini gösteren akış şeması, Şekil 3.35'te detaylı bir şekilde sunulmaktadır. Bu akış şeması, algoritmanın işleyişini ve adım adım nasıl çalıştığını görsel olarak özetlemekte olup, nesne algılama ve çerçeve renklendirme süreçlerini göstermektedir.

```
imw = 640
imh = 480

person_index = labels.index("person")

#For determine the object is on the left side or right side of the screen Last Time
last_object_side = None

while True:
    #For determine the object is on the left side or right side of the screen First Time
    object_side = None

    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

            # If detected object is not a person, skip
            if int(classes[i]) != person_index:
                continue

            # Get bounding box coordinates and draw box
            ymin = int(max(1, (boxes[i][0] * imh)))
            xmin = int(max(1, (boxes[i][1] * imw)))
            ymax = int(min(imh, (boxes[i][2] * imh)))
            xmax = int(min(imw, (boxes[i][3] * imw)))

            # Calculate object center
            object_center_x = (xmin + xmax) // 2

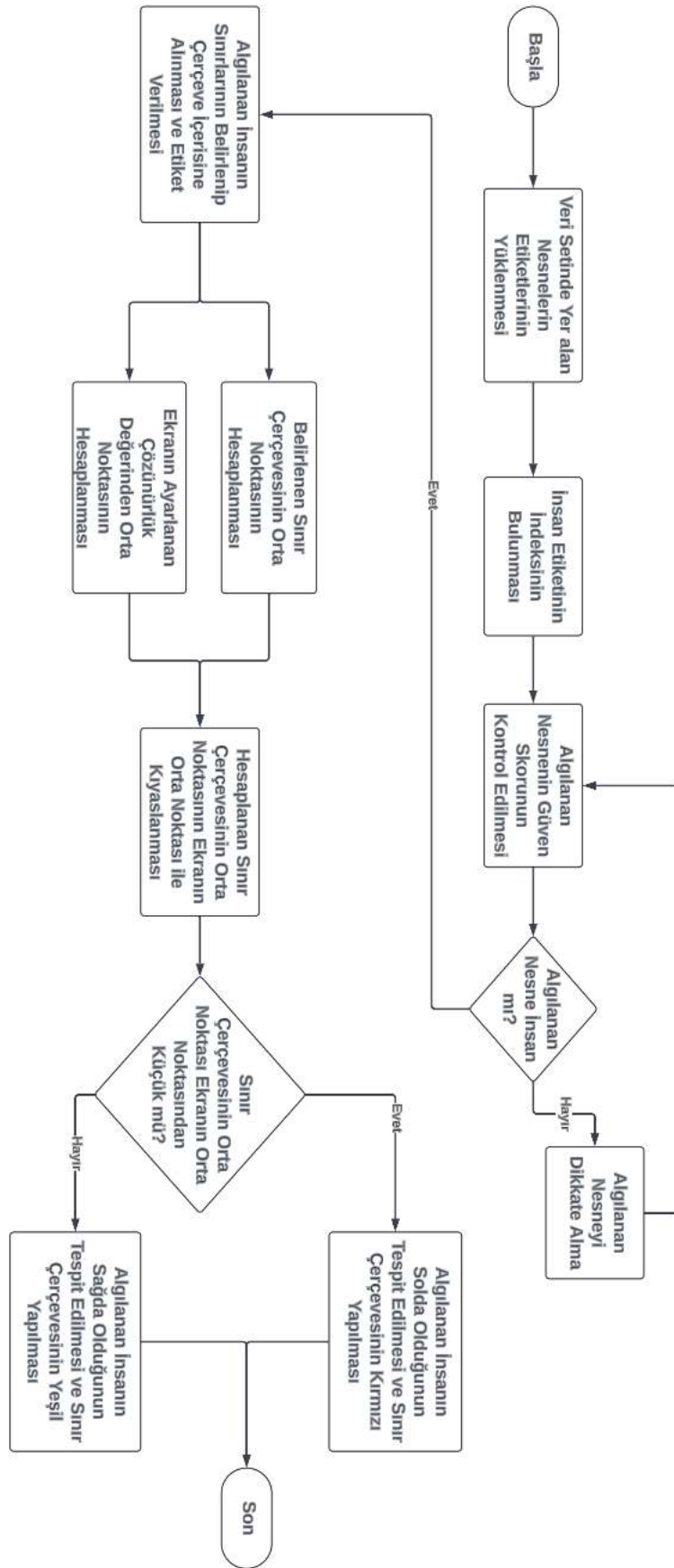
            # Define screen center
            screen_center_x = imw // 2

            # Check if object is on the left or right side of the screen
            if object_center_x < screen_center_x:
                # Draw red rectangle if object is on the left side
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 0, 255), 2)
                object_side = 'left'
            else:
                # Draw green rectangle if object is on the right side
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 255, 0), 2)
                object_side = 'right'

            # Draw label
            object_name = labels[int(classes[i])]
            label = '%s: %d%%' % (object_name, int(scores[i]*100))
            labelSize, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
            label_ymin = max(ymin, labelSize[1] + 10)
            cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseline-10), (255, 255, 255), cv2.FILLED)
            cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)

    if object_side is not None and object_side != last_object_side:
        last_object_side = object_side
```

Şekil 3. 34. Algılanan İnsanların Konumunun Tespit Edilmesi

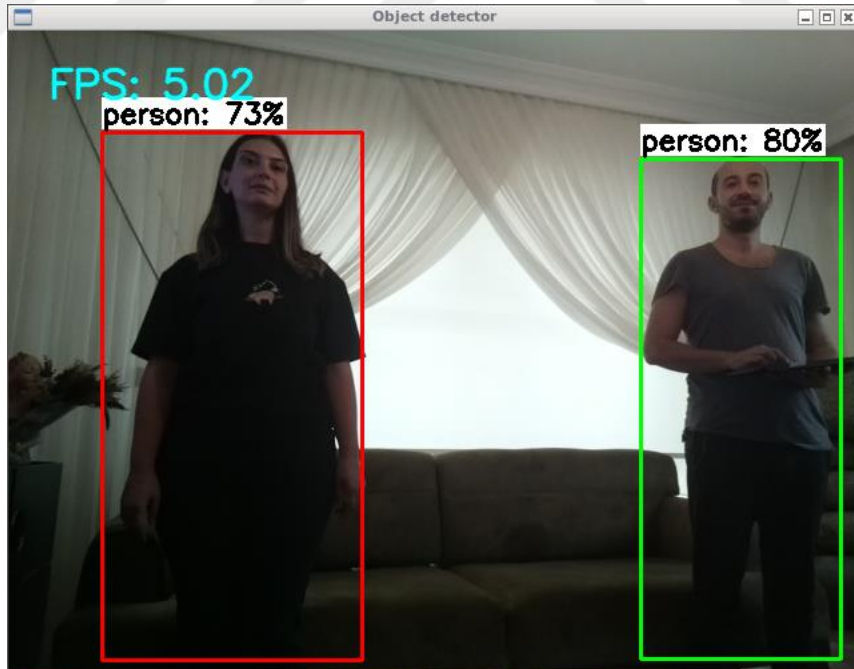


Şekil 3. 35. Algılanan İnsanların Konum Tespit Algoritması Akış Şeması

Sınırlayıcı çerçevelerin algılanan insanın konumuna göre renk deęiřtirecek řekilde ayarlanması, geliřtirilen algoritmanın test sũrecinde algılanan insanın konumunun doęruluęunu daha etkili ve kolay bir řekilde deęerlendirilebilmesine olanak tanımaktadır. řekil 3.36 ve řekil 3.37’de algoritmanın, ekranın farklı bũlgelerinde algılanan insanlar iin ũrettięi ıktılar yer almaktadır.



řekil 3. 36. Gerek Zamanlı Algılanan İnsanın Konumunun Tespit Edilmesi



řekil 3. 37. Gerek Zamanlı Birden ok İnsanın Konumunun Tespit Edilmesi

3.2.4.4. Konum Tabanlı Dikey Kanat Yönlendirilmesi

Geliştirilen sistemde, sırası ile sadece insanlar algılanıp daha sonra algılanan insanların konumlarının tespit edilmesi sağlanmıştır. Bu sistemin bir klimaya entegre edilmesi düşünüldüğü için klimada hava akışını yönlendiren dikey hava kanadının hareketini sağlayan step motorun, algılanan insanın konumuna göre kontrol edilmesi amaçlanmaktadır.

Algılanan insanın konumu, belirli bir süre boyunca izlenmektedir. Eğer 1 dakika boyunca algılanan insanın sol tarafta olduğu tespit edilirse, step motor sağa doğru döndürülerek dikey kanadı bu yöne hareket ettirmektedir. Bu şekilde, insanı takip etmek için sürekli bir yönlendirme sağlanmaktadır. Eğer insan sağ tarafa geçip 1 dakika boyunca durursa, step motor sola dönerek dikey kanadı bu sefer sola doğru hareket ettirmektedir. Böylelikle insanların doğrudan hava akışına maruz kalmaları önlenerek, konfor seviyesinin artırılması sağlanmaktadır.

Şekil 3.34'te gösterilen algoritmaya step motor entegrasyonu yapılarak algılanan insanın konumuna göre step motorun hareketi sağlanmaktadır. Geliştirilen algoritmaya eklenen step motor kontrolünün kod örneği Şekil 3.38'de yer almaktadır. Algoritmanın çalışma sürecini gösteren akış şeması Şekil 3.39'da yer almaktadır.

```

imw = 640
imh = 480

def move_motor(steps, direction):
    cw_seq = [[1,0,0,1],[0,0,0,1],[0,0,1,1],[0,0,1,0],[0,1,1,0],[0,1,0,0],[1,1,0,0],[1,0,0,0]]
    ccw_seq = [[1,0,0,0],[1,1,0,0],[0,1,0,0],[0,1,1,0],[0,0,1,0],[0,0,1,1],[0,0,0,1],[1,0,0,1]]
    sequence = cw_seq if direction == 'right' else ccw_seq
    init()
    try:
        for _ in range(steps):
            for step in sequence:
                set_step(*step)
                time.sleep(0.002)

person_index = labels.index("person")
last_object_side = None
left_side_timer = 0
right_side_timer = 0
left_turn_done = False
right_turn_done = False

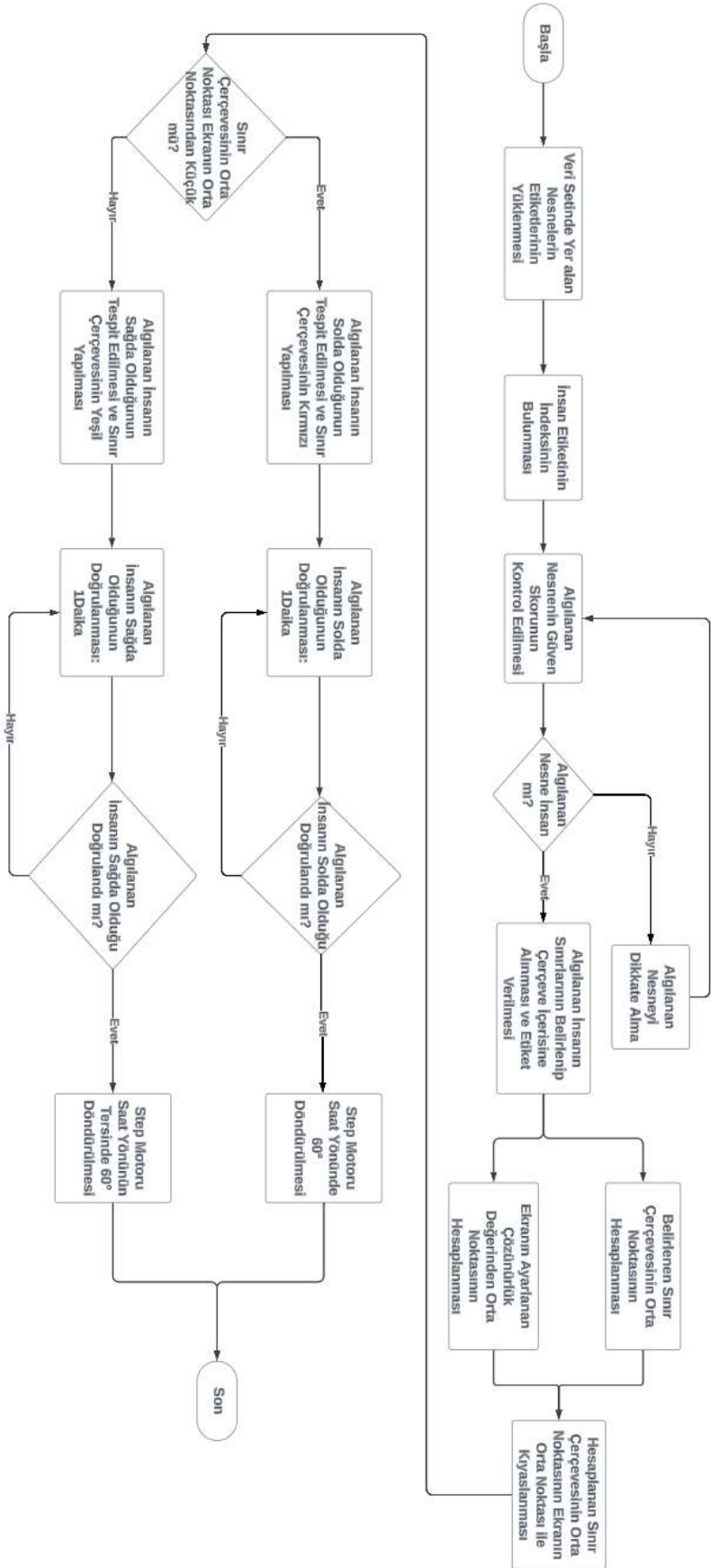
while True:
    object_side = None
    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):
            # If detected object is not a person, skip
            if int(classes[i]) != person_index:
                continue
            # Get bounding box coordinates and draw box
            ymin = int(max(1,(boxes[i][0] * imh)))
            xmin = int(max(1,(boxes[i][1] * imw)))
            ymax = int(min(imh,(boxes[i][2] * imh)))
            xmax = int(min(imw,(boxes[i][3] * imw)))
            # Calculate object center
            object_center_x = (xmin + xmax) // 2
            # Define screen center
            screen_center_x = imw // 2
            # Check if object is on the left or right side of the screen
            if object_center_x < screen_center_x:
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 0, 255), 2)# Draw red rectangle if object is on the left side
                object_side = 'left'
            else:
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 255, 0), 2)# Draw green rectangle if object is on the right side
                object_side = 'right'

            # Draw label
            object_name = labels[int(classes[i])]
            label = '%s: %d%%' % (object_name, int(scores[i]*100))
            labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
            label_ymin = max(ymin, labelSize[1] + 10)
            cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED)
            cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)

    if object_side is not None:
        if object_side != last_object_side:
            last_object_side = object_side
            left_side_timer = 0
            right_side_timer = 0
            left_turn_done = False
            right_turn_done = False
        else:
            if object_side == 'right' and not right_turn_done:
                left_side_timer = 0
                right_side_timer += 1
                if right_side_timer == 60: # People detect right side on the screen until 1 min
                    move_motor(85, 'left') # 512 steps for 360 degrees and 85 steps for 60 degrees
                    right_side_timer = 0
                    right_turn_done = True
                    # Cut the power of motor when motors rotation completed
                    GPIO.output(IN1, GPIO.LOW)
                    GPIO.output(IN2, GPIO.LOW)
                    GPIO.output(IN3, GPIO.LOW)
                    GPIO.output(IN4, GPIO.LOW)
            elif object_side == 'left' and not left_turn_done:
                right_side_timer = 0
                left_side_timer += 1
                if left_side_timer == 60: # People detect left side on the screen until 1 min
                    move_motor(85, 'right') # 512 steps for 360 degrees and 85 steps for 60 degrees
                    left_side_timer = 0
                    left_turn_done = True
                    # Cut the power of motor when motors rotation completed
                    GPIO.output(IN1, GPIO.LOW)
                    GPIO.output(IN2, GPIO.LOW)
                    GPIO.output(IN3, GPIO.LOW)
                    GPIO.output(IN4, GPIO.LOW)

```

Şekil 3. 38. Algılanan İnsanın Konumuna Göre Step Motor Kontrolü



Şekil 3. 39. İnsan Konumuna Göre Step Motor Kontrol Algoritması Akış Şeması

Birden fazla insanın bulunduğu ortamlarda geliştirilen sistemin etkinliği, algılanan insan sayısını dikkate alarak optimize edilmiştir. Bu sistem, insanların konumunu izleyerek, ekranın sağ ve sol bölgelerindeki insanların tespitini gerçekleştirmekte ve her bir insana birer birim olarak değer atamaktadır. Sağ ve sol bölgelerdeki insan sayısı arasındaki farklılıklar dinamik olarak değerlendirilmektedir.

Örneğin, sol tarafta üç kişi bulunurken sağ tarafta bir kişi algılandığında, sol taraftaki insan popülasyonu daha yoğun olduğu için sistem, hava akışını sağa yönlendirmek üzere step motoru hareket ettirmektedir. Eğer her iki tarafta da eşit sayıda insan varsa, hava akışını dengelemek amacıyla dikey kanatlar ortalanacak şekilde yönlendirilmektedir.

Bu optimizasyon süreci, yalnızca insan sayısını değil, aynı zamanda bireylerin hareketlerini de dikkate alarak daha dinamik bir kontrol sağlamaktadır. Örneğin, bir birey bir konumdan diğerine geçerse veya yeni bir birey alana girerse, sistem anında bu değişiklikleri algılamakta ve hava akışını yeniden ayarlamaktadır. Ancak, bu ayarlamalar, algılanan insanın o bölgede belirli bir süre kaldığını doğruladıktan sonra gerçekleştirilmektedir. Bu doğrulama süreci, gereksiz dikey kanat hareketlerini önlemekte ve daha stabil bir kontrol mekanizması sağlamaktadır. Bu sayede sistem, sürekli hareket eden dikey kanatlar yerine, enerji tasarrufu sağlayan ve kullanıcı konforunu artıran bir kontrol mekanizması sunmaktadır.

Algoritmanın esnekliği, algılanan insan sayısındaki anlık değişimlere hızla yanıt verme yeteneği ile birleştirilmiştir. Bu sayede, birden fazla insanın bulunduğu ortamlarda bile etkili bir hava sirkülasyonu sağlanırken, kullanıcıların konforu her zaman ön planda tutulmaktadır. Şekil 3.40'ta ekranın sağında ve solunda algılanan insanların sayısının tespit edilmesini sağlayan kod örneği yer almaktadır. Algılanan insanların sayısını hesaplamak amacı ile geliştirilen kodun çalışma sürecini detaylandırmak ve görselleştirmek üzere Şekil 3.41'de algoritmanın akış şeması sunulmaktadır.

```

imW = 640
imH = 480

def move_motor(steps, direction):
    cw_seq = [[1,0,0,1],[0,0,0,1],[0,0,1,1],[0,0,1,0],[0,1,1,0],[0,1,0,0],[1,1,0,0],[1,0,0,0]]
    ccw_seq = [[1,0,0,0],[1,1,0,0],[0,1,0,0],[0,1,1,0],[0,0,1,1],[0,0,0,1],[1,0,0,1]]
    sequence = cw_seq if direction == 'right' else ccw_seq
    init()
    try:
        for _ in range(steps):
            for step in sequence:
                set_step(*step)
                time.sleep(0.002)

person_index = labels.index("person")
last_object_side = None
left_side_timer = 0
right_side_timer = 0
left_turn_done = False
right_turn_done = False

while True:

    object_side = None
    # Initialize variables to keep track of the number of people on the left and right sides of the screen
    num_people_left = 0
    num_people_right = 0
    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

            # If detected object is not a person, skip
            if int(classes[i]) != person_index:
                continue

            # Get bounding box coordinates and draw box
            ymin = int(max(1, (boxes[i][0] * imH)))
            xmin = int(max(1, (boxes[i][1] * imW)))
            ymax = int(min(imH, (boxes[i][2] * imH)))
            xmax = int(min(imW, (boxes[i][3] * imW)))

            # Calculate object center
            object_center_x = (xmin + xmax) // 2
            # Define screen center
            screen_center_x = imW // 2
            # Check if object is on the left or right side of the screen
            if object_center_x < screen_center_x:
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 0, 255), 2) # Draw red rectangle if object is on the left side
                num_people_left += 1 # Increment the number of people on the left side
                object_side = 'left'
            else:
                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 255, 0), 2) # Draw green rectangle if object is on the right side
                num_people_right += 1 # Increment the number of people on the right side
                object_side = 'right'

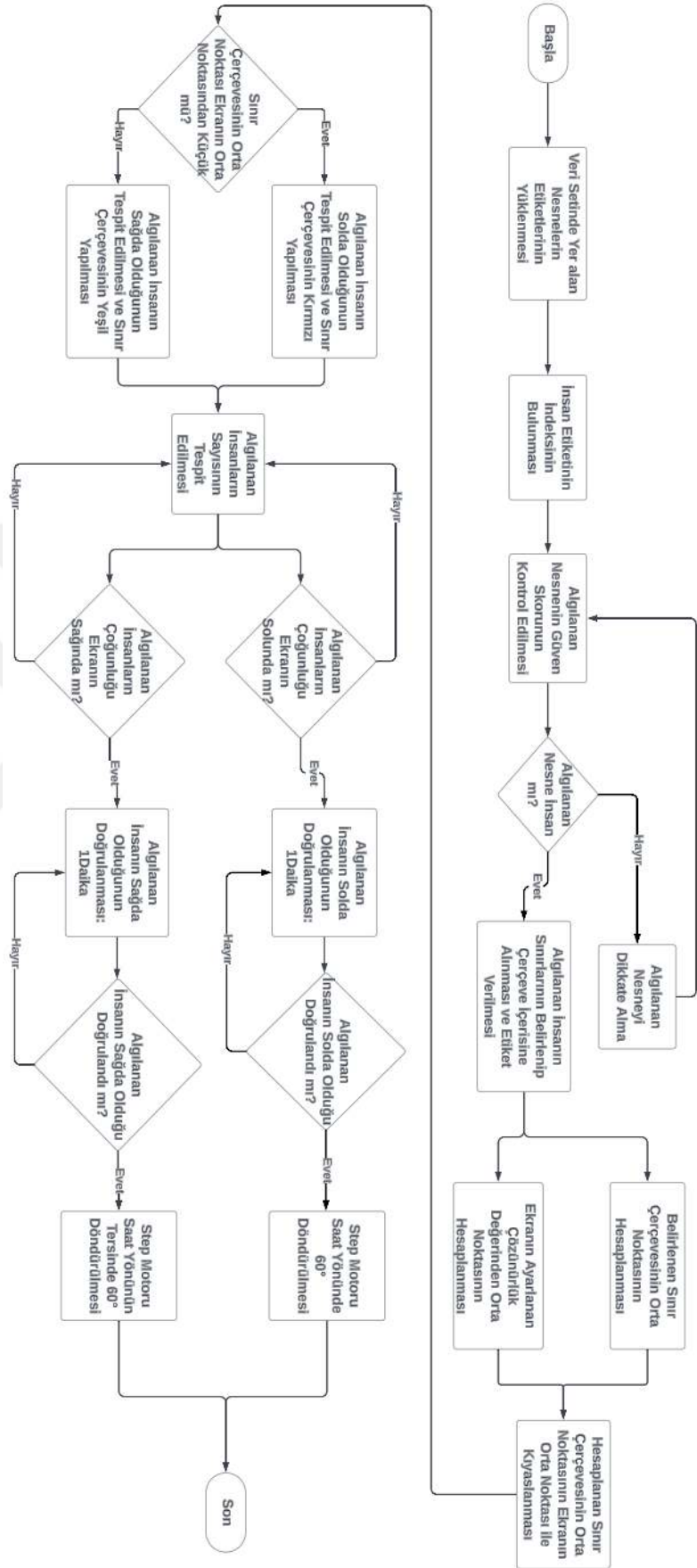
            # Draw label
            object_name = labels[int(classes[i])]
            label = '%s: %d%%' % (object_name, int(scores[i]*100))
            labelSize, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
            label_ymin = max(ymin, labelSize[1] + 10)
            cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseline-10), (255, 255, 255), cv2.FILLED)
            cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)

    # Draw the number of people on the left side of the screen
    cv2.putText(frame, 'Left: {}'.format(num_people_left), (50, imH - 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    # Draw the number of people on the right side of the screen
    cv2.putText(frame, 'Right: {}'.format(num_people_right), (imW - 200, imH - 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    if object_side is not None:
        if object_side != last_object_side:
            last_object_side = object_side
            left_side_timer = 0
            right_side_timer = 0
            left_turn_done = False
            right_turn_done = False
        else:
            if object_side == 'right' and not right_turn_done:
                left_side_timer = 0
                right_side_timer += 1
                if right_side_timer == 60: # People detect right side on the screen until 1 min
                    move_motor(85, 'left') # 512 steps for 360 degrees and 85 steps for 60 degrees
                    right_side_timer = 0
                    right_turn_done = True
                    # Cut the power of motor when motors rotation completed
                    GPIO.output(IN1, GPIO.LOW)
                    GPIO.output(IN2, GPIO.LOW)
                    GPIO.output(IN3, GPIO.LOW)
                    GPIO.output(IN4, GPIO.LOW)
            elif object_side == 'left' and not left_turn_done:
                right_side_timer = 0
                left_side_timer += 1
                if left_side_timer == 60: # People detect left side on the screen until 1 min
                    move_motor(85, 'right') # 512 steps for 360 degrees and 85 steps for 60 degrees
                    left_side_timer = 0
                    left_turn_done = True
                    # Cut the power of motor when motors rotation completed
                    GPIO.output(IN1, GPIO.LOW)
                    GPIO.output(IN2, GPIO.LOW)
                    GPIO.output(IN3, GPIO.LOW)
                    GPIO.output(IN4, GPIO.LOW)

```

Şekil 3. 40. Algılanan İnsanların Sayısının Tespiti

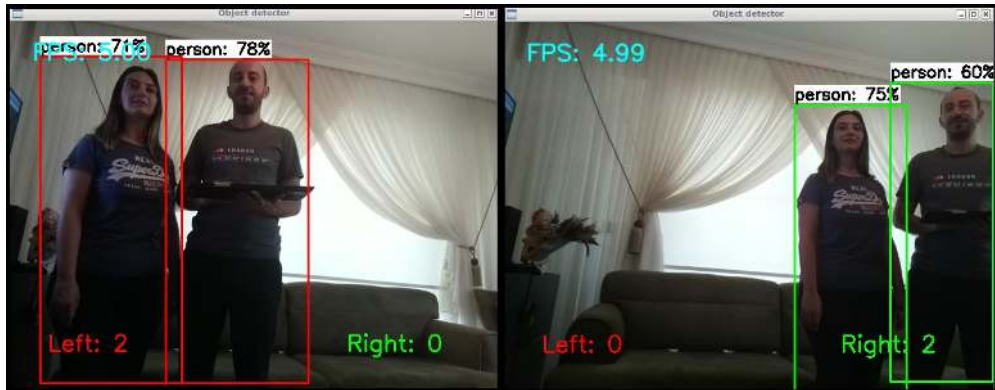


Şekil 3. 41. Algılanan İnsan Sayısının Hesaplanma Algoritması Akış Şeması

Şekil 3.40'ta sunulan algoritmanın temelinde yer alan "num_people_left" ve "num_people_right" değişkenleri, algılanan insanların konumlarına bağlı olarak güncellenmektedir. Sol tarafta algılanan insanlar için "num_people_left" değişkeni artarken, sağ tarafta algılananlar için "num_people_right" değişkeni artmaktadır. Bu değişkenlerin güncellenmesiyle, her iki bölgede algılanan insan sayısı dinamik olarak tespit edilebilmektedir. Ekranın farklı bölgelerinde algılanan insanların sayısına ilişkin geliştirilen sistemin verdiği çıktılar Şekil 3.42 ve Şekil 3.43'te yer almaktadır.



Şekil 3. 42. Gerçek Zamanlı Birden Çok İnsanın Algılanması 1. Durum



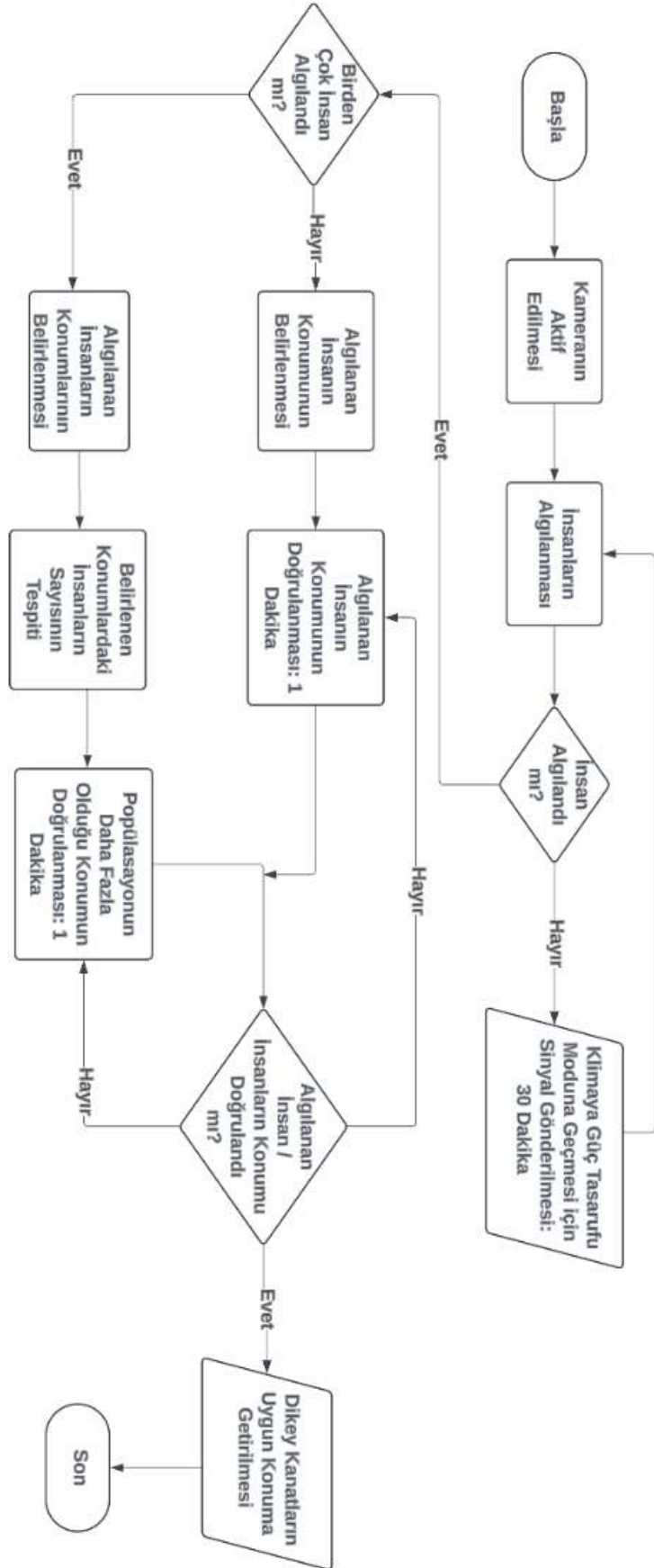
Şekil 3. 43. Gerçek Zamanlı Birden Çok İnsanın Algılanması 2. Durum

Son olarak, sistemin enerji verimliliğini artırmak amacıyla, belirli bir süre boyunca insan algılanmazsa, klimanın enerji tasarrufu moduna geçmesi için bir komut gönderilmektedir. Bu süre zarfında, algılanan insan hareketleri dikkatlice izlenir ve herhangi bir varlık tespiti yapılmazsa, enerji tasarrufu modu devreye girer. Bu şekilde, odada insan bulunmadığı durumlarda enerji tüketimi minimize edilerek çevresel etki azaltılmaktadır.

Enerji tasarrufu modunun devreye girmesi, sistemin kullanıcı dostu olmasını sağlamaktadır. Kullanıcılar, klima sistemini manuel olarak kapatmayı unutsa bile, sistem otomatik olarak enerji tasarrufu moduna geçerek gereksiz enerji tüketimini önlemektedir. Bu özellik, özellikle büyük alanlarda veya ticari ortamlarda enerji yönetimini kolaylaştırır ve daha sürdürülebilir bir kullanım sunar. Enerji tasarrufu modunun devrede olduğu süre boyunca odada tekrar bir insan algılandığında sistem otomatik olarak normal çalışma moduna geri dönmektedir. Bu süreç, kullanıcı konforunu sürekli olarak sağlamak amacıyla hızlı ve verimli bir şekilde gerçekleştirilir. Örneğin, bir kullanıcı odaya girdiğinde, sistem hızla algılama yapar ve hemen önceden ayarlanan konfor sıcaklığına geri dönmektedir. Bu sayede, kullanıcılar herhangi bir müdahalede bulunmadan istedikleri konfor seviyesine ulaşabilirler.

Bu yaklaşım hem kullanıcı konforunu artırmakta hem de enerji kaynaklarının daha etkin bir şekilde kullanılmasını sağlamaktadır. Enerji tasarrufu modunun devreye girmesi, çevresel sürdürülebilirliğe katkıda bulunurken, karbon ayak izinin azaltılmasına da yardımcı olmaktadır. Bu sistem, modern yaşamın getirdiği enerji tüketim sorunlarına yenilikçi bir çözüm sunarak çevresel farkındalığı artırmaktadır.

Bu adımın tamamlanmasıyla yapay zekalı klima sisteminin algoritması sonuçlanmıştır. Yapay zeka algoritması, enerji verimliliği ve kullanıcı konforunu ön planda tutarak optimize edilmiştir. Sistem, her aşamada kullanıcı ihtiyaçlarını ve çevresel etkileri dikkate alarak tasarlanmıştır. Şekil 3.44'te geliştirilen yapay zeka destekli insan algılama sisteminin akış şeması yer almaktadır. Bu şema, sistemin işleyişini ve enerji tasarrufu moduna geçiş sürecini detaylı bir şekilde açıklamaktadır.



Şekil 3. 44. İnsan Algılama Sistemi Akış Şeması

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Geliştirdiğim yapay zekalı klima sistemi, modern yaşamın vazgeçilmez bir parçası olan iklim kontrolünde yenilikçi bir adımı temsil etmektedir. Bu sistem, kullanıcıların konforunu artırmak ve enerji verimliliğini optimize etmek amacıyla tasarlanmıştır. Yüz tanıma teknolojisi sayesinde kişiselleştirilmiş kullanıcı deneyimleri sunarken, insan algılama sistemi sayesinde de algılanan insanların konumu belirlenerek hava akışı optimize edilebilmektedir. Bu bölümünde, geliştirilen yapay zekalı kontrol sisteminin farklı test koşullarında performansı değerlendirilmektedir. Ayrıca geliştirilen sistemin çeşitli senaryolarda nasıl tepki verdiği, kullanım esnekliği ve etkinliği gibi kritik unsurlar analiz edilmektedir. Elde edilen bulgular, yapay zeka destekli klima sistemlerinin gelecekteki iklim kontrol teknolojilerindeki rolünü anlamamıza ve geliştirmemize yardımcı olmaktadır.

4.1. İç Ortam Sıcaklık Ölçümlerinin Doğrulanması

Geliştirilen sistemde iç ortam sıcaklığının ölçülebilmesi için NTC termistör kullanılmıştır. Bu komponentin sıcaklıkla olan ilişkisini karakterize etmek için β (B) parametresi denklemi kullanılmıştır. Bu formül ile termistörün direnç değişimlerinden gerçek sıcaklık verisi elde edilmektedir. Böylece, NTC termistörün direnç değerlerinden elde edilen veriler, sıcaklık değerlerine dönüştürülerek sistem tarafından kullanılabilir hale getirilmiştir.

Kullanılan NTC termistör ile ölçülen ortam sıcaklık verileri, UART protokolü aracılığıyla Thonny programının seri ekranı üzerinden okunabilmektedir. Bu yöntem, sıcaklık verilerinin anlık olarak izlenmesine olanak tanıyarak sistemin performansının sürekli olarak değerlendirilmesini sağlamaktadır. Şekil 4.1’de, seri ekran üzerinden okunan sıcaklık verileri yer almaktadır. Seri ekran üzerinden elde edilen sıcaklık verileri, komponentin ölçtüğü sıcaklığın değerlendirilmesinde önemli bir araç görevi görmektedir ve gerekli ayarlamaların uygulanmasını sağlamaktadır.

```
Shell x
Sıcaklık: 22,01°C
Sıcaklık: 22,05°C
Sıcaklık: 22,03°C
Sıcaklık: 22,05°C
Sıcaklık: 22,04°C
Sıcaklık: 22,02°C
Sıcaklık: 22,07°C
Sıcaklık: 22,01°C
Sıcaklık: 22,05°C
Sıcaklık: 22,01°C
Sıcaklık: 22,04°C
Local Python 3 • /usr/bin/python3
```

Şekil 4. 1. Seri Ekran Üzerinden Okunan Sıcaklık Değerleri

NTC termistör ile ölçülen sıcaklık verileri, farklı ortam sıcaklıklarına sahip odalarda test edilmiş ve ölçümlerin doğruluğu değerlendirilmiştir. Testler, termistörün çeşitli sıcaklık koşullarında nasıl performans gösterdiğini ve ölçtüğü sıcaklıkların doğruluğunu belirlemek için yapılmıştır. Odalardaki sıcaklık değişimleri, geniş bir aralıkta test edilmiş ve her bir test senaryosunda sistem tarafından ölçülen sıcaklık verileri kaydedilmiştir. Tablo 4.1’de kullanılan NTC termistörün farklı ortam sıcaklıklarında gerçekleştirilene testlerin ölçüm sonuçları yer almaktadır.

Tablo 4. 1. İç Ortam Sıcaklık Ölçüm Testleri

Test Edilen İç Ortam Sıcaklığı	NTC Üzerinden Ölçülen Sıcaklık
18,5°C	18,79 °C
22,5 °C	22,82 °C
27,2 °C	27,55 °C
36,0°C	36,43 °C
43,2°C	43,47 °C

NTC testleri, kalibreli endüstriyel bir termometre cihazı referans alınarak gerçekleştirilmiştir. Tablo 4.1’de görüldüğü gibi NTC sensörü üzerinden elde edilen sıcaklık değerleri ile endüstriyel referans cihazın gösterdiği değerler arasında yaklaşık olarak 0,3 - 0,4 °C arasında bir farklılık belirlenmiştir. Bu farklılık, NTC sensörünün son kullanıcı odaklı bir cihazda kullanılacağı düşünüldüğünde kabul edilebilir bir düzeydedir. Ölçülen sıcaklık değerlerindeki bu tür küçük farklılıkların geliştirilen sistemin işlevselliği ve kullanılabilirliği üzerinde minimal bir etkisi olması beklenmektedir.

Test sonuçları, NTC üzerinden sıcaklık ölçümü yapabilmek için geliştirilen algoritmanın doğruluğunu kanıtlamaktadır. Ölçülen sıcaklık verileri geliştirilen sisteme geri bildirim olarak iletilerek, sistemin iç ortam sıcaklığı ile ilgili doğru analizler yapmasını sağlamaktadır. Sonuç olarak yapılan testler, algoritmanın etkinliğini ve sistemin iç ortam sıcaklığıyla ilgili doğru tahminler yapma yeteneğini desteklemektedir.

4.2. Gerçek Zamanlı Yüz Tanıma Sistemi Testleri

Geliştirilen gerçek zamanlı yüz tanıma sisteminde, Raspberry Pi platformu üzerinde OpenCv görüntü işleme kütüphanesi ve Haar Cascade makine öğrenimi algoritması kullanılarak tasarlanmıştır. Öncelikle tanınması istenen yüzlerden oluşan bir veri seti oluşturulmuştur. Veri seti, farklı poz ve açılardan çekilmiş yüz görüntülerinden oluşmaktadır. Daha sonra oluşturulan veri setlerine istenilen ortam sıcaklık bilgisi ve öncelik numarası eklenerek veri tabanına kaydedilmektedir. Veri tabanına kaydedilen veri setlerinin gösterimi Şekil 4.2’de yer almaktadır.



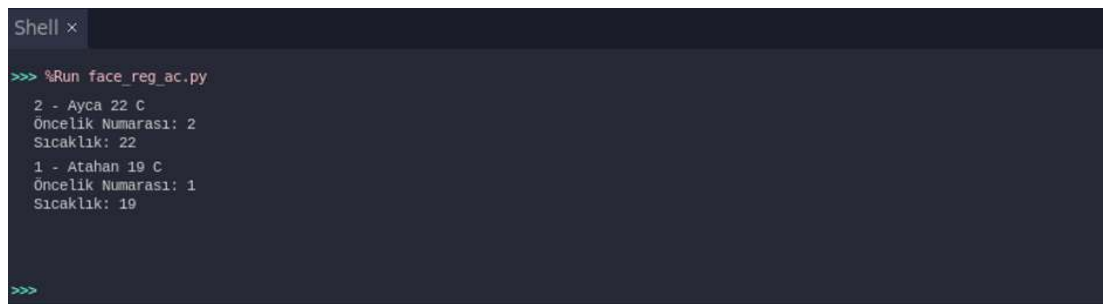
Şekil 4. 2. Veri Tabanına Kaydedilen Veri Setlerinin Formatı

Tanınması istenilen kişilerin veri seti klasörleri Şekil 4.2’de gösterilen formattaki gibi oluşturulmalıdır. Çünkü sistem, tanınan kişinin öncelik numarasını, ismini ve istediği ortam sıcaklığının bilgisini bu klasörün ismini yorumlayarak elde etmektedir. Daha sonra veri tabanına kaydedilen veri setleri kullanılarak sistemin eğitimi gerçekleştirilmektedir. Eğitim aşamasında, algoritma tanınması istenen yüzlerin özelliklerini, belirlenen ortam sıcaklık bilgilerini ve kişilerin öncelik numarasını öğrenmektedir. Test aşamasında, sistem tanımlama yapması gereken yüzlerin görüntülerini alır ve eğitim sırasında öğrendiği özelliklerle karşılaştırarak

yüzleri tanımlamaya çalışır. Bu süreçte, sistemin doğru tanıma yapma yeteneği ölçülmüş ve sistem gerekli iyileştirmeler yapılınca kadar tekrar tekrar test edilmiştir.

İlk aşamada, sistem için 10 farklı görüntüden oluşan bir veri seti kullanılmıştır. Bu veri seti, aynı ışık açısından çekilmiş olup, farklı pozlardaki yüzleri içermektedir. Gerçekleştirilen testlerde, sistem için doğruluk oranı yaklaşık olarak %78 civarında ölçülmüştür. Bu düşük başarı oranının ana nedeni, farklı ışık açılarının sistemin yüzleri algılamasını ve tanımlamasını olumsuz etkilemesidir. Özellikle, değişen ışık koşulları altında, kamera karşısındaki yüzlerin tanınması zorluklarla karşılaşmaktadır. Bu durum, sistemin performansını önemli ölçüde etkilemektedir ve doğruluk oranını düşürmektedir.

Sistemin düşük doğruluk oranına rağmen geri bildirim verme süresi oldukça kısadır. Geri bildirim verme hızını gerçek zamanlı olarak kontrol etmek için iki adet LED kullanılmıştır. Algılanan yüz eğer tanınıyorsa yeşil LED yanarken, tanınmıyorsa kırmızı LED yanacak şekilde bir test algoritması tasarlanmıştır. Ayrıca algoritmaya, tanınan yüzlerin öncelik numaraları ve istedikleri ortam sıcaklık bilgilerinin seri ekran üzerinden okunabilmesi için gerekli eklemeler de yapılmıştır. Yapılan testlerde, FPS değeri ortalama 4,32 olarak ölçülmüştür. Fakat FPS değeri düşük olmasına rağmen, ekranda bir yüz algılanıp, tanınması için beklenirken, sistem çok daha önce yeşil veya kırmızı LED'i yakarak yüzün tanınıp tanınmadığı hakkında geri bildirim verebilmektedir. Bunun yanı sıra tanınan yüzden anlamlı verilerin seri ekran üzerinden gözlemlenebilmesi de çok daha kısa sürede gerçekleşmektedir. Şekil 4.3'te tanımlanan yüzlerden elde edilen anlamlı bilgilerin seri ekranda gösterilmesi yer almaktadır.



```
Shell x
>>> %Run face_reg_ac.py
2 - Ayca 22 C
Öncelik Numarası: 2
Sıcaklık: 22
1 - Atahan 19 C
Öncelik Numarası: 1
Sıcaklık: 19
>>>
```

Şekil 4. 3. Anlamlı Bilgilerin Seri Ekranda Gösterilmesi

İkinci aşamada; sistem, farklı ışık açıları ve farklı pozisyonlarda 100 farklı görüntüden oluşan bir veri seti kullanılarak eğitilmiştir. Yapılan testler sonucunda, sistemin kamera karşısındaki yüzleri doğru algılama ve tanıma oranı yaklaşık %97 olarak ölçülmüştür. Ancak, bu artan doğruluk oranı, geri bildirim verme süresinde önemli bir artışa neden olmuştur. Gerçekleştirilen testlerde FPS değeri ortalama 1,25 olarak ölçülmüştür. Tasarlanan geri bildirim test algoritması ile sistem test edildiğinde LED'lerin çok geç bir sürede yanarak geri bildirim verdiği gözlemlenmiştir. Sistemin, yüz tanıma doğruluğunun artması, geri bildirim süresinin uzamasına ve dolayısıyla sistem performansının olumsuz etkilenmesine neden olmaktadır. Bu durum, geri bildirim hızı ile yüz tanıma doğruluğu arasında bir denge kurulması gerektiğini ortaya koymaktadır. Bu dengenin kurulabilmesi için sistemin optimize edilmesi gerekmektedir.

Üçüncü aşamada; sistem, farklı ışık açıları ve farklı pozisyonlarda 50 farklı görüntüden oluşan bir veri seti kullanılarak eğitilmiştir. Yapılan testler sonucunda, sistemin kamera karşısındaki yüzleri doğru algılama ve tanıma oranı yaklaşık %90 olarak ölçülmüştür. Gerçekleştirilen testlerde FPS değeri ortalama 3,27 olarak ölçülmüştür. Test algoritması ile sistem test edildiğinde geri bildirim verme süresinin kabul edilebilir bir sürede olduğu gözlemlenmiştir.

Her üç aşamada da yüz tanıma işlemi için kamera ile yüz arasında 30 cm mesafe bırakılmıştır. Ancak, yapılan deneyler sonucunda, kameradan uzaklaştıkça yüzün algılanıp tanınma oranının düştüğü gözlemlenmiştir. Bu durum, sistem performansının mesafe faktörüne duyarlı olduğunu ve belirlenen mesafenin önemli bir parametre olduğunu ortaya koymaktadır. Gerçekleştirilen testler, yüz tanıma sisteminin çeşitli koşullar altındaki performansını kapsamlı bir şekilde değerlendirmek için kritik öneme sahiptir. Elde edilen veriler, sistemin iyileştirilmesi ve daha geniş bir uygulama yelpazesinde kullanılabilirliğinin artırılması için yol gösterici olacaktır. Bu veriler arasında doğruluk oranı, geri bildirim süresi, FPS süresi ve algılama mesafesi gibi önemli parametreler bulunmaktadır. Tablo 4.2'de bu testler sonucunda elde edilen veriler yer almaktadır.

Tablo 4. 2. Gerçekleştirilen Test Sonucunda Elde Edilen Parametreler

Aşama	Eğitilen Görüntü Sayısı	Doğruluk Oranı (%)	Geri Bildirim Süresi (s)	FPS (s)	Algılama Mesafesi (cm)
1	10	78	<1	4,32	30
2	100	97	4	1,25	30
3	50	90	1-2	3,27	30

Farklı test senaryoları üzerinde yapılan detaylı değerlendirmeler neticesinde, yüz tanıma sisteminin eğitim aşamasında 50 farklı görüntü içeren bir veri setinin kullanılmasına karar verilmiştir. Bu seçim, sistemin doğruluk oranı ile geri bildirim süresi arasında bir denge kurulmasını sağlamaktadır. Geliştirilen sistem, kamera tarafından yakalanan görüntülerdeki yüzlerin ekrana yansıtılması yerine, tanıma işlemi sonucunda kullanıcılara geri bildirim sağlamak amacıyla LED ışıklar kullanmaktadır. Bundan dolayı sistemdeki önemli öncelikler, yüz tanıma doğruluk oranı ve geri bildirim süresi olup, FPS değeri daha az önem arz etmektedir. Yüz tanıma doğruluk oranı ve geri bildirim süresi, sistem performansını belirleyen temel parametrelerdir. Bu nedenle, sistem tasarımında bu parametrelerin optimize edilmesi, kullanıcı deneyimini artırmak için kritik öneme sahiptir.

Yüksek bir doğruluk oranı, sistemin güvenilirliğini ve yüz tanıma yeteneğini arttırırken, aynı zamanda geri bildirim süresinin kısa olması da kullanıcı deneyimini iyileştirir ve sistemin etkinliğini artırmaktadır. Bu dengeli yaklaşım, sistemin en uygun performans seviyesinde çalışmasını sağlayarak, kullanıcı taleplerine hızlı ve etkin bir şekilde yanıt vermesini sağlamaktadır.

Geliştirilen yapay zeka destekli yüz tanıma sisteminin klima gibi iç mekân uygulamalarında kullanılması, ev içi mahremiyet ve güvenlik kaygılarını öne çıkarmaktadır. Bu sistem, kamera aracılığıyla ev içindeki kullanıcıların görüntülerini işleyerek çalıştığı için kullanıcıların gizliliği önemli bir endişe kaynağıdır. Bu bağlamda, sistem tasarımı sırasında güvenlikle ilgili dezavantajları aşmak için dikkatli adımlar atılmıştır. Öncelikle, kullanıcıların mahremiyetini korumak ve veri güvenliğini sağlamak amacıyla, kullanıcıların görüntülerinden oluşan veri setleri bulut

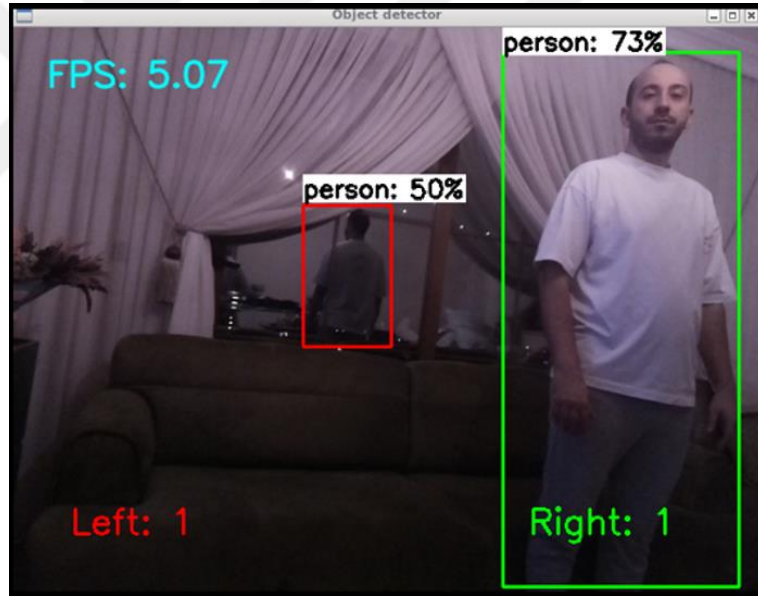
tabanlı bir yapı yerine donanımsal bir kaynak olan Raspberry Pi gibi yerel bir platformda depolanmaktadır. Bu yaklaşım, kullanıcıların ev içi aktivitelerine dair hassas verilerin buluta yüklenmesiyle ilişkili güvenlik risklerini en aza indirirken, veri güvenliğini ve mahremiyeti artırmayı amaçlamaktadır.

Donanımsal depolama, sistemin internet bağlantısına ihtiyaç duymadan çalışabilmesini sağlamakta ve böylece dışarıdan erişim ve veri sızıntısı risklerini azaltmaktadır. Bu tasarım yaklaşımı, kişisel verilerin güvenliğini ve mahremiyetini koruyarak, kullanıcıların güvenlik endişelerini gidermeyi ve sistemin kabul edilebilirlik düzeyini artırmayı amaçlamaktadır. Yerel depolama, bulut tabanlı alternatiflere göre daha güvenli bir çözüm sunmakta ve kullanıcıların güvenliği ve mahremiyeti üzerinde daha fazla kontrol sahibi olmalarını sağlamaktadır. Bu şekilde, geliştirilen yapay zeka destekli sistemin güvenilir ve kullanıcı dostu olması sağlanırken, kullanıcıların veri güvenliği ve mahremiyeti de gözetilmiş olmaktadır.

4.3. Gerçek Zamanlı İnsan Algılama Sistemi Testleri

Geliştirilen yapay zekalı insan algılama sistemi, Raspberry Pi platformu üzerinde TensorFlow makine öğrenimi kütüphanesi ve YOLOv4 Tiny yapay zeka modeli kullanılarak oluşturulmuştur. Bu sistem, insanların algılanmasını sağlayarak klima gibi iç mekân sistemlerini kişiselleştirmeyi amaçlamaktadır. Algılanan insanın konumuna göre, hava akışı yönlendirilerek konfor seviyesi artırılmaktadır. Sistem, geniş bir nesne sınıflandırma veri seti olan COCO ile eğitilmiş YOLOv4 Tiny modelini temel almaktadır. Ancak, sistem sadece insan algılama amacıyla tasarlandığı için COCO veri setinde yer alan nesnelerden sadece insan sınıfı kullanılmaktadır. Bu sayede, sistemin doğruluğu insan algılamasına odaklanarak optimize edilmiş ve hedeflenen uygulama senaryolarına daha uygun hale getirilmiştir. Yapay zeka tabanlı bu sistem, çeşitli uygulama senaryolarında insanların konumunu hassas bir şekilde tespit edebilmekte ve buna bağlı olarak hava akışını yönlendirerek kullanıcı konforunu artırmaktadır. Bu teknolojik çözüm hem enerji tasarrufu sağlayarak hem de kullanıcı deneyimini iyileştirerek iç mekân yaşamını daha etkili ve rahat hale getirmeyi hedeflemektedir.

Geliştirilen yapay zeka destekli insan algılama sistemlerinin değerlendirilmesinde, elde edilen yaklaşık %92'lik doğruluk oranı önemli bir ölçüttür. Ancak, bu doğruluk oranını etkileyen çeşitli faktörler bulunmaktadır. Özellikle, sistemin yanlış algılama yapma eğilimi, doğruluk oranının azalmasına yol açmaktadır. Bu bağlamda, yapay zeka sisteminin yanlış algılama yapma durumlarından biri, yansıma görüntülerinin gerçek insan varlıkları olarak algılanmasıdır. Örneğin, cam yüzeyler veya aynalardan yansıyan görüntüler, algoritma tarafından insan varlığı olarak yorumlanabilmektedir. Bu yanlış algılama durumu, özellikle odadaki toplam insan sayısının doğru bir şekilde hesaplanmasını zorlaştırmaktadır. Çünkü yansıma görüntüleri, gerçek insan varlıklarıyla karıştırılarak sistemin toplam insan sayısını yanlış bir şekilde tahmin etmesine neden olmaktadır. Şekil 4.4'te sistemin yanlış algılama yapma durumu yer almaktadır.

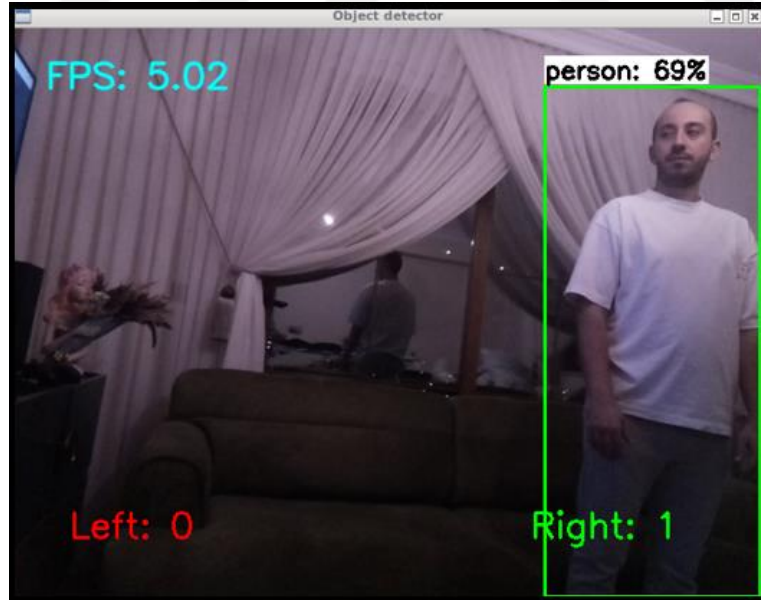


Şekil 4. 4. Yansıyan Görüntülerin Algılanması

Yansıma görüntülerinden kaynaklanan hataların azaltılması ve doğruluk oranının artırılması amacıyla, algoritmaların insan varlığını belirleme sınır değerinin optimal olarak ayarlanması için kapsamlı bir dizi test gerçekleştirilmiştir. Bu testlerin sonucunda, geliştirilen sistemde ilk olarak nesne algılamak için kullanılan sınır değeri 0,4 olarak belirlenmiştir ve üst değer 1 olarak kabul edilmiştir. Sistem, bir nesneyi algıladığında bir güven puanı hesaplar ve eğer güven puanı 0,4 ile 1 arasında ise bunun

kesin nesne olduğunu yorumlayıp algılanan nesne için uygun etiketi bulmak amacı ile veri setinde yer alan görüntüler ile karşılaştırmaktadır. Bu yaklaşım, algılanan insanın güven puanının %40'ın üzerinde olması durumunda sistem tarafından algılanan nesnenin kesin bir insan olduğuna karar verilmesini sağlamaktadır. Ancak, bu sınır değeri aynı zamanda yansıma görüntülerini de insan olarak algılayabilmekte ve yanlış geri bildirimlere neden olabilmektedir. Bu nedenle, yapılan testlerin ardından sınır değerinin 0,58 olarak ayarlanmasına karar verilmiştir.

Belirlenen sınır değeri, daha kesin bir insan algılama yeteneği sağlamış ve yansıma görüntülerinden kaynaklanan hataları minimuma indirmiştir. %58 güven puanı eşiğinin belirlenmesi, algılanan nesnelerin gerçek insan varlıkları olma olasılığını artırırken, yanlış algılamaları azaltmıştır. Bu sayede, sistemin doğruluk oranı ve güvenilirliği önemli ölçüde artırılmıştır. Sınır değerinin 0,58 olarak ayarlandıktan sonra yapılan yansıma görüntüsü testinde elde edilen görsel Şekil 4.5'te yer almaktadır.



Şekil 4. 5. Yansıyan Görüntülerin Algılanmaması

İnsanları algılamak için belirlenen sınır değerinin 0,6 olarak ayarlanması, sistemdeki yanlış algılamaların daha da azalmasına neden olmuştur. Ancak, bu değişiklik bazı durumlarda sistemin algılaması gereken insanları da algılayamamasına

yol açmıştır. Bu durum, daha yüksek bir sınır değerinin sistemin hassasiyetini artırırken aynı zamanda gerçek nesnelerin tespit edilme oranını azalttığını göstermektedir. Bundan dolayı, yapılan testlerin sonuçları değerlendirildiğinde, 0,58 sınır değerinin geliştirilen sistem için optimal değer olduğu doğrulanmıştır. Testler, kamera tarafından anlık olarak farklı ışık açılarında elde edilen insan ve yansıma görüntülerini içeren canlı görüntüler üzerinde gerçekleştirilmiştir. Bu testler, her bir sınır değeri için toplamda 50 kez tekrarlanmıştır ve her testte kamera ile insan arasındaki mesafe sabit tutulmuştur. Gerçekleştirilen sınır değeri belirleme testlerinin sonuçları Tablo 4.3'te yer almaktadır.

Tablo 4. 3. Sınır Değeri Belirleme Testi Sonuçları

Sınır Değeri	Doğru Algılanan İnsan Sayısı	Yanlış Algılanan İnsan Sayısı	Genel Doğru Algılama Oranı	Algılama Mesafesi (cm)
0,4	50	23	%54	200
0,45	50	21	%58	200
0,5	50	17	%66	200
0,55	49	11	%76	200
0,58	49	3	%92	200
0,59	46	3	%86	200
0,6	40	2	%76	200

Sınır değeri optimize edildikten sonra sistemin aynı anda farklı sayıda insan algılanması durumunda çalışması test edilmiştir. Toplamda 6 insan ile gerçekleştirilen testlerde farklı kombinasyonlar ile insanlar ekranın sağında ve solunda konumlandırılmıştır. Daha sonra sistemin algoritması test edilmiştir. Ekranın hangi bölgesindeki insan sayısı daha fazla ise sistem 1 dakika boyunca bunu doğrulamış daha sonra step motoru uygun yöne hareket ettirmiştir. Geliştirilen sistem, sürekli olarak insan takibini gerçekleştirebilme özelliğine sahip olduğu için insanların konumları değiştiğinde bile ekranın sağ ve sol tarafındaki insan popülasyon oranı sürekli olarak hesaplanmakta ve sistem, anlık olarak gerekli aksiyonları alarak yatay kanatları uygun şekilde ayarlayabilmektedir.

Akademik arařtırmalar ve piyasadaki ürünler incelendiğinde, insan algılama amacıyla farklı teknolojilerin kullanıldığını gözlemlenmiştir. Bu teknolojiler arasında yaygın olarak PIR sensörleri, radar ve lidar sensörleri bulunmaktadır. Ancak, bu sensörlerin her biri belirli dezavantajlarla karşı karşıyadır. PIR sensörleri, özellikle hareketsiz insanları algılamada zorluklar yaşayabilir ve evcil hayvanların varlığında yanıltıcı sonuçlara sebep olabilmektedir. Ayrıca, ortamdaki insan popülasyonunu doğru bir şekilde ölçme konusunda sınırlıdır.

Lidar ve radar sensörleri, nesnelerin konumunu ve popülasyonunu ölçebilir, dolayısıyla hareketsiz insanları da algılayabilirler. Ancak, hareketsiz nesneler arasında doğrudan ayırım yapamazlar. Ayrıca, bu sensörler sadece insanları değil, evcil hayvanlar gibi diğer hareketli nesneleri de algılayabilir ve bu durum yanlış sonuçlara yol açabilir. Karmaşık ortamlarda, lidar ve radar sensörlerinin doğruluğu azalabilir ve insanları diğer nesnelerden ayırt etmek zorlaşır. Bu nedenle, bu sensörlerle sadece insanları algılayacak bir filtreleme yapmak zordur. Her iki sensör türü de belirli koşullarda etkili olsa da, çeşitli hareketli nesnelerin bulunduğu ortamlarda yanıltıcı sonuçlar verebilir. Tablo 4.4'te, nesne algılama için kullanılan çeşitli sensörlerin ve geliştirilen yapay zeka destekli kamera sisteminin çalışma performansı belirli durumlar için karşılaştırılmıştır.

Tablo 4. 4. Farklı Sensörler ile Yapay Zekalı Kameranın Karşılaştırılması

Durumlar	PIR Sensör	LIDAR Sensörü	Radar Sensörü	Yapay Zeka Destekli Kamera
Sadece İnsanların Algılanması	✗	✗	✗	✓
Hareketsiz İnsanların Algılanması	✗	✗	✗	✓
Ortamdaki İnsanların Sayısının Hesaplanması	✗	✓	✓	✓
İnsanların Konum Tespiti	✗	✓	✓	✓
Mahremiyet Koruması	✓	✓	✓	✗

Tablo 4.4'te sunulduđu üzere, yapay zeka destekli kamera sistemi sadece insan algılama, popölasyon ve konum belirleme konularında diğör sensör teknolojilerine göre daha üstündür. Ancak, kameralı sistemlerin mahremiyetle ilgili bazı dezavantajları bulunmaktadır. Bu dezavantajı ortadan kaldırmak adına geliştirilen sistemde, bulut tabanlı bir veri tabanı yerine donanımsal bir veri tabanı kullanılmıştır. Yüz tanıma sisteminde kullanılan bu çözüm, insan algılama sisteminde de uygulanmıştır. Sistemde insan algılamak için kullanılan COCO veri seti, Raspberry Pi üzerine takılı SD kart içerisine kaydedilmiştir. Bu sayede sistem, nesne algılama işlemi sırasında doğrudan bu kaydedilen veri setini kullanarak herhangi bir internet bağlantısına ihtiyaç duymamaktadır. Donanımsal veri tabanı kullanımıyla güvenlik önlemleri artırılmış ve mahremiyet kaygıları azaltılmıştır. Böylece sistem daha güvenli hale getirilmiştir.

5. SONUÇ VE ÖNERİLER

Yapay zeka destekli sistemler, günümüzde endüstriyel alanlarda üretim ve kalite süreçlerinin iyileştirilmesi için yaygın olarak kullanılmaktadır. Bu sistemler, veri analizi, önleyici bakım ve otomatikleştirme gibi çeşitli uygulamalarda kullanılarak işletmelerin verimliliğini artırmakta ve maliyetleri azaltmaktadır. Geliştirilen klima sistemi, bu teknolojinin son kullanıcıya yönelik uygulanmasının bir örneğidir. Yapay zeka algoritmaları, klimanın işlevselliğini optimize etmek, kullanıcı deneyimini artırmak ve enerji tüketimini düzenlemek için kullanılmaktadır. Bu sayede, kullanıcılara daha konforlu bir ortam sağlarken aynı zamanda enerji tasarrufu da sağlanmaktadır. Bu entegrasyon, son kullanıcıların gereksinimlerine daha duyarlı bir ürün ortaya koymayı amaçlamaktadır. Sonuç olarak, yapay zeka destekli klima sistemi, kullanıcıların taleplerini karşılamak ve modern teknolojinin avantajlarından yararlanmak için etkili bir çözüm sunmaktadır.

Geliştirilen sistemde, yüz tanıma ve insan algılama uygulamaları için iki farklı yapay zeka algoritmaları tercih edilmiştir. Bu tercihin altında yatan temel neden, her bir algoritmanın belirli kullanım senaryolarında daha etkili olmasıdır. Yüz tanıma uygulaması için Haar Cascade algoritması kullanılmıştır. Çünkü bu algoritma yüz tanıma görevlerinde yüksek doğruluk seviyelerine ulaşırken, işlemci kaynaklarını minimal seviyede kullanmaktadır. Haar Cascade, yüzlerin belirli özelliklerini tanımlamak için önceden eğitilmiş bir model kullanmaktadır. Bu nedenle özellikle yüz tanıma için idealdir. Diğer yandan, insan algılama sistemi için YOLO yapay zeka algoritması tercih edilmiştir. YOLO algoritması, bir görüntüdeki nesneleri tek bir işlemde tespit edebilme özelliğine sahiptir ve bu da insan algılama gibi genel nesne tespiti görevlerinde oldukça etkilidir. YOLO, görüntüyü birden fazla ölçekte işleyerek nesneleri tespit eder ve sınıflandırabilir. YOLO'nun çoklu ölçek işleme yeteneği, farklı boyutlarda ve konumlarda bulunan nesnelerin doğru ve hızlı bir şekilde tespit edilmesini sağlar. Bu nedenle, insan algılama gibi genel nesne tespiti görevleri için YOLO algoritması daha uygun bir seçenektir. Bu şekilde, her bir algoritmanın güçlü yönlerinden faydalanarak hem yüz tanıma hem de insan algılama işlemleri daha etkili bir şekilde gerçekleştirilebilmektedir.

Yapılan akademik arařtırmalar, yapay zeka destekli yüz tanıma sistemlerinin özellikle güvenli erişim uygulamalarında yaygın olarak kullanıldığını göstermektedir. Geliştirilen klima sisteminde ise yüz tanıma özelliđi, kullanıcılar için benzersiz bir kişiselleştirme aracı olarak öne çıkmaktadır. Bu sistem, her kullanıcının tercihlerine göre klima ayarlarını otomatik olarak tanımlayarak, konforlarını en üst düzeye çıkarmak için çalışmaktadır. Örneđin, bir kullanıcı serin bir ortamı tercih ederken, diđer bir kullanıcı daha sıcak bir ortamı tercih edebilir. Yüz tanıma sistemi, bu farklı tercihleri algılayarak her kullanıcıya özel bir iklimlendirme deneyimi sunmaktadır. Bu kişiselleştirme özelliđi, kullanıcıların klima kontrolünü daha kolay ve etkili hale getirirken, aynı zamanda enerji tasarrufu sağlayarak sürdürülebilir bir yaşam tarzına katkıda bulunmaktadır.

Tez kapsamında geliştirilen bir diđer yapay zekalı sistem olan insan algılama sistemi, odadaki insanların sayısını ve konumunu hassas bir şekilde belirleyerek klima performansını optimize etmektedir. Bu sistem, odadaki insan sayısını algılayarak klimanın hava akışını otomatik olarak ayarlar ve konumlarına göre dikey kanatları hareket ettirir. Örneđin, odanın bir tarafında yoğun bir insan varlığı algılandığında, insanların bulunduğu konumun tersine doğru dikey kanatları ayarlar. Böylece, hava akışı doğrudan insanların üzerine gelmez ve daha konforlu bir iklimlendirme sağlanır. Ayrıca, belirli bir süre boyunca odada insan algılanmadığında, klima otomatik olarak enerji tasarruf moduna geçer. Böylece kullanıcıların enerji maliyetlerini düşürür. Bu özellikler, insan algılama sisteminin kullanıcıların konforunu artırmasının yanı sıra enerji verimliliğini de optimize ederek çevre dostu bir yaşam tarzına katkı sağlamaktadır.

Geliştirilen yapay zeka destekli yüz tanıma ve insan algılama sistemlerinin testleri, sınırlı işlem gücüne sahip Raspberry Pi ve standart bir kamera kullanılarak gerçekleştirilmiştir. Yapılan testlerin sonucunda yüz tanıma sisteminde %90, insan algılama sisteminde ise %92 doğruluk oranı elde edilmiştir. Elde edilen sonuçlar, geliştirilen sistemin, düşük maliyetli ve yaygın olarak kullanılan donanım ve ekipmanlarla bile tatmin edici performans gösterebildiğini ortaya koymaktadır. Bu durum geliştirilen sistemin, yüksek çözünürlüklü kamera veya yüksek işlem gücü

kapasitesine sahip donanıma olan ihtiyacının olmaması, son kullanıcı odaklı uygulamalara sorunsuz bir şekilde entegre edilebileceğini göstermektedir. Bu doğrultuda geliştirilen sistem, kullanıcı dostu ve ekonomik bir çözüm sunarak, geniş bir kullanım alanına sahip olabilir.

Geliştirilen sistemde kullanıcı mahremiyeti düşünüldüğü için donanımsal bir veri tabanı kullanılarak internetten bağımsız çalışması sağlanmıştır. Gelecekte, bulut tabanlı bir veri tabanının entegrasyonu ile birlikte, sistem daha geniş bir veri analizi ve özelleştirme potansiyeline sahip olabilir. Bulut tabanlı yapı, geniş ölçekte veri analizi yapma ve daha kapsamlı kişiselleştirme imkânı sunarak kullanıcıların klima deneyimini optimize etme potansiyelini artırabilir.

Bu tür geliştirmelerin beraberinde getirdiği önemli bir husus da veri güvenliğinin sağlanmasıdır. Veri güvenliğini sağlamak için AES-256 şifreleme algoritması ve HTTPS protokolü gibi güçlü şifreleme yöntemleri ve güvenlik protokolleri benimsenebilir. Özellikle yüz tanıma sisteminde, kullanıcılar öncelikle kendi görüntüleri ile bir veri seti oluşturmakta ve bu veri seti kaydedilmektedir. Bu süreçte, verilerin bulut tabanlı veri tabanına güvenli bir şekilde aktarılması ve saklanması kritik önem taşımaktadır. Kullanıcı görüntüleri, AES-256 algoritması ile şifrelenerek bulut veri tabanında saklanabilir ve sadece yetkili kullanıcılar tarafından erişilebilir. Aynı zamanda, sistem çalışırken hem yüz tanıma sisteminde hem de insan algılama sisteminde sürekli olarak bir görüntü alınıp veri seti ile karşılaştırma yapılmaktadır. Bu süreçlerde, gerçek zamanlı veri alışverişi için HTTPS protokolü kullanılabilir ve veriler anında AES-256 ile şifrelenerek veri güvenliği sağlanabilir.

Ayrıca, kimlik doğrulama ve yetkilendirme mekanizmalarının güçlendirilmesi de gerekmektedir. İki faktörlü kimlik doğrulama gibi güvenlik önlemleri kullanılarak yalnızca yetkili kişilerin veri tabanına erişimi sağlanmalıdır. Düzenli güvenlik güncellemeleri ve yama yönetimi ile verilerin bütünlüğü ve gizliliği korunmalıdır. Bu entegrasyonlar, kullanıcı deneyimini iyileştirirken aynı zamanda veri güvenliğini sağlamak adına önemli adımlar olacaktır.

Sistemin gelecekteki potansiyel gelişim alanlarından biri de insan pozisyonlarını tespit edebilme yeteneğinin eklenmesidir. Örneğin sistem; kullanıcıların ayakta durma, oturma veya yatma gibi farklı pozisyonlarını belirleyebilir. Bu bilgi kullanılarak, kullanıcı yatarken klimanın otomatik olarak uyku moduna alınması gibi ek konfor sağlayan özellikler eklenebilir. Ayrıca, diğer algılanan pozisyonlara göre de çeşitli senaryolar türetilip, sistemin çalışması bu senaryolara göre optimize edilebilir. Bu tür iyileştirmeler, kullanıcı deneyimini ve enerji verimliliğini artırırken, sistemin kullanıcıların ihtiyaçlarına daha duyarlı hale gelmesini sağlayacaktır. Böylece hem kullanıcı konforu artırılabilir hem de enerji tüketimi optimize edilerek sürdürülebilir bir kullanım sağlanabilir.

Günümüzde, iklim kontrolü teknolojilerinin akıllı ve verimli hale getirilmesi hem kullanıcı konforunu artırmak hem de enerji tüketimini optimize etmek için giderek daha önemli hale gelmektedir. Geliştirilen yapay zeka destekli klima sistemi, modern yaşamın vazgeçilmez bir parçası olan iklim kontrolünü daha akıllı ve etkili bir hale getirmektedir. Sistem, kullanıcı tercihlerini dikkate alarak özelleştirilmiş bir deneyim sunmanın yanı sıra enerji verimliliğini optimize ederek enerji tasarrufu sağlamaktadır. Bu tez çalışması, geliştirilen yapay zeka destekli klima sisteminin kullanıcı deneyimini nasıl iyileştirdiğini ve enerji verimliliğini nasıl artırdığını araştırmaktadır. Yapılan testler ve analizler ile elde edilen sonuçlar, bu teknolojinin gelecekteki iklim kontrol sistemlerindeki potansiyel etkisini vurgulamakta ve daha sürdürülebilir bir yaşam için önemli bir adım oluşturmaktadır.

KAYNAKLAR

- [1] Güven, C. Yapay Zekâ Yöntemleri Kullanılarak Akıllı Ev Sistemi Geliştirilmesi. Mersin Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı, Mersin, 2020, 96 s. (Yüksek Lisans Tezi).
- [2] Yazıcı, O. Kullanıcı Davranışlarını Öğrenebilen, Yapay Zekâya Sahip Akıllı Ev Otomasyonu Uygulaması. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Ana Bilim Dalı, Edirne, 2022, 159 s. (Yüksek Lisans Tezi).
- [3] Mennicken, Sarah & Vermeulen, Jo & Huang, Elaine. (2014). From today's augmented houses to tomorrow's smart homes. 105-115. 10.1145/2632048.2636076.
- [4] Alam, M. R., Reaz, M. B. I., Ali, M. A. M. A Review of Smart Homes—Past, Present, and Future. IEEE Transactions on Systems Man and Cybernetics, 2012, 42(6), 1190-1203.
- [5] Gonzales, C. R., Woods, E. R. Digital Image Processing. Pearson Education, New York, 2018, 1022 s.
- [6] Lee, K. Son, M. DeepSpotCloud: Leveraging Cross-Region GPU Spot Instances for Deep Learning. IEEE 10th International Conference on Cloud Computing. 2017, 98-105.
- [7] Daş, R., Polat, B. & Tuna, G. (2019). Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 31 (2), 571-581. DOI: 10.35234/fumbd.608778.
- [8] Kuyumcu, O. Otomotiv Endüstrisinde Araç Parçalarının Görüntü İşleme Yöntemi ile Kalite Kontrol Uygulaması. Gebze Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Mühendisliği Ana Bilim Dalı, Gebze, 2022, 67 s. (Yüksek Lisans Tezi).
- [9] Rushdi, R. I. Face Recognition Using Image Processing And Machine Learning Methods. Atılım University, The Graduate School Of Natural And Applied Sciences, The Department Of Computer Engineering Systems, Ankara, 2020, 99 s. (Master's Thesis).
- [10] Redmon, J., Divvala, S., Girshick R., Farhadi, A., You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 779-788, DOI: 10.1109/CVPR.2016.91.
- [11] Girit, A. Drowsy Driver Detection Using Image Processing. Middle East Technical University, The Graduate School Of Natural And Applied Sciences, The Department Of Electrical And Electronic Engineering, Ankara, 2014, 118 s. (Master's Thesis).
- [12] Thabet, B. A., Amor, B. N. Enhanced Smart Doorbell System Based on Face Recognition. 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2015, 373-377, DOI: 10.1109/STA.2015.7505106.

- [13] Çapşek, F. M., Karacı, A. Derin Öğrenme ve Görüntü İşleme Yöntemlerini Kullanarak Yüz ve Göz Hareketleri ile Bilgisayar Kontrolü. *El-Cezeri Fen ve Mühendislik Dergisi*. 2022, 9(4), 1170-1177.
- [14] Eldem, A., Eldem, H., Palalı, A. Görüntü İşleme Teknikleriyle Yüz Algılama Sistemi Geliştirme. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. 2017, 6(2), 44-48.
- [15] Jacintha, V., Simon J., Tamilarasu S., Thamizhmani R., Thanga Y. K., Nagarajan J. A Review on Facial Emotion Recognition Techniques. 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0517-0521, DOI: 10.1109/ICCSP.2019.8698067.
- [16] Das, A., Ansari, W. M., Basak, R. Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, 1-5, DOI: 10.1109/INDICON49873.2020.9342585.
- [17] Adusumalli, H., Kalyani, D., Sri, K. R., Pratapteja, M., Rao, P. D. R. V. P. Face Mask Detection Using OpenCV. Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, 1304-1309, DOI:10.1109/ICICV50876.2021.9388375.
- [18] Zuo, F., de With, N. H. P., Real-Time Embedded Face Recognition For Smart Home. *IEEE Transactions on Consumer Electronics*. 2005, 51(1), 183-190.
- [19] Vaidya, B., Patel, A., Panchal, A., Mehta, R., Vaghasiya, P. Smart Home Automation With A Unique Door Monitoring System For Old Age People Using Python, Open CV, Android And Raspberry Pi. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, 82-86, DOI: 10.1109/ICCONS.2017.8250582.
- [20] Munir, A., Ehsan, K. S., Raza, M. M. S., Mudassir M., Face and Speech Recognition Based Smart Home. 2019 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 2019, 1-5, DOI: 10.1109/CEET1.2019.8711849.
- [21] Bhanse, K. V., Jaybhave, D. M., Face Detection and Tracking Using Image Processing on Raspberry Pi, 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2018, 1099-1103, DOI: 10.1109/ICIRCA.2018.8597246.
- [22] Sharma, R., Goyal, N., Firoda, R., Suresh, R. M., Kishor, I., Kumar, M., IoT based Smart Mirror Using OpenCV with Voice Assistant, *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2022, 6(6), DOI:10.13140/RG.2.2.11938.40640.
- [23] Kadhim, T. A., Hariri, W., Smaoui Z. N., Aissa, B. D., A Face Recognition Application For Alzheimer's Patients Using ESP32-CAM And Raspberry Pi, *Journal of Real-Time Image Processing*, 2023, 20(5), DOI: 10.1007/s11554-023-01357-w.
- [24] Lin, Z., Li, Y., Design And Implementation Of Classroom Attendance System Based On Video Face Recognition, 2019 International Conference On Intelligent

Transportation, Big Data & Smart City (ICITBS), 2019, 385-388, DOI: 10.1109/ICITBS.2019.00101.

[25] Suchitra, Suja, P., Tripathi, S., Real-Time Emotion Recognition From Facial Images Using Raspberry Pi II, 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2016, 666-670, DOI: 10.1109/SPIN.2016.7566780.

[26] Fernandez, V. E., Pardo, G. H., Jimenez, G. D., Freire, P. L., Built-In Face Recognition For Smart Photo Sharing In Mobile Devices, 2011 IEEE International Conference On Multimedia And Expo, Barcelona, Spain, 2011, 1-4, DOI: 10.1109/ICME.2011.6012057.

[27] Karuppasamy, K., Shruthi, G., Sathish, C., Viswamithran, B., Vinisha, M., Face Detection OpenCV Based ATM Security System, International Journal of Research in Engineering, Science and Management, 2021. 4(8), 78-82.

[28] Nagaraj, P., Sathish, S., Kumar, M. M., Khan, A. A., Mohiuddin, A., Haroon, S. M., Yolo: Human Detection-Based Intelligent Home Automation using IOT, 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, 1-6, DOI: 10.1109/ICCCI56745.2023.10128615.

[29] Shu, Y., Hu, L., A Vision-based Human Posture Detection Approach for Smart Home Applications, International Journal of Advanced Computer Science and Applications (IJACSA), 2023, 14(10), 209 – 216, DOI: 10.14569/IJACSA.2023.0141023.

[30] Wahyutama, A., B., Hwang M., YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins, Electronics, 2022, 11(9):1323, DOI: 10.3390/electronics11091323.

[31] Sharar, L., Abas, H. K., Smart Vision-Based HVAC System, Proceedings of 2019 Electrical Engineering Symposium (EES2019), 2019, 45 – 50.

[32] Huang, Q., Hao, K., Development Of CNN-Based Visual Recognition Air Conditioner For Smart Buildings, Journal of Information Technology In Construction (ITcon), 2020, 361-373 DOI: 10.36680/j.itcon.2020.021.

[33] <https://www.raspberrypi.com/> Erişim Tarihi: 14.03.2024

[34] Süzen, A., A., Duman, B., Şen, B. Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 26-28 Haziran, 2020, Ankara, Türkiye (Bildiriler Kitabı, 1-5).

[35] Jainuddin, A., A., A., Hou, C., Y., Baharuddin, Z., M., Yussof, S. Performance Analysis of Deep Neural Networks for Object Classification with Edge TPU. 8th International Conference on Information Technology and Multimedia (ICIMU), 24-26 Ağustos, 2020, Selangor, Malaysia (Bildiriler Kitabı, 323-328).

[36] Gupta, S., G. Smart Sensors And Sensing Technology. Springer, Berlin, 2008, 411s.

- [37] Litwiller, D. CCD vs. CMOS: Facts and Fiction, *Photonics Spectra*, 2001, 35(1), 154-158.
- [38] Luštica, A., CCD And CMOS Image Sensors In New HD Cameras," *Proceedings ELMAR, Zadar, Croatia*, 2011, 133-136.
- [39] Lan, Y., Yu, L., Chen, G., Yang, S., Chang, A., Construction and Characterization of NTC Thermistors at Low Temperature, *International Journal of Thermophysics*, 2010, 31(8), 1456-1465, DOI: 10.1007/s10765-010-0790-0.
- [40] White, D., R., Temperature Errors In Linearizing Resistance Networks For Thermistors, *International Journal of Thermophysics*, 2015, 36(12), 3404-3420, DOI: 10.1007/s10765-015-1968-2.
- [41] https://cdn.ozdisan.com/ETicaret_Dosya/617513_223207.pdf Erişim Tarihi: 05.04.2024
- [42] <https://www.microchip.com/en-us/product/mcp3008> Erişim Tarihi: 15.04.2024
- [43] Qun, Q., Xue-Dong, J., Shi-Qing, Z., Design Of Stepping Motor Control System Based On AT89C51 Microcontroller, *Procedia Engineering*, 2011, 2276-2280, DOI: 10.1016/j.proeng.2011.08.426.
- [44] Roy, S., T., Kabir, H., Chowdhury, M., A., M., Simple Discussion On Stepper Motors for The Development of Electronic Device, *International Journal of Scientific & Engineering Research*, 2014, 5(1), 1089-1096.
- [45] <https://techexplorations.com/blog/arduino/blog-the-difference-between-unipolar-and-bipolar-stepper-motors/> Erişim Tarihi: 07.04.2024
- [46] <https://pdf.direnc.net/upload/28byj-48-reduktorlu-step-motor-datasheet.pdf> Erişim Tarihi: 07.04.2024
- [47] Yılmazlar, E., Erdemir, V., Kuşçu, H., Güllü, A., Design Of Stepper Motor Control Interface With Embedded Systems, *International Journal Of Engineering Research And Development*, 2018, 14(6), 17-22.
- [48] https://cdn.ozdisan.com/ETicaret_Dosya/603053_5216258.pdf Erişim Tarihi: 07.04.2024
- [49] Erişti, E. Görüntü İşlemede Yeni Bir Soluk, *OPENCV. XII. Akademik Bilişim Konferansı Bildirileri*, 10-12 Şubat, 2010, Muğla (Bildiriler Kitabı, 446 s.)
- [50] <https://opencv.org/> Erişim Tarihi: 10.04.2024
- [51] Bradski, G., Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Amerika Birleşik Devletleri, 2008, 557 s.
- [52] Lin, Z., Li, Z. Design and Implementation of Classroom Attendance System Based on Video Face Recognition. 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 12-13 Ocak, 2019, Changsha, China (Bildiriler Kitabı, 385-388).

- [53] Rahmad, C., Asmara, A., R., Putra, H., R., D., Dharma, I., Darmono, H., Muhiggin, I. Comparison Of Viola-Jones Haar Cascade Classifier And Histogram Of Oriented Gradients (HOG) For Face Detection. The 1st Annual Technology, Applied Science, and Engineering Conference, 29-30 Ağustos, 2019, East Java, Indonesia (Bildiriler Kitabı, 732s.).
- [54] Asmara, A., R., Ridwan, M., Budiprasetyo, G. Haar Cascade and Convolutional Neural Network Face Detection in Client-Side for Cloud Computing Face Recognition. International Conference on Electrical and Information Technology (IEIT), 14-15 Eylül, 2021, Malang, Indonesia (Bildiriler Kitabı, 1-5).
- [55] Viola, P., Jones, M., J. Robust Real-Time Face Detection. International Journal of Computer Vision. 2004, 57(2), 137-154.
- [56] Minu, M., S., Kshitij, A., Tiwari, A., Rampuria, P. Face Recognition System Based On Haar Cascade Classifier. International Journal of Advanced Science and Technology. 2020, 29(5), 3799,3805.
- [57] Mantoro, T., Ayu, A., M., Suhendi. Multi-Faces Recognition Process Using Haar Cascades And Eigenface Methods. 6th International Conference On Multimedia Computing And System (ICMCS), 10-12 Mayıs, 2018, Morocco, Fas, DOI: 10.1109/ICMCS.2018.8525935.
- [58] Özel, A., M., Baysal, S., S., Şahin, M. Derin Öğrenme Algoritması (YOLO) ile Dinamik Test Süresince Süspansiyon Parçalarında Çatlak Tespiti. Avrupa Bilim ve Teknoloji Dergisi. 2021, 26, 1-5.
- [59] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wng, G., Cai, J., Chen, T. Recent Advances In Convolutional Neural Networks. Pattern Recognition. 2018, 77, 354-377.
- [60] Kamel, A., Sheng, B., Yang, P., Li, P., Shen, R., Feng, D., D. Deep Convolutional Neural Networks For Human Action Recognition Using Depth Maps And Postures. IEEE Transactions on Systems Man and Cybernetics. 2018, 49(9), 1806-1819.
- [61] <https://www.upgrad.com/blog/basic-cnn-architecture/> Erişim Tarihi: 12.04.2024
- [62] Shinde, S., Kothari, A., Gupta, V. YOLO based Human Action Recognition and Localization. Procedia Computer Science. 2018, 133, 831-838.
- [63] www.tensorflow.org Erişim Tarihi: 20.04.2024
- [64] Howal, S., Jadhav, A., Arthshi, C., Nalavade, S., Shinde, S. Object Detection for Autonomous Vehicle Using TensorFlow. Intelligent Computing, Information and Control Systems ICICCS. Ed.: A. Pasumpon Pandian, Klimis Ntalianis, Ram Palanisamy, Springer, Berlin, Almanya, 2020, 732s.
- [65] Cheung, P., Beckett, P., Kumar, K., D. A Recovery-Point Mechanism for Low-Power Embedded ML Applications. Research Square, 2023, DOI:10.21203/rs.3.rs-3774323/v1.
- [66] <https://thonny.org/> Erişim Tarihi: 15.04.2024

[67] Petkovšek, M., Nemec, M., Zajec, P. Algorithm Execution Time And Accuracy Of NTC Thermistor-Based Temperature Measurements in Time-Critical Applications. Mathematics. 2021, 9(18), 2266s.

[68]<https://medium.com/geeky-bawa/face-identification-using-haar-cascade-classifier-af3468a44814> Erişim Tarihi: 20.04.2024

[69] Rangkuti, H., A., Athala, H., V., Indallah, H., F., Febriansyah, F. Optimizing Hand Gesture Recognition Using CNN Model Supported by Raspberry pi for Self-Service Technology. International Journal On Informatics Visualization. 2023, 7(1), 58-69.

[70] <https://cocodataset.org/#home> Erişim Tarihi: 25.04.2024

[71] Caesar, H., Uijlings, J., Ferrari, V. COCO-Stuff: Thing and Stuff Classes in Context. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18-23 Haziran, 2018, Salt Lake City, USA (Bildiriler Kitabı, 1209-1218).