T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ROBOT HÜCRELERİNDE KAMERALI GÜVENLİK SİSTEMİNİN OPTİMAL OLARAK TASARLANMASI

Merdan ÖZKAHRAMAN

DOKTORA TEZİ

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Danışman

Prof. Dr. Haydar LİVATYALI

Şubat, 2022

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ROBOT HÜCRELERİNDE KAMERALI GÜVENLİK SİSTEMİNİN OPTİMAL OLARAK TASARLANMASI

Merdan ÖZKAHRAMAN tarafından hazırlanan tez çalışması 23.02.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı, Mekatronik Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. Haydar LİVATYALI
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Haydar LİVATYALI, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Cüneyt YILMAZ, Üye
Yıldız Teknik Üniversitesi _____

Doç. Dr. Zeki Yağız BAYRAKTAROĞLU, Üye
İstanbul Teknik Üniversitesi _____

Dr. Öğr. Üyesi Ertuğrul BAYRAKTAR, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Bekir AKSOY, Üye
Isparta Uygulamalı Bilimler Üniversitesi _____

Danışmanım Prof. Dr. Haydar LİVATYALI sorumluluğunda tarafımca hazırlanan, Robot Hücrelerinde Kameralı Güvenlik Sisteminin Optimal Olarak Tasarlanması başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Merdan ÖZKAHRAMAN

Aileme

ve

biricik eşime

TEŞEKKÜR

Yıldız Teknik Üniversitesinde doktora programına başladığım günden itibaren eğitimde rolü olan herkese, üniversite personeline, ders döneminde bilgi ve yönlendirmeleri ile akademik gelişimime büyük katkı sunan Mekatronik Mühendisliği Bölümü öğretim üyelerine, tez konumun seçilmesinde büyük fikir sahibi olan, sonraki tez izleme toplantılarında ise tecrübesi ve bilgisi ile elinden gelen tüm desteği esirgemeyen Sayın Doç. Dr. Cüneyt Yılmaz'a, tez izleme toplantılarında tecrübesiyle yol gösteren destek olan Sayın Doç. Dr. Zeki Yağız Bayraktaroğlu'na ve tabii ki tüm doktora sürecim boyunca danışmanlığımı yürüten her konuda bana değerli öğütleriyle yol gösteren, ilgili süreçteki tüm endişelerimde yanımda olan, gerek akademik gerekse de sosyal olarak deneyimleriyle bana yol gösteren, emeğini ödeyemeyeceğim saygıdeğer Prof. Dr. Haydar Livatyalı Hocama teşekkürü borç bilirim.

Personeli olduğum, Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi Mekatronik Mühendisliği bölümüne, doktora yaptığım süreçte desteklediklerini esirgemeyen, başta dekanımız Prof. Dr. Reşat Şelbaş'a, Bölüm başkanımız Doç. Dr. Fatih Yılmaz'a, eski oda arkadaşım, kıymetli Hocam Doç. Dr. Abdullah Genç'e, tezim süresince yapay zekâ alanında tüm sorularıma sabırla cevap veren, emeklerini esirgemeyen Sayın Dr. Öğr. Üyesi Bekir Aksoy'a, mesleğime başladığım günden itibaren hep yanımda olan, kendisine hissettiğim manevi bağı kelimelerle anlatamayacağım, emeklerini hiçbir zaman ödeyemeyeceğim çok değerli hocam Prof. Dr. Hilmi Cenk Bayrakçı Hocama tüm kalbimle teşekkürü borç bilirim.

Bir evlat, bir eş ve bir baba olma duygusunu bana yaşatan sevgili aileme, maddi manevi her zaman yanımda olan canım babama, desteğini ve sevgisini kelimelerle anlatamayacağım bir tanecik anneme, bütün bu süreçte tüm stresli anlarımda yanımda olan, ben bu satırları yazarken bile tezimi bitirebilmem için küçük oğlumuza tek başına sabırla bakan sevgilim, eşim Didem'e ve belki büyürse okur umuduyla oğlum Vefa'ya, tüm ailemizin birlikte ve mutlu nice güzel yarınları olması dileğiyle teşekkür ederim.

Merdan ÖZKAHRAMAN

İÇİNDEKİLER

SİMGE LİSTESİ xii

KISALTMA LİSTESİ ixii

ŞEKİL LİSTESİ xii

TABLO LİSTESİ xv

ÖZET xvi

ABSTRACT xviii

1 GİRİŞ 1

1.1 Literatür Özeti 1

1.1.1 Robot Kol Kullanılan İş Hücrelerinin Literatürdeki Yeri 1

1.1.2 Robot Kol Kullanılan İş Hücrelerinin İş ve İşçi Güvenliği Açısından Literatürdeki Yeri 4

1.1.3 Robot Kol Kullanılan İş Hücrelerinin Görüntü İşlemeye Dayalı Sistemler Açısından Literatürdeki Yeri 7

1.1.4 Robot Kol Kullanılan İş Hücrelerinin Görüntü İşlemeye Dayalı Sistemler Açısından Literatürdeki Yeri 11

1.1.5 Robot Kol Kullanılan İş Hücrelerinin En Uygunu Bulma Açısından Literatürdeki Yeri 13

1.2 Tezin Amacı 13

1.3 Hipotez 18

1.4 Tezin Arkaplanı 15

1.5 Tezin Önemli Noktaları 18

1.6 Araştırma Soruları 20

1.7 Tezin Değeri 20

1.8 Genel Bakış 21

2 MATEMATİKSEL MODEL 23

2.1 Genel Yaklaşımlar 23

2.2 En Uygun Çözümün Bulunması (Optimizasyon) 32

2.2.1 Optimizasyon Kısıtları 34

2.2.2 Kamera Çözünürlüğünün Optimizasyon Kısıtlarına Etkisi 35

3 SİSTEM TASARIMI 39

3.1 Matematiksel Modelin Uygulaması 39

3.2 En Uygun Çözümün Yapay Zekâ Algoritması Kullanılarak Doğrulanması 49

3.3 2 Boyutlu Kamera Görüntülerinin Birleştirilmesi 52

4 SİSTEMİN GERÇEKLENMESİ 56

4.1 En Uygunu Bulmak İçin Bilgisayar Yazılımının Gerçeklenmesi 56

4.2 Robot Kol Kullanılan İş Hücresinin Deneysel Tasarımının Gerçeklenmesi 60

4.3 Deneysel Robot Kol Gövde Yapısı 61

4.4 Deneysel Robot Kol Kontrolcü Yapısı 63

4.5 Deneysel İş Hücresinin Çevresel Yapısı 66

4.6 Robot Kol Kullanılan İş Hücresinin Görüntü İşlemeye Dayalı Güvenlik Sisteminin Gerçeklenmesi 70

4.7 Programlama Dili 71

4.8 Görüntü İşleme Kütüphaneleri 73

4.9 Görüntü İşlemeye Dayalı Güvenlik Sistemi Yazılımı 75

5 GÜVENLIK SISTEMINDE YAPAY ZEKÂ YAKLAŞIMI 83

5.1 Yapay Zekâ Tabanlı Nesne Tanıma ve Sınıflandırma 83

5.2 Veri Seti 85

5.3 Yapay Zekâ Mimarilerinin Karşılaştırılması ve Modellerin Sonuçları 89

6 DENEY DÜZENEĞI TASARIMI 99

6.1 Deney Düzeneğinin Oluşturulması 99

6.2 Hız Kontrollü Yabancı Cismin Sisteme Etkisi 103

6.3 Aydınlanma Şiddetinin Sisteme Etkisi 104

7 BULGULARIN DEĞERLENDIRILMESI 107

8 SONUÇ VE ÖNERILER 112

KAYNAKÇA 115

A EKLER 119

TEZDEN ÜRETİLMİŞ YAYINLAR 133

SİMGE LİSTESİ

$\Delta\theta$	Açısal Değişim
ω	Açısal Hız
Ø	Çap
v	Çizgisel Hız
R	Dairenin Çapı
r	Dairenin Yarıçapı
n	Değişken
°	Derece Sembolü
fi	Eğitilebilir Doğrusal Olmayan Eşlemeler
wi	Eğitilebilir Parametreler
∈	Elemanıdır
b	Elips Kısa(Minör) Çap
a	Elips Uzun(Majör) Çap
S	Emniyet Mesafesi
B	Etkili Piksel Sayısı (Halka Piksellerle)
A	Etkili Piksel Sayısı(Halka Pikselsiz)
X'	F(X) Fonksiyonunun Maksimum Ve Minimum Noktası
D	Görüntü Sensöründe Etkili Piksel
K	İnsan Vücudunun ya da Bölümlerinin Tahmin Edilen Yaklaşma Hızı
h	Kamera Yerleşim Yüksekliği
α	Kameraların Lens Açısının Yarısı
α1	Kameraların Lens Açısının Yarısı Çözüm Kümesinin 1. Elemanı
α2	Kameraların Lens Açısının Yarısı Çözüm Kümesinin 2. Elemanı
α3	Kameraların Lens Açısının Yarısı Çözüm Kümesinin 3. Elemanı
θ	Kameraların Yatay Eksenle Arasındaki Açı
kg	Kilogram
cos	Kosinüs
t2	Makinenin Reaksiyon Süresi
C	Maksimum Kullanılabilir Piksel Alanı
m	Metre
µs	Mikrosaniye

mm	Milimetre
ms	Milisaniye
T	Periyot
π	Pi Sayısı
rad	Radian
yi	Residual Unit için Girdi
s	Saniye
sn	Saniye
°C	Santigrat Derece
cm	Santimetre
sin	Sinüs
tk	Sinyalin Belirli Konumda Kalış Süresi
Z	Tam Sayılar
t1	Tedbirin Reaksiyon Süresi
C	Tehlike Alanına Doğru Olası Yaklaşma
Vpp	Tepeden Tepeye Voltaj
E	Toplam Piksel Sayısı
∂	Türev
V	Voltaj
f(x)	x Bilinmeyenine Bağlı Fonksiyon
%	Yüzde
Δt	Zamansal Değişim
f(α,h)	α ve h Bilinmeyenine Bağlı Fonksiyon

KISALTMA LİSTESİ

OpenCV	Açık Kaynak Bilgisayarla Görme
IAS	Akıllı Destek Sistemleri
ABS	Akrilonitril Bütadien Stiren
TCL	Araç Komut Dili
ResNet	Artık Sinir Ağı
AB	Avrupa Birliği
EN	Avrupa Normu
HIC	Baş Yaralanma İndeksi
BSD	Berkeley Yazılım Dağıtımı
CAD	Bilgisayar Destekli Grafik Tasarım
CV	Bilgisayarla Görme
PIC	Çevresel Arabirim Kontrol Cihazı
QVGA	Çeyrek Video Grafik Dizisi
NB	Dar Bant
PWM	Darbe Genişlik Modülasyonu
SVM	Destek Vektör Makineleri
FTP	Dosya Transfer Protokolü
EBM	Elektron Işınlı Ergitme
IMS	Entegre Üretim Sistemleri
FDM	Eriyik Yığarak Modelleme
UART	Evrensel Asenkron Alıcı / Verici
USB	Evrensel Seri Veriyolu
CNN	Evrişimli Sinir Ağı
ADAC	Genel Alman Otomobil Kulübü
GMLVQ	Genelleştirilmiş Matris Öğrenme Vektör Niceleme
GLVQ	Genelleştirilmiş Öğrenme Vektör Kuantizasyonu
GTX	Giga Texel Shader eXtreme

GB	Gigabayt
LDA	Gizli Dirichlet Tahsisi
VGG	Görsel Geometri Grubu
GPU	Grafik İşlemci Ünitesi
ILSVRC	ImageNet Büyük Ölçekli Görsel Tanıma Yarışmasını
IOS	Iphone İşletim Sistemi
2D	İki Boyutlu
TWI	İki Telli Arayüz
HRC	İnsan-Robot İşbirliği
COBOT	İşbirlikçi Robotlar
DT	Karar Ağaçları
SSD	Katı Hâl Sürücüsü
SGC	Katı Zemin Kürleme
LOM	Lamine Nesne İmalatı
MacOS	Macintosh İşletim Sistemi
MP	Megapiksel
CPU	Merkezi İşlem Birimi
MS	Microsoft
PetG	Polietilen Tereftalat
PLA	Polilaktik asit
C	Programlama Dili
C#	Programlama Dili
C++	Programlama Dili
PSPF	Pubik Simfizis Azami Kuvveti
RAM	Rastgele Erişimli Bellek
RF	Rastgele Orman
ReLu	Rektifiye Lineer Birim
HDD	Sabit Disk Sürücüsü
SLE	Seçmeli Lazer Ergitme
SLS	Seçmeli Lazer Sinterleme

SPI	Seri Çevre Arayüzü
NN	Sinir Ağı
RX	Sinyal Alıcı Port
TX	Sinyal İletici Port
STL	Standart Üçgen Dili
SLA	Steryolitografi
TTL	Transistörden Transistöre Mantığı
FTDI	Uluslararası Geleceğin Teknoloji Cihazları firmasının devre kartı
ISO	Uluslararası Standartlar Teşkilatı
3D	Üç Boyutlu
VGA	Video Grafik Dizisi
VC	Viskoz Kriterler
AI	Yapay Zekâ

ŞEKİL LİSTESİ

Şekil 1.1 Tezin gelişim durumu 22
Şekil 2.1 Robot sistemi/hücresi ile ilgili standartlar arasındaki ilişkilerin grafiksel görüntüsü [8] 23
Şekil 2.2 Bir robot kol ve dik bir açıyla yerleştirilmiş bir kamera ve ihtiva eden sistem görseli 25
Şekil 2.3 İki robot kol ve dik bir açıyla yerleştirilmiş bir kamera ihtiva eden sistem görseli 26
Şekil 2.4 İki robot kol ve dik bir açıyla yerleştirilmiş iki kamera ve ihtiva eden sistem görseli 26
Şekil 2.5 Üç robot kol ve dik bir açıyla yerleştirilmiş üç kamera ihtiva eden sistem görseli 27
Şekil 2.6 İki kameralı ve açısı değişebilir sistem görseli 28
Şekil 2.7 İki kamera ve bir robot (üstte) ve önden (sol altta) ve yandan (sağ altta) görünümleri olan bir sistemin çizimi 29
Şekil 2.8 Kesik konik parametrelerinin şekilsel gösterimi 29
Şekil 2.9 Amaç fonksiyonunun üç boyutlu grafiği 31
Şekil 2.10 f(x) foksiyonunun minimum noktası ve –f(x) fonksiyonunun maksimum noktası [40] 33
Şekil 2.11 Piksel ve çözünürlük arasındaki ilişkinin benzetimi [46] 37
Şekil 3.1 Sistem tasarımını parametrelerini belirleyen optimizasyon algoritmasının akış şeması 40
Şekil 3.2 İki ve üç kameralı sistemler için kamera lens açısına göre optimum kamera yüksekliği. 48
Şekil 3.3 Optimum olmayan oryantasyon açısı ve yükseklikte yerleştirilen kamera tarafından taranan alanların farkı 49
Şekil 3.4 Derin öğrenme algoritmasının akış şeması 51
Şekil 3.5 Tek bir kamera kullanıldığında oluşabilen kör bir alan 53
Şekil 3.6 Her piksele dayalı olarak iki simetrik kameranın kısmen maskelenmiş görüntülerinin birleştirilmesi 54
Şekil 3.7 Kısmen maskelenmiş gerçek zamanlı farklı kamera görüntülerinin birleştirilmesi. 55
Şekil 4.1 En uygunu bulma yazılımının görseli 56
Şekil 4.2 En uygunu bulma yazılımının seçenek dizisi görseli 58
Şekil 4.3 En uygunu bulma yazılımı sonuçlar sekmesi görseli 59

Şekil 4.4 En uygunu bulma yazılımı örnek sonuç görseli 60
Şekil 4.5 3 Boyutlu yazıcıda üretilmiş robot kol gövde parçaları 62
Şekil 4.6 3 Boyutlu yazıcıda üretilmiş robot kol gövdesi 63
Şekil 4.7 Servo motor 64
Şekil 4.8 Servo motorlar yerleştirilmiş robot kol gövdesi 65
Şekil 4.9 Arduino Nano 66
Şekil 4.10 En uygunu bulma yazılımına göre sistem tasarım parametreleri 67
Şekil 4.11 20x20mm 6 kanallı alüminyum sigma profil 68
Şekil 4.12 Sigma profil bağlantı elemanları 68
Şekil 4.13 Örnek pleksi malzeme 69
Şekil 4.14 Sigma profilden oluşturulmuş çerçeve tasarımının görseli 70
Şekil 4.15 Güvenlik sisteminin kontrol şeması. 76
Şekil 4.16 Güvenlik sistemi akış diyagramı 77
Şekil 4.17 Matematiksel modele göre tasarımı yapılmış iş hücresi 78
Şekil 4.18 Güvenli alan dışında kalan kısımların maskelenmesi 79
Şekil 4.19 Hareket tespitinin olmadığı durumlar için "movement indicator" görseli 80
Şekil 4.20 Python programının konsol ekranı 82
Şekil 5.1 Tasarlanan görüntü sınıflandırma sisteminin işlem adımları 83
Şekil 5.2 ResNet öğrenme bloğu 84
Şekil 5.3 Birinci giriş yapan obje sınıfları 86
Şekil 5.4 İkinci giriş yapan obje sınıfları 87
Şekil 5.5 Sınıflandırma için iş akışı diyagramı 89
Şekil 5.6 Resnet-152 mimarisi için öğrenme oranı tablosu 90
Şekil 5.7 Resnet-152 modelinin on nesneli karmaşıklık matrisi 91
Şekil 5.8 SqueezeNet mimarisi için öğrenme oranı tablosu 92
Şekil 5.9 SqueezeNet modelinin on nesneli karmaşıklık matrisi 92
Şekil 5.10 AlexNet mimarisi için öğrenme oranı tablosu 93
Şekil 5.11 AlexNet modelinin on nesneli karmaşıklık matrisi 93
Şekil 5.12 SVM modelinin on nesneli karmaşıklık matrisi 94
Şekil 5.13 RF modelinin on nesneli karmaşıklık matrisi 95
Şekil 5.14DT modelinin on nesneli karmaşıklık matrisi 96

Şekil 6.1 Kontrollü hız deneyi için üç boyutlu yazıcıda basılmış farklı boyutlardaki aparatlar 101

Şekil 6.2 Hız kontrollü objenin uyarı bölgesine giriş yörüngesi ve emniyet alanları 102

Şekil 6.3 Hız kontrollü objenin alarm bölgesine giriş yörüngesi ve emniyet alanları 102

Şekil 6.4 İki farklı grafik işlemci ve kamera çözünürlüğü kullanılan deneysel sistemin istenmeyen kontrollü hıza sahip nesnelere karşı tepki süresi grafiği 103

Şekil 6.5 Luxmetre 105

Şekil 6.6 İki farklı grafik işlemcisi ve kamera çözünürlüğünü kullanılan deneysel sistemin farklı aydınlanma şiddetlerine karşı tepki süresi grafiği 106

Şekil 7.1 Büyük ölçekli bir robot çalışma hücresinde en uygun şekilde tasarlanmış güvenlik sisteminin uygulanması 109

Şekil 7.2 Robot kol güvenlik mesafesinin grafiği 111

TABLO LİSTESİ

Tablo 1.1 Robotik sistemler ve görüntü işleme üzerinde gerçekleştirilen son yapay zekâ (AI) uygulamaları. 12

Tablo 3.1 İki Kameralı Sistemde Kamera Yüksekliğinin, Lens Açısına Bağlı Olarak Sınır Değerleri 43

Tablo 3.2 İkiden Fazla Kameralı Sistemde Kamera Yüksekliğinin, Lens Açısına Bağlı Olarak Sınır Değerleri 46

Tablo 3.3 En uygun şekilde yerleştirilmiş kameradan alınan görüntüler ile değiştirilen kamera yüksekliğinden alınan görüntüler arasındaki benzerlik yüzdesi 52

Tablo 3.4 En uygun şekilde yerleştirilmiş kameradan alınan görüntüler ile değiştirilen kamera oryantasyon açısından alınan görüntüler arasındaki benzerlik yüzdesi 52

Tablo 5.1 Nesnelerin gerçek ve piksel boyutu 87

Tablo 5.2 Karşılaştırılan nesnelerin özellikleri 88

Tablo 5.3 Kullanılan mimarilerin test sonuçlarıı 96

Tablo 5.4 Veri seti üzerinde derin öğrenme ve makine öğrenmesi algoritmalarının tahmin süreleri 97

Tablo 6.1 Deney düzeneğinde kullanılacak cismin çizgisel hız, açısal hız ve kontrol programındaki karşılık süreleri 100

Tablo 7.1 Ticari bir örneğin ve tasarlanan deneysel düzeneğinin özelliklerinin karşılaştırılması 110

Tablo 7.2 Ticari bir örneğin ve tasarlanan güvenlik sistemlerinin maliyet karşılaştırması (2022 yılı için yaklaşık fiyatlar) 110

ÖZET

Robot Hücrelerinde Kameralı Güvenlik Sistemlerinin Optimal Tasarımı

Merdan ÖZKAHRAMAN

Mekatronik Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Prof. Dr. Haydar LİVATYALI

Günümüzde, üretim sistemlerinde robot kullanılan iş hücreleri vazgeçilmez hale gelmiştir. Bu sistemler için insan ve iş güvenliği en önemli unsurlardan biridir. Görüntü işlemeye dayalı güvenlik yöntemlerinde üç boyutlu (3D) kamera kullanımı ticarileşmiş bir uygulamadır. Tek 3D kameranın kullanıldığı güvenlik sistemlerinde maliyetin yüksek olması ve kör noktaların oluşması dezavantaj yaratmaktadır. Bu sebeplerle, doktora tezinde, uluslararası güvenlik standartlarıyla uyumlu, iki boyutlu (2D) birden fazla kamera kullanılan bir güvenlik sistemi tasarımı yapılmıştır. Kamera sayısı, kamera yerleşim açısı ve kamera yükseklikleri optimizasyon parametreleri olarak ele alınıp, en uygun çözüm sunulmuştur. Çözümün doğrulanması için iki kamera kullanılan bir deney düzeneği kurulmuştur. Bu düzenekte optimal ve optimal olmayan kamera yerleşimlerinden alınan görseller, derin öğrenme mimarisi olan ResNet-152 kullanılarak karşılaştırılmıştır. Kör noktaları ortadan kaldırmak için, stereo görüntüleme yöntemlerine alternatif, 2D görüntü birleştirme tekniği önerilmiştir. Doğrulama deneyleri, sekiz aydınlatma seviyesinde iki farklı grafik işlemci ve iki farklı kamera çözünürlüğü kullanılarak gerçekleştirilmiştir. Güvenlik sistemine yüksek hızlı girişlerin güvenilir bir şekilde ve 0,1 saniye yanıt süresiyle algılandığı doğrulanmıştır. Ayrıca, ticari sistemler en az

400 lux üstünde kullanılabilirken, tasarlanan sistemin 120 lux aydınlatma seviyesine kadar etkili şekilde çalıştığı kanıtlanmıştır.

Kamera görüntülerine dayalı güvenlik sistemlerinde, yapay zekâ tabanlı nesne algılama mimarilerinin kullanımı sistem esnekliğini, doğruluğunu ve kararlılığını arttırmaktadır. Doktora çalışmasında, sistem denetleyicisi, üç derin öğrenme (ResNet-152, AlexNet, SqueezeNet) ve üç makine öğrenmesi modülü (SVM, RF ve DT) ile birleştirilmiştir. Modüller, alarm bölgesine giren on farklı nesnenin farklı kalitelerdeki 550şer fotoğraf görüntüsü kullanılarak eğitilmiştir. Böylece endüstriyel ortamın değişen koşulları; Kamera titreşimleri, gölgeler, yansımalar, aydınlatma değişimleri vb. bozucu etkiler veri setine dâhil edilmiştir. SqueezeNet derin öğrenme modeli, %95 ile en iyi doğruluğu vermiştir. Ancak makine öğrenmesi tabanlı modellerden DT, yakın doğrulukta tahmini derin öğrenme tabanlı modellerden 100 kat daha hızlı sağladığı bulunmuştur. Böylece güvenlik sistemi olası her türlü değişikliğe hızla uyum sağlayabilmekte ve endüstriyel üretimde oluşabilecek zaman kayıplarının önüne geçilebilmektedir.

Anahtar Kelimeler: Robotik, optimizasyon, iş-hücresi güvenlik sistemi, yapay zekâ, görüntü sınıflandırma

ABSTRACT

Optimal Design of Camera Based Safety Systems in Robotic Cells

Merdan ÖZKAHRAMAN

Department of Mechatronics Engineering

Doctor of Philosophy Thesis

Supervisor: Prof. Dr. Haydar LİVATYALI

Today, work cells using robots have become indispensable for production systems. Human and occupational safety is one of the most important elements for these systems. It has been observed that three-dimensional (3D) camera technology is used in safety methods based on image processing. In safety systems where more than one camera system is used, the high cost and the inability to intervene in blind spots for the robot workstation also creates a major disadvantage. For these reasons, in this doctoral thesis, a two dimensional (2D) safety system using more than one camera was designed in compliance with international safety standards. The number of cameras, camera orientation angle and camera heights were considered as optimization parameters and the most appropriate solution was presented. An experimental setup using two cameras was set up to verify the solution. Images taken from optimal and non-optimal parameters through this setup were compared using ResNet-152, a deep learning algorithm. To eliminate blind spots, 2D image fusion technique has been proposed as an alternative to the stereo imaging methods. Validation experiments were performed using two different graphics processors and two different camera resolutions under varying lighting

conditions. It has been verified that high-speed intrusions into the safety system are detected reliably and with a response time of 0.1 seconds. In addition, while commercial systems cannot operate below 400 lux, the designed system has been proven to work effectively at a minimum illumination of 120 lux.

The use of artificial intelligence based object detection architectures in the safety system based on camera images significantly improves system flexibility, accuracy, and stability. In addition, in this thesis, the system controller is combined with three alternative deep learning (ResNet-152, AlexNet, SqueezeNet) and three machine learning modules (SVM, RF and DT). These modules were trained using photographic images of ten different foreign objects entering the alarm zone. Ten different classes were determined and a total of 5500 camera images were used as dataset. Camera vibrations, shadows, reflections, lighting changes, etc., for use in system verification in the ever-changing conditions of the industrial environment. disruptive effects are also included in the dataset. The SqueezeNet deep learning model gave the best accuracy rate of 95%. However, machine learning based models were found to have 100 times faster prediction time than deep learning-based models. Thus, the safety system can quickly adapt to any possible change, and time losses that may occur in industrial production.

Keywords: Robotics, optimization, work-cell safety system, artificial intelligence, image classification

1
GİRİŞ

1.1 Literatür Özeti

Doktora tezinde yer alan bölümlerin, Dünya'da yapılmış akademik çalışmalar açısından değerlendirilmesi birkaç başlık altında toplanabilmektedir. Bunlar; Robot kolların kullanıldığı iş hücrelerinin literatürdeki genel tanımları ve durumları, robot kolların kullanıldığı iş hücrelerinde iş ve işçi güvenliğine dair çalışmalar, robot kolların kullanıldığı iş hücrelerinde görüntü işlemeye dayalı çalışmalar ve robot kolların kullanıldığı iş hücrelerinde optimizasyon çalışmaları olarak özetlenebilmektedir.

1.1.1 Robot Kol Kullanılan İş Hücrelerinin Literatürdeki Yeri

Bernhardt vd. (2007), insan görevini yerine getirmek yerine destekleyen akıllı destek sistemleri (IAS) geliştirmeyi amaçlamaktadır. Böylelikle, esneklik tam otomatik montaj sistemleriyle gerçekleştirilmemeli, bunun yerine insan çalışanlarının daha iyi entegrasyonuna izin verilen bir sistem ortaya koymaktır [12]. Yapılmış olan çalışma, doktora tez çalışması açısından değerlendirildiğinde, bir güvenlik sistemi üzerine değil daha çok sistem paydaşları arasındaki dağılım ve hiyerarşi odaklı olduğu görülmektedir. Fakat, robot kollardan yararlanılan bir iş hücresinde paydaşların çalışma prensibinde beklentiler ve performanslar açısından yaklaşıldığında bir bakış açısı sunmaktadır.

Oberer vd. (2007), yeni teknolojiler ve süreçler, robotların doğruluğunun ve insanların uyumluluğunun potansiyelini tam olarak kullanmak için doğrudan insan-robot etkileşimi ihtiyacını arttırdığını öngörmektedirler. Bu nedenle, ilgili robot manipülatörünün tehlikeli potansiyeli en aza indirilmesini savunmaktadırlar. Sınırlar ayarlanmalı ve böylece robot ve insan arasında istenmeyen bir temas durumunda, insan için sadece kabul edilebilir bir yaralanma şiddetinin kalması gerektiğini söylemektedirler. Endüstriyel robot sistemleri için mevcut standardizasyon, yakın insan-robot işbirliğinin konusunu yeterince ele almamakta,

böylece en yeni teknolojinin uygulanmasını sınırlamaktadır. Dolayısıyla çalışmada, otomotiv endüstrisindeki yöntemler, robotikteki duruma aktarılabilirliklerini araştırılmaktadır. Robot çarpışması sırasında, insan çarpışma testi mankenleri olarak adlandırılan antropomorfik test cihazları, otomobil çarpışmalarında insanın kinematik tepkilerine benzemektedir. Robot kolunun baş ve göğüse çarpmasını sağlamak için LS-DYNA içinde operatör ve endüstriyel bir robotu temsil eden kukla FAT ES-2 ile bir simülasyon kurulumu gerçekleştirilir. Ortaya çıkan Baş Yaralanma İndeksi (HIC), göğüs için Viskoz Kriterler (VC) ve pelvis için Pubic Symphysis Tepe Gücü (PSPF) tartışıldı, robotikteki durum için potansiyel ve sınırlamaları gösterilmektedir [9]. Robot kollardan yararlanılan bir iş hücresinin önemini ortaya koymuşlar ve buna uygun bir sistem üzerinde çalışmışlardır. Biz çalışmamızda bu tür sistemlerin gerekli olduğunu kabul ederek bu sistemleri daha güvenli hale getirebilecek bir sistem üzerinde durmaktayız. Dolayısıyla yapmış oldukları çalışma otomasyon sistemlerine dair bir bakış açısı sunmaktadır.

Zaeh vd. (2007), üretim şirketlerinin yaşlanan iş güçleri, demografik değişim nedeniyle sanayileşmiş ülkelerin çoğunda yaşlanan nüfusları yansıtma olduğunu vurgulamışlardır. Özellikle manuel seri montajda, maliyet düşürme baskısı yüksek olmakla birlikte, tüm gereksinimleri karşılayamayan engelli ve engelli sayısı, çalışanların daha yüksek bir ortalama yaşı ile artmakta olduğunu söylemektedirler. Bu nedenle, demografik değişimi bilinçli olarak dikkate almak için üretim yapıları ve işyerleri gelecekte gerekli olduğunu ön görülmektedir. Geniş bir ekonomik kabul için, varsayım, eski çalışanların üretkenliği düşürmeden veya onaylanmış montaj organizasyon kurulumlarını değiştirmeden katma değer süreçlerine doğrudan entegrasyonu olması gerektiği söylenmektedir. Dolayısıyla bu çalışmada, yeni bir sistematik yaklaşım gösterilmektedir ve yaşlıların başarılı bir şekilde entegrasyonu için özel serbestlik derecelerinin kullanımını önerilmektedir [13]. Bu bilimsel yayını projemiz açısından değerlendirdiğimizde, endüstriyel alanda otomasyon bir sistem üzerinde çalışmışlardır. Bu sistemlerde paydaşlar arasındaki iş paylaşımı ve görev tanımları açısından, otomasyon sistemlerine dair bir bakış açısı sunmaktadır.

Krüger vd. (2009), montaj süreçlerinin esnekliği ve değiştirilebilirliği, çalışan ve otomatik montaj sistemi arasında yakın bir işbirliği gerektiğini vurgulamışlardır.

İnsan ve robotlar arasındaki etkileşim, özellikle bir robot akıllı bir yardımcı olarak görev yaptığı zaman, bireysel karmaşık montaj süreçlerinin verimliliğini arttırdığını söylemektedirler. İşbirliğini destekleyen mevcut teknolojiler ve insan-makine işbirliği biçimleri hakkında bir değerlendirme sunmuşlardır. Etkin bileşen tedariki ve lojistik de dâhil olmak üzere kooperatif bir sistemin organizasyonel ve ekonomik yönlerini karşılaştırmışlardır [3]. Robot kollardan yararlanılan bir iş hücresinin önemini ortaya koymuşlar ve buna uygun bir sistem üzerinde çalışmışlardır. Yapılmış olan doktora tez çalışması açısından değerlendirildiğinde bu tür sistemlerin gerekli olduğu kabul edilerek, bu sistemleri daha güvenli hale getirebilecek bir sistem üzerinde durulmaktadır. Dolayısıyla robot kollardan yararlanılan bir iş hücresinin kooperatif yapısının organizasyonel ve ekonomik yönlerine dair bir bakış açısı sunulmaktadır.

Takata vd. (2011), imalat şirketlerinin bugün, rekabet güçlerini güçlendirmek için üretim sistemlerinin hem değişebilirliğini hem de verimliliğini artırmaları gerektiğini vurgulamaktadırlar. Hibrit montaj sistemleri bu konuya karşı çözüm olarak geliştiğini ve bu sistemde, insanlar ve robotlar montaj görevlerini yerine getirmek için işbirliği yaptığını söylemektedirler. Dolayısıyla bu çalışmada, hibrid montaj sistemlerinde insan ve robot tahsisini planlamak için bir yöntem önermektedir. Bu yöntem, ilk insan ve robot tahsisini seçmemizi mümkün kılmakta, ürün modellerinde ve üretim hacimlerinde gelecekteki değişikliklerin olası senaryolarını dikkate alarak robot yatırım ve işgücü maliyeti de dâhil olmak üzere beklenen toplam üretim maliyetini en aza indirmektedir [1]. Dolayısıyla, robot kollardan yararlanılan bir iş hücresinin önemini ortaya koymuşlardır. Ayrıca buna uygun bir sistem üzerinde çalışmışlardır. Bu tür sistemlerin gerekli olduğu kabul edilerek, bu sistemleri daha güvenli hale getirebilecek bir sistem üzerinde çalışılması gerekmektedir. Dolayısıyla çalışmada otomasyon sistemlerine dair bir bakış açısı sunulmaktadır.

Durgun vd. (2016), zenginleştirilmiş yaşam koşulları, insan yaşamını uzatarak, yaşlanan nüfusun işyerinde istihdam edilmesine yol açmakta ve kadın istihdamının desteklenmesi ve çalışma alanlarında ergonomik koşulların iyileştirilmesi zorunlu hale gelmiş olduğunu vurgulamışlardır. 4. sanayi devrimi kapsamında geliştirilen

işbirlikçi robotlar (COBOT) ve giyilebilir teknolojiler, endüstriyel robotların daha fazla kullanılmasını ve emek yoğun montaj süreçlerinde katma değeri olmayan işlemlerin ortadan kaldırılmasını sağlamaktadır. Dolayısıyla çalışmalarında, otomotiv endüstrisinde gelişmiş güvenlik sistemlerini orta ölçekli yük robotlarına entegre eden ve AB FP7 program projesi ROBO-PARTNER'in bir parçası olarak geliştirilen hibrid montaj hücre tasarımı sunmuşlardır [14]. Robot kollardan yararlanılan bir iş hücresinin önemini insani ve sosyal şartları da vurgulayarak ortaya koymuşlar ve proje destekli bir sistem üzerinde çalışmışlardır. Dolayısıyla bu çalışmada robot kollardan yararlanılan bir iş hücresinin nasıl olabileceğine ve önemine dair bir bakış açısı sunulmaktadır.

Flanagin, (2016), genel robotik sistemler ve performans analizi ile ilgili bir anket sunmaktır. Aynı zamanda robotikle ilgili 3 alanda da bilgi vermektedir. Bunlar; Yardımlı robot sistemleri, insan-robot etkileşimli robot sistemleri ve otonom robotlar [15]. Dolayısıyla, robot kollardan yararlanılan iş hücrelerinin güncelliği ve durumu hakkında genel bir görüş sunulmaktadır.

1.1.2 Robot Kol Kullanılan İş Hücrelerinin İş ve İşçi Güvenliği Açısından Literatürdeki Yeri

Ogorodnikova, (2008), yıllar boyunca, genel güvenlik ve güvenilirlik alanlarında çeşitli risk değerlendirme yöntemleri ve güvenlik standartları geliştirilmiş ve bunlardan sadece birkaçı insan robotu etkileşim alanına uygun olarak tanımlanmış olduğunu vurgulamaktadır. Dolayısıyla bu çalışmada, insan merkezli robotik çalışma hücresinin risk tahmini ve güvenli tasarımı ile ilgili çeşitli yaklaşımları bir araya getirmiştir. İnsan faktörleri, tehlike analizi, risk değerlendirmesi, güvenlik teknolojisi, insan faktörleri, tehlike analizi, risk değerlendirmesi, güvenlik teknolojisi, insan robotu etkileşimi alanındaki güvenliğin nasıl sağlanabileceğine ve geliştirileceğine dair çeşitli fikirlere yer vermiştir [16]. Dolayısıyla güvenlik standartlarının güncelliğinin nasıl kullanılabileceği, korunabileceği ve dahası geliştirilebileceğine dair bir bakış açısı sunulmaktadır.

Haddadin vd. (2009), fiziksel insan-robot etkileşimi ve işbirliği, gittikçe artan bir önem ve robotik araştırmalarında ana odak noktası haline gelmesi vurgulamışlardır. Ayrıca insan kullanıcıları veya belirsiz ortamlar ile yüksek mobilite ve doğrudan

etkileşim için tasarlanan bir robotun vazgeçilmez bir gerekliliği, insanın hiçbir zaman bir tehdit oluşturmaması gerektiğini söylemektedirler. Ek olarak çalışmada en önemli yaralanma mekanizmalarının çeşitli yönlerini kapsayan, insan-robot etkileşimlerindeki güvenli sistematik değerlendirmemize genel bir bakış sunmuşlardır. Böyle bir manipülatörden kaynaklanan potansiyel yaralanma riskini ölçmek için DLR-Hafif Robot III ile yapılan darbe testleri Alman Otomobil Kulübü'nün (ADAC) standart otomobil çarpışma test tesisleri kullanılarak gerçekleştirilmiştir. Bu testlere dayanarak, farklı ağırlığa sahip birkaç endüstriyel robot değerlendirilmiş ve robot kütlesinin ve hızının etkisi araştırılmıştır. Sonucunda belirli durumlarda düşük ataletli robotlar için bile insan için ciddi bir tehdit oluşturabildiğini savunmuşlardır [17]. Dolayısıyla, güvenlik standartlarının güncelliğinin nasıl kullanılabileceği, korunabileceği ve dahası geliştirilebileceğine dair bir bakış açısı sunulmaktadır.

Matthias vd. (2011), küçük parça montaj operasyonlarının otomasyonuna yönelik endüstriyel gereksinimler, teknolojiyi, insan işçileriyle paylaşılan ortamlarda çalışmaya uygun, en yüksek esneklik ve kullanım kolaylığı sergileyen, ölçeklenebilir robotik otomasyon yönüne yönlendiğini ön görmektedirler. Zorluklardan biri, bu koşullar altında personel güvenliği için çözümler geliştirmektir. Dolayısıyla çalışmada, işbirlikçi robotlar için risk değerlendirmesi için mevcut uygulanabilir yaklaşımlar ve ilgili düşük düzeydeki yaralanma risklerini daha iyi çözebilecek daha ayrıntılı bir gelecek yöntemi ele alınmaktadır [18]. Dolayısıyla, güvenlik standartlarının güncelliğinin nasıl kullanılabileceği, korunabileceği ve dahası geliştirilebileceğine dair bir bakış açısı sunulmaktadır.

Vasic vd. (2013), güvenlik, insan robot etkileşimlerinde (HRI) önemli bir husus olduğunu ve robotların onları çevreleyen insanlara zarar verebilecek güçlü hareketler yapabileceğini vurgulamaktadırlar. Kazaları önlemek için, potansiyel zarar kaynaklarını tanımlamak, robotun etrafındaki kişilerin hangisinin en büyük tehlike içinde olduğunu belirlemek ve robotun bu kişiye verebileceği yaralanma türlerini değerlendirmenin önemli olduğunu öngörmektedirler. Dolayısıyla bu çalışmada, robotların tehlikeli araçları manipüle ettiği ve aşırı hızlı ve zorlu bir şekilde hareket ettiği endüstriyel ortamlardaki güvenlik konularının gözden

geçirilmesiyle başlamaktadır. Ardından kalabalık (insanın yaşadığı) ortamlarda çalışan artan sayıdaki otonom mobil robotlarla ilgili konuları ele almaktadırlar. Tamamen otonom araçların yollarda ve yayaların güvenliği üzerindeki potansiyel faydalarını tartışmaktadırlar [19]. Dolayısıyla, otomasyon sistemlerinin güvenliğinin vurgulandığı bir sistem ortaya konulmuştur. Bununla beraber güvenlik standartlarının güncelliğinin nasıl kullanılabileceği, korunabileceği ve dahası geliştirilebileceğine dair bir bakış açısı sunulmaktadır.

Michalos vd. (2014), AB ROBO-PARTNER isimli bir projeyi sunmaktadırlar. Proje, montaj işlemleri için en son endüstriyel otomasyon sistemlerinin insan yetenekleriyle birlikte entegrasyonunu hedeflemektedir. Operatörlerin otonom ve uyarlamalı robot sistemleriyle güvenli işbirliğini içerecek bir hibrit çözüm elde etmek için robot gücü, hız, öngörülebilirlik, tekrarlanabilirlik ve hassasiyetin insan zekâsı ve becerilerini birleştirmeye odaklanılmıştır. Bu noktada çalışmalarındaki temel noktalar güvenli insan-robot işbirliği (HRC) için sezgisel ara yüzlerin geliştirilmesi, güvenlik stratejileri ve ekipmanlarının kullanılması, koruma çiti olmayan insan robot montaj hücrelerinin kullanılmasına izin verilmesi, verimli planlama programlama ve montajın gerçekleştirilmesi için yöntem ve araçların geliştirilmesidir [20]. Dolayısıyla, otomativ endüstrisinde uygulanmak üzere işçi çalışma süreleri, yük kaldırma potansiyelleri ve otomasyon sistem üzerindeki esneklik açısından, kooperatif robot tasarımının karşılaştırmaları yapılmıştır. Dolayısıyla, hibrit robot teknolojisinin ve insan-robot işbirliği kavramının gelişimini ve endüstrideki durumunu anlamak açısından fayda sağlasa da çalışma kendi spesifik alanıyla kısıtlı kalmıştır.

Weitschat vd. (2018), koruma çiti olmayan robot hücrelerin insanlarla üretim alanlarına başarılı bir şekilde uygulanmasında güvenlik kısıtlarının önemli bir engel oluşturduğu ve güvenlik gereklilikleri nedeniyle verimsizliğe sebep olduğunu söylemektedirler. Çalışmalarında, montaj işlerinde insan-robot işbirliği için yeni bir yaklaşım sunmuşlardır. Robotun daha iyi performans göstermesi için, işbirliğine giren insanların güvenliğini korurken robotun maksimum performanstan yararlanılması için bir insan kolunun kinematiği ve dinamiği, kabiliyetini ve ulaşılabilirliğini belirlemek için kontrol odaklı dinamik bir model ile

açıklanmaktadır [5]. Dolayısıyla, görüntü işleme kullanmadan teknisyenin el ve kol eklemlerinin kinematik ve dinamik analizleriyle insanın yaralanmaya yol açabilecek hareketlerinin tahmini yapılarak iş güvenliğini farklı bir şekilde sağlamayı amaçlamışlardır. Bu açıdan farklılıkları gözeterek güvenlik kriterleri geliştirme açısından farklı bir bakış açısı sunulmaktadır.

1.1.3 Robot Kol Kullanılan İş Hücrelerinin Görüntü İşlemeye Dayalı Sistemler Açısından Literatürdeki Yeri

Gecks vd. (2005), güvenli insan robotu işbirliğini sağlamak için çalışma alanı birkaç sabit kamera tarafından denetlenen bir endüstriyel robot sistemi yer almaktadır. Tüm robot transfer hareketleri, farklı bir görüntü yöntemi kullanarak engelleri tespit ederek çarpışma açısından kontrol edilmektedir. Bir çarpışma tespit edildiğinde, robot hareket yolu buna göre değiştirmektedir. Sunulan algoritma, robotların verimli görüntü işleme yoluyla gerçek zamanlı davranış sergileyen güvenli toplama ve yerleştirme işlemlerini gerçekleştirmesini sağlamaktadır [16]. Dolayısıyla, kameralı bir otomasyon sistemi kurularak robot kollardan yararlanılan bir iş hücresinin paydaşlarının iş planlamasına dair bir sistem ortaya konulmuştur. Bu açıdan, görüntü işleme teknolojisinin otomasyon sistemlerinde nasıl kullanılabileceğine dair bir bakış açısı sunulmaktadır.

Krüger vd. (2005), sanayi üretiminde insan ve robotlar, kazaları önlemek için genellikle birbirinden ayrı olarak çalıştığını vurgulamışlardır. Bu sayede, birlikte çalışabilen bir otomatik ve manuel montajın avantajları esnek ve verimli bir üretim için kullanılamadığını söylemektedirler. İnsan ve robotun ayrılmasının üstesinden gelmek mümkün olsaydı, robotların doğruluğu ve hızı, insan işçilerinin esnekliği ve güvenilirliği ile birleştirileceğini öngörmektedirler. Özellikle karmaşık montaj ve taşıma görevleri için, bu birleşimin yararlı olabileceği söylenmektedir. Robot ve insanın ortak çalışma alanını denetleyen dijital 3D görüntü analizine dayanan bir sistem geliştirilmiştir [4]. Dolayısıyla, kameralı bir otomasyon sistemi kurularak robot kollardan yararlanılan bir iş hücresinin paydaşlarının iş planlamasına dair bir sistem ortaya konulmuştur. Bu açıdan, görüntü işleme teknolojisinin otomasyon sistemlerinde nasıl kullanılabileceğine dair bir bakış açısı sunulmaktadır. Bununla

birlikte, kamera yerleşimleri, sayısı konusunda uygun bir çözüm önerisi sunan, esnek bir tasarım ortaya koyulmamıştır.

Kuhn vd. (2006), insan ve robot işbirliği açısından rehberli robot hareketlerini güvenceye almak için bir yöntem sunmaktadırlar. Bunun için, robotun izin verilen maksimum hızını insana ya da bir sonraki engele olan mesafeye göre sınırlandırır ve etkileşimli insan tarafından sağlanan rehberlik bilgilerini kullanarak etkili bir hız oluşturulmaktadır. Bu nedenle, bir kameranın iki heterojen veri tipini ve bir kuvvet tork sensörünü birleştirmişlerdir. Kameralar, farklı bir görüntü yöntemi uygulayarak robotun çalışma alanını izlemek için kullanılmaktadır. Bu engel bilgisi göz önünde bulundurulduğunda, robot ve insanlar arasında ya da çevrede bulunan nesneler arasında mesafeler hesaplanmaktadır. Her bir görüntü içindeki mesafe, genişletilmiş bir fark görüntü yöntemi ile belirlenmektedir. Her kameradan elde edilen mesafeler, çalışma alanındaki nesne mesafesine gerçek robotu yaklaştırmak için kaynaştırılmaktadır. Bu mesafe, robotun izin verilen maksimum hızını düzenlemektedir ve kuvvet-tork sensörü kılavuz bilgisini, yani miktarı, kuvvetin ve momentin yönünü sağlamaktadır. Bu bilgi, robotun izin verilen maksimum hızı dikkate alarak hareketini oluşturmak için kullanılmaktadır [17]. Dolayısıyla, kameralı bir otomasyon sistemi kurularak robot kollardan yararlanılan bir iş hücresinin güvenlik sebebiyle hız ve güç planlamasına dair bir sistem ortaya koyulmuştur. Dolayısıyla bu makale bize görüntü işleme teknolojisinin otomasyon sistemlerinde nasıl kullanılabileceğine dair bir bakış açısı sunulmaktadır. Fakat, kamera yerleşimleri, sayısı konusunda uygun bir çözüm önerisi sunan esnek bir tasarım ortaya koyulmamıştır.

Kuhn vd. (2007), bir kamera görüntüsünde çok sayıda bilinen ve çoklu bilinmeyen nesne arasındaki minimum mesafeyi hızlı bir şekilde belirlemek için bir yöntem sunmaktadırlar. Bilinen nesneler, bilinen geometri, konum, yönlendirme ve konfigürasyona sahip nesnelerdir. Dolayısıyla bilinmeyen nesneler, bir vizyon sensörü tarafından algılanması gereken ancak bilinmeyen geometri, konum, yönelim ve konfigürasyona sahip nesnelerdir. Ancak bilinen nesneler 3D olarak modellenmekte ve genişletilmekte ve daha sonra bir kamera görüntüsüne yansıtılmaktadır. Kamera görüntüsü bilinen ve bilinmeyen nesneler ve nesne

olmayan alanlarda da dâhil olmak üzere nesne alanlarına ayrılmıştır. Yansıtılan modelin kamera görüntüsünde bilinmeyen olarak sınıflandırılan nesne alanlarını kesişmediği en geniş genişleme yarıçapını arayarak mesafe, tahmini olarak tahmin edilmektedir. Yöntem, yalnızca minimum hesaplama süreleri gerektirmektedir ve gözetim ve güvenlik uygulamaları için kullanılabilmektedir. Bir kamera görüntüsündeki bilinen ve çoklu bilinmeyen nesneler arasındaki minimum mesafeyi hızlı bir şekilde belirlemek için bir yöntem sunulmaktadır. Bilinen nesneler, bilinen geometri, konum, yönlendirme ve konfigürasyona sahip nesnelerdir. Bilinmeyen nesneler, bir vizyon sensörü tarafından algılanması gereken ancak bilinmeyen geometri, konum, yönelim ve konfigürasyona sahip nesnelerdir. Dolayısıyla bilinen nesneler 3D olarak modellenmekte ve genişletilmekte ve daha sonra bir kamera görüntüsüne yansıtılmaktadır. Kamera görüntüsü, bilinen ve bilinmeyen nesneler dahil olmak üzere nesne alanlarına ve nesne olmayan alanlara sınıflandırılmaktadır. Yansıtılan modelin kamera görüntüsünde bilinmeyen olarak sınıflandırılan nesne alanlarını kesişmediği en geniş genişleme yarıçapını arayarak mesafe, tahmini olarak verilmektedir. Yöntem sadece minimum hesaplama süreleri gerektirir ve gözetim ve güvenlik uygulamaları için kullanılabilmektedir [18]. Dolayısıyla, kameralı bir otomasyon sistemi kurularak robot kollardan yararlanılan bir iş hücresine dair bir sistem ortaya konulmuştur. Bu açıdan, görüntü işleme teknolojisinin otomasyon sistemlerinde nasıl kullanılabileceğine dair bir bakış açısı sunulmaktadır. Fakat, kamera yerleşimleri, sayısı konusunda uygun bir çözüm önerisi sunan esnek bir tasarım ortaya koyulmamıştır.

Kragic vd. (2009), bir robotun çevreyi görsel olarak algılama ve bu bilgiyi farklı görevlerin yürütülmesi için kullanması, yani, görüntü işleme sistemlerinin robotik alana entegrasyonu konusunda derinlemesine bir inceleme yaparak çok yönlü bir çalışma sunmaktadırlar [19]. Dolayısıyla, robotik sistemlerde görüntü işleme uygulamalarını her yönüyle ele alan bir derleme ile güncel ve akademik örnekleri sunulmuştur. Bu açıdan, genel itibariyle görüntü işleme teknolojisiyle robotik sistemlerin arasındaki bağlantıyı anlamlandırabilmek için yararlı olduğu düşünülmektedir.

Moreno vd. (2012), yapılandırılmış ışık sistemleri, 3D modelleri elde etmek için basit ve etkili araçlar olduğu vurgulanmaktadır. Kullanıma hazır bileşenler, bir veri projektörü ve bir kamera ile inşa edilmiş, pahalı lazer tarayıcılarla hassas bir şekilde dağıtımı ve karşılaştırması kolay olduğu söylenmektedir. Ancak böyle bir yüksek hassasiyet, kamera ve projektörün her ikisinin de hassas bir şekilde kalibre edilmesi durumunda mümkün olmaktadır. Sağlam kalibrasyon yöntemleri kameralar için iyi oluşturulmuştur, ancak kameralar ve projektörler her ikisi de aynı matematiksel modelle tanımlanabilirken, bu yöntemleri projektörlere nasıl uyarlayacağımız net değildir. Sonuç olarak, yapmış oldukları çalışmada, projektör kalibrasyon tekniklerinin birçoğu, sadeleştirilmiş bir modelden yararlanırken, lens bozulmasını ihmal ederek, hassasiyet kaybına yol açmaktadır. Projektör görüntü düzlemindeki 3 boyutlu noktaların görüntü koordinatlarını tahmin etmek için yeni bir yöntem sunulmaktadır. Yöntem, kalibre edilmemiş bir kameraya dayanır ve alt piksel hassaslığına ulaşmak için yerel homografileri kullanılmaktadır. Sonuç olarak, herhangi bir kamera modeli, radyal ve teğetsel distorsiyon katsayıları olan genişletilmiş iğne deliği modeli veya daha karmaşık lens distorsiyonu modellerine sahip olanlar da dâhil olmak üzere projektörü tanımlamak için kullanılabilmektedir [20]. Dolayısıyla, görüntü işleme sistemi ortaya konulmuştur. Bu açıdan, görüntü işleme teknolojisinin uygulamalarına dair bir bakış açısı sunulmaktadır.

Cherubini vd. (2016), endüstriyel kooperatif robotlar konsepti kapsamında homokinetik eklem montajı için işbirliğine dayalı bir insan-robot üretim hücresinin geliştirilmesini gerçekleştirmişlerdir. Bu noktada asıl yaklaşımları, robot-insan arasındaki ve robot-çevre arasındaki doğrudan fiziksel teması başarıyla yönetebilmektir. Ayrıca, bir risk analizi ile önerilen kurulumun güvenlik standartlarıyla uyumlu olduğunu da söylemektedirler [21]. Dolayısıyla, yapmış oldukları görüntü işleme ile teknisyenin el takibini yaparak robot kolun uç işlevcisinin yörünge kontrolünü yaptıkları görülmektedir. Bu açıdan, ise çalışma uzayı alanı taranıp robotun bütün halinde acil durdurma üzerine çalışılmaktadır. Farklılıkları gözeterek güvenlik standartları konusunda yarar sağlamaktadır.

1.1.4 Robot Kol Kullanılan İş Hücrelerinin Görüntü İşlemeye Dayalı Sistemler Açısından Literatürdeki Yeri

Derin öğrenme, birden çok işleme katmanı kullanarak verilerdeki üst düzey soyutlamaları modellemeye çalışan bir yapay zekâ (AI) mimarisi dalıdır. Derin öğrenme modellerinden biri olan evrişimli sinir ağı (CNN), 1998 yılında LeCun tarafından el yazısı sayılarını tanımak ve nesne tanımaya yardımcı olmak için kullanılmıştır [21]. Daha sonra CNN'ler güçlü bir sınıflandırma aracı olarak ortaya çıkmıştır. Nesne sınıflandırma yarışmalarında sıklıkla kullanılmaktadır [22, 23].

Çeşitli alanlardaki yapay zekâ uygulamalarına bakıldığında tıbbi, endüstriyel imalat, hizmet ve kişisel kullanım amaçlı robotik sistemler bulunmaktadır. ResNet'in nesne algılama ve anlamsal görüntü segmentasyonu gibi çeşitli görevlerde önceki modellerden daha iyi performans gösterdiği rapor edilmiştir [24, 25]. SqueezeNet modeli, AlexNet'in verimliliğine ve kabiliyetine meydan okumak için tasarlanmıştır [26]. Neredeyse 50 kat daha az parametreyle, derin sinir ağı, AlexNet'in büyük ölçüde küçültülmüş bir boyutta elde ettiği yüksek doğruluk düzeyine ulaşmayı başarmıştır.

Gerçekleştirilen tez çalışmasında, deneysel bir kamera tabanlı güvenlik sisteminde kullanım için CNN tabanlı yapay zekâ algoritmaları ResNet, AlexNet ve SqueezeNet seçilmiştir. Çitsiz robotik çalışma hücrelerinde görüntü tabanlı bir güvenlik sistemi için CNN mimarisinin kullanılmasının yanıt hızını artırması beklenmektedir. CNN tabanlı modüllerin doğruluğunu karşılaştırmak için sistemde makine öğrenmesi tabanlı sınıflandırma algoritmaları da test edilmiştir.

Destek vektör makineleri (SVM), istatistiksel verilere dayalı kontrollü sınıflandırma algoritmalarıdır [27]. SVM, veri madenciliğinden görüntü sınıflandırmaya, yüz tanımadan ses analizine kadar birçok uygulamada kullanılabilir. SVM, doğrusal ve doğrusal olmayan destek vektör makineleri olarak iki alternatife ayrılmıştır. Sınıflandırma işlemi, iki sınıfı ayırmak için uygun bir düzlem seçilerek doğrusal destek vektör makinelerinde gerçekleştirilir. Doğrusal olmayan bir destek vektör makinesinde doğrusal bir düzlem çizilemediğinden, farklı çekirdek yöntemleri görüntüyü sınıflandırmaktadır.

Karar ağaçları (DT), hem sınıflandırma hem de regresyon problemlerinde kullanılan en popüler makine öğrenme algoritmalarından biridir [28]. Amacı, veri özelliklerinden basit kurallar çıkararak ve bu kuralları öğrenerek bir değişkenin değerini tahmin eden bir model oluşturmaktır. Rastgele orman (RF) algoritması ise bir karar ağacı gibi hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Çalışma prensibi çoklu karar ağaçları oluşturur. Bir sonuç üretmeye hazır olduğunda bu karar ağaçlarındaki ortalama değer alınır ve sonuç üretilmektedir.

Literatürde robotik sistemler üzerinde yapay zekâ tabanlı sınıflandırma uygulamaları gözden geçirilmiştir. Derin öğrenme, makine öğrenme modelleri ve doğruluk performansları Tablo 1.1'de gösterilmiştir. Buna göre ResNet, AlexNet ve SqueezeNet derin öğrenme algoritmaları en umut verici sonuçları vermiştir. SVM, RF ve DT makine öğrenimi modelleri de iyi performans göstermektedir. Ek olarak daha hızlı olduğu görülmüştür. Bu nedenle, tez çalışmasında bu altı model seçilmiştir. Derin öğrenme ve makine öğrenmesi modellerinin doğrulukları ve tahmin süreleri nesne sınıflandırma algoritmalarına dayalı olarak karşılaştırılmıştır.

Tablo 1.1 Robotik sistemler ve görüntü işleme üzerinde gerçekleştirilen son yapay zekâ (AI) uygulamaları

Model	**Veriseti**	**Doğruluk Oranı**	**Kaynak**
GLVQ, GMLVQ, ve SVM	4000 görsel	96%'ya kadar.	[29]
SVM	180 çerçeve	82%'ya kadar.	[30]
SVM	284 örnek	93%'ya kadar.	[31]
LDA, NB, NN, RF, ve SVM	11 sınıf	98%'ya kadar.	[32]
ResNet-50	30 görsellik 22 sınıf	80%'ya kadar.	[33]
SqueezeNet	885 görsel	90%'ya kadar.	[34]
AlexNet	3000 görsel	84%'ya kadar.	[35]

1.1.5 Robot Kol Kullanılan İş Hücrelerinin En Uygunu Bulma Açısından Literatürdeki Yeri

Lien vd. (2001), hibrid montaj sistemleri, manuel finisaj hattını besleyen bir otomatik montaj bölümünden oluştuğunu söylemektedirler. Standart temel yapıya sahip ürünlerin orta hacimli montajı ve manuel son işlem gerektiren birçok sistem için rekabet ettiklerini öngörmüşlerdir. Manuel montaj için iki temel ilke açıklanmaktadır. Üretim hacminin fonksiyonu olarak olası çıktılar için teorik bir model, montaj operasyonlarının sayısı, ürün çeşitlerinin sayısı ve istasyon çalışma süresi dengesizliği sunulmuştur. Bu model, belirli bir dizi koşul için performansın tahmin edilmesini ve optimal sistemin seçilmesini sağlamaktadır [36]. Dolayısıyla, bir otomasyon sisteminin teorik modeliyle birlikte üretimi artırmak amacıyla bir optimizasyon problemi ortaya koyulmuştur. Bu açıdan, otomasyon sistemlerinde en uygunu bulma çalışması konusunda bir bakış açısı sunulmaktadır.

Pichler vd. (2017), insan-robot işbirliği yapan robotik sistemlerin endüstriyel alanlarda uygulamalardan yoksun kaldığını söylemektedirler. Dolayısıyla bu alandaki payın artması için makalelerinde, insan robot etkileşimlerinin sezgisel bir şekilde oluşturulmasına olanak tanıyan insan robot işbirliği için bir platform sunmaktadırlar. Bu platformu kullanarak, çalışanın insan ve robot arasında farklı seviyelerde paylaşılan bir otonomluk sağladıklarını söylemektedirler [37]. Dolayısıyla, insan robot arasındaki görev ve zaman paylaşımı üzerine çalışma yaptıklarını ve bunun için bir platform ortaya koymuş oldukları görülmüştür. Endüstriyel alanda robot-insan işbirliğinin olduğu robotik sistemlerin yaygınlaşmasına yönelik avantajları arttırabileceği görülmektedir.

1.2 Tezin Amacı

Yapılmış olan tez çalışmasının amacı, robot kol kullanılan iş hücrelerinde, iş ve işçi güvenliğini sağlamak amacıyla iki-boyutlu kameraların konum, pozisyon ve sayıları belirlenerek var olan sisteme etkisini tespit etmektir. Ek olarak bu çalışmanın amacı, literatürde eksiklik olarak görülen bu belirsiz değişkenleri bir optimizasyon problemi olarak ele almaktır. Tasarlanmış olan matematiksel model ile öngörülmüş bir senaryo üzerinde uygulanması ve kıyaslanan model ile doğruluğunun karşılaştırılması hedeflenmektedir. Burada doğrulamanın temel amacı

gerçekleştirilmiş olan çalışma ile var olan yüksek maliyetli sistemlere kıyasla daha verimli çalışması sağlanacak şekilde optimize edilmiş uygun maliyetli bir alternatif sunulduğunu göstermektir.

Yapılan tez çalışmanın ana odaklarından biri de mevcut iş istasyonlarında kör noktanın kalmaması iş ve işçi güvenliğini pozitif yönde etkileyeceği gibi maliyetlerin düşürülmesi ülkemiz üretim sanayi için de bir katma değer sağlamasının sağlanmasıdır. Ek olarak yapılacak en uygun şekle sokma çalışmaları ile mevcut akademik literatüre de bir yenilik getirmesi beklenmektedir.

1.3 Hipotez

Robot kol kullanılan iş hücrelerinde güvenliğin sağlanması konusunda çok fazla araştırma yapılmıştır. İlgili alanda standartlar ile geliştirilmiş ve en verimli hale getirilmeye çalışılmıştır. Buna rağmen, güvenlik konusunda asıl gözlemi yapacak olan ekipman, kamera, sensör vs. hakkında literatürde az sayıda çalışma olduğu görülmektedir. Özellikle bu güvenlik sistemlerinin sayısı, yerleşimi ve kendi özellikleri üzerine çok az sayıda çalışma bulunmaktadır. Bununla birlikte, maliyeti yüksek sistemlere alternatif olabilecek, fakat, matematiksel olarak optimize bir sistem üzerine bir çalışma görülmemektedir.

Görüntü işleme yöntemleriyle gerekli olan iş güvenliği en verimli haline gelecek şekilde çalışmalar ortaya konulmuş olsa da bu görüntü işleme sensörlerinin optimum kullanılması konusuna nispeten daha az dikkat gösterildiği anlaşılmaktadır. Dolayısıyla görüntü işleme sensörlerinin en uygun kullanılması ile ilgili birçok soru hala cevaplanmamıştır. Farklı robot kol boyutlarına göre, farklı kamera çözünürlüklerine göre, bu görüntü sensörlerinin yerleşim ve pozisyon faktörleri bilinmemektedir. Bu teorik önerileri doğrulamak için literatürde ampirik bir kanıt mevcut olmasa da, yapılmış olan çalışmada matematiksel bir model ortaya konarak, gerçekleştirilmesi ve doğrulaması yapılmış bir senaryoyla bir çözüm önerilmektedir.

Yaygın olarak kullanılan stereo görüntüleme yöntemlerine alternatif olarak yeni bir iki boyutlu görüntü birleştirme tekniğinin kör noktaları ortadan kaldırmada daha başarılı olabileceği önerilmektedir.

Endüstriyel ortamın, kamera titreşimleri, gölgeler, yansımalar, aydınlatma değişimleri vb. bozucu etkilerini gidermek, daha doğru ve kararlı bir güvenlik sistemi ortaya koyabilmek için geleneksel görüntü işleme metotlarının yerine yapay zekâ tabanlı mimarilerin kullanılması önerilmektedir.

1.4 Tezin Arkaplanı

Son yıllarda robotik teknolojisinin geldiği nokta düşünüldüğünde Dünya için vazgeçilmez bir noktada olduğu görülmektedir. Yaşamın her alanında rol almaya başlayan bu teknolojinin son yüzyıl içerisinde köklü bir tarihe sahip olduğu bilinmektedir. Son elli yıl içerisinde ise robotik teknolojilerin, özellikle endüstriyel üretim alanında vazgeçilmez hale geldiği görülmektedir. Günümüzde endüstriyel alanda robot teknolojilerinin geldiği nokta düşünüldüğünde sadece insan gücünden yararlanılan bir üretim metodunun sürdürülebilir olmadığı görülmektedir. Dolayısıyla gelişen teknolojinin endüstriyel üretim alanında daha verimli üretim sağlanabilmesi için yapılan çalışmaların artmış olduğu görülmektedir.

Otomasyon sistemlerinde hedeflenen hızlı ve doğruluğu yüksek üretim amacı aynı parçanın defalarca üretilmesi, üretilen ürün çeşitliliği konusundaki esnekliği azaltmıştır. Buna rağmen, hızlı değişen teknoloji ve farklı müşteri tercihleri üretim bandını etkilediği gibi son üründe de öngörülebilen veya öngörülemeyen değişiklikler yaratmaktadır. Bu değişikliklere uyum sağlayabilmek için üretim esnekliği gerekmektedir [1]. Ancak, üretimde esneklik konusu seri üretim maliyetleri düşünüldüğünde uygulanabilmesi kolay değildir. Bu düşünceyle birkaç farklı üretim sistemi ortaya konmuştur. Bunlardan biri yeniden yapılandırılabilir üretim sistemleri [2], diğeri hibrit üretim sistemleridir [3]. Hibrit sistemler insan ve robot işbirliğini içeren üretim sistemidir. Bu sistemlerde, robot verimliliğinin yanı sıra insan faktörünün üretimdeki potansiyel esnekliğinin kullanımını da en üst düzeye çıkarmak amaçlanmaktadır. İnsan ve robot işbirliğini içeren teknolojilerdeki son gelişmeler, robotların aynı çalışma alanında insanlarla çalışmasını sağlayabilir [4].

Çitsiz robotik üretim sistemleri, son birkaç on yıl içerisinde ortaya çıkmıştır ve hala geliştirilmesi devam etmektedir. Bu sistemler, robot çalışma alanları ve komşu alanlar arasında herhangi bir ayırıcı içermeyen çalışma hücrelerinden oluşmaktadır.

Dolayısıyla iş ve işçi güvenliği gereksinimlerinin yerine getirilmesi için bazı aktif önlemler gerekmektedir. Çitsiz ve robot destekli insan güdümlü bir üretim sistemi, çitlerle çevrili tam otomatik sistemlere göre önemli avantajlara sahiptir. Üretimde başarılı bir uygulamanın önündeki en kritik engel, güvenlik gerekliliklerinden kaynaklanan verimsizliktir [5]. Üç boyutlu (3-D) kameralar, mevcut standartlara göre görüntü işleme tabanlı güvenlik sistemlerinde en yaygın olanıdır [6, 7].

Tek bir üç boyutlu kamera kullanan yüksek maliyetli güvenlik sistemleri, kör noktalara müdahale edilmemesi nedeniyle önemli bir dezavantaja sahiptir. Ü boyutlu kamera yerine birden fazla düşük maliyetli, iki boyutlu (2-D) kamera kullanan bir sistem bu sorunu çözebilir. Ancak, kör noktaları ortadan kaldırarak daha geniş bir alanı taramak için kameraların yerleşim ve yönelimlerine en uygun seçimi yapmak çok önemlidir. Şu anda, "ISO 10218: Endüstriyel Ortamlar için Robotlar Güvenlik Gereksinimleri"nde [8] açıklanan standartlar, birincil robot hücrelerinin güvenlik operasyonları için geçerlidir [9]. Robot ile insan arasındaki mesafe belirli bir güvenlik seviyesinin altında ise robot kolunun yavaşlaması ve durması gerekir. Böylece manipülatörlerin işçilere çarpması engellenir. Tasarlanan güvenlik sisteminde uyarı, alarm ve çalışma bölgeleri olmak üzere üç adet güvenlik alanı bulunmaktadır. Robot kolunun giriş algılandığında sırasıyla, uyarı bölgesinde yavaşlaması, ardından alarm bölgesinde ise durması gerekmektedir. Bu tez çalışmasında, mevcut uluslararası standartları göz önünde bulundurarak robotik iş hücreleri için çok sayıda düşük maliyetli iki boyutlu kamera kullanan bir güvenlik sistemi tasarlanmıştır. Çitsiz robotik iş hücresinde kamera sayı ve yerleşimleri için optimal bir hesaplama yöntemi önerilmiştir.

Üretim sektöründeki robotik iş hüclerinde çitlerin yanı sıra sensör tabanlı ve kamera tabanlı güvenlik sistemleri kullanılmaktadır. Kamera tabanlı güvenlik sistemleri en ileri teknoloji olarak kabul edilebilir. Bu tür sistemlerde iş hücrelerindeki değişiklikler nedeniyle iş hücresindeki yapının yeniden ayarlanması gerekmektedir. Güvenlik sisteminin çalışma esnekliğine uyum sağlayabilmesi ve çevre koşullarından kaynaklanan gürültünün önüne geçilebilmesi için geleneksel görüntü işleme yöntemlerine yapay zeka algoritmalarının dahil edilmesi kaçınılmazdır.

Robot kol kullanılan iş hücresinin güvenlik sistemindeki amaç, giren yabancı cisimleri tanımak ve ayırt etmek olduğunda, yapay zeka tabanlı bir algoritmanın, sistemin performansını iyileştirebileceği öngörülmektedir. Geleneksel görüntü işleme tabanlı güvenlik sistemleri, dost nesneleri güvenilir bir şekilde tanıyamaz. Bu dost nesneler, bir iş parçası veya hücre bölgesi içinde olmasına izin verilen bir operatör olabilir. Geleneksel sistemlerin bu nesneleri tanıması ve robot kolunun çalışmasını engellememesi için bazı ek donanımlara ihtiyaç duyulmaktadır. Yapay zeka tabanlı güvenlik sistemleri bu konuda çok daha verimlidir. Nesnelerin sistem tarafından istenen ve istenmeyen şekillerde tanınması ile sistemin güvenilirliği arttırılmaktadır. Ancak geleneksel sistemlerin çalışma ortamındaki titreşim, gölgeler, aydınlanma gibi gürültü kaynaklarından etkilendiği bilinmektedir. Artan güvenilirlik ile gelecekteki değişikliklere hızlı tepki veren bir güvenlik sistemi kurmak mümkündür.

Yapılan doktora çalışmasında önerilen tasarım ile çevresel gürültünün sistemin doğruluğunu etkilememesi hedeflenmiştir. Uluslararası güvenlik standartlarına göre robot iş hücresine müdahale edebilecek nesneler yapay zeka algoritmaları kullanılarak sınıflandırılabilir. Yapay zeka (AI), insan zekasını yüksek doğrulukla taklit ederek, öğrenerek ve belirli amaçlara ve görevlere hizmet etmek için "bilgiyi" tekrar tekrar kullanarak bir sistemden elde edilen verileri yorumlayabilen sistemler olarak tanımlanmaktadır [10]. Yapay zeka sistemleri, veriler üzerinde tahmin, sınıflandırma ve kümeleme gerçekleştirmektedir. Ayrıca verilerin belirli bir amaç doğrultusunda işlenmesini sağlamaktadır. Doğayı ve insanı taklit ederek uzman sistemler, bulanık mantık, yapay sinir ağları, genetik algoritmaları, makine öğrenmesi, derin öğrenme gibi problem çözen yapay zeka teknikleri geliştirilmiştir. AI teknikleri, hastalık teşhisi, otonom kontrollü sistemler, yenilenebilir enerji sistemleri, endüstriyel üretim ve robotik sistemler gibi birçok alanda farklı çözümler sunabilmektedir [11].

Bu kısa girişin ardından, tezin önemli noktaları, hipotez, tezin amacı, araştırma soruları, tezin değeri, genel bakış ve literatür özeti sunulmuştur. Önerilen sistemin tasarımı ve optimizasyonu, kör noktalardan kaçınmak için görüntü işleme yönteminin bir açıklaması ile takip edilmiştir. Ardından, matematiksel model

açıklanarak sistem tasarımına geçilmiştir. Optimize edilmiş tasarımın derin öğrenme algoritması kullanılarak test edilmesi, deney düzeneği tasarımı bölümünü, doğrulama testlerinin sonuçları ve tartışmaları, yapay zeka tabanlı nesne sınıflandırma mimarisi ve karşılaştırmalar görülmektedir. Tezin sonuçlar ve öneriler kısmında bulgular tartışılmış ve açıklanmıştır.

1.5 Tezin Önemli Noktaları

Çeşitli görüntü işleme yöntemleriyle iş ve işçi güvenliğinin sağlanması konusu; Endüstriyel üretimde robot kol kullanılan iş hücrelerinin, güvenilirliğinin arttıracağının öngörülmesiyle birlikte büyük bir ilgi görmüştür. Fakat, birden fazla kamera sisteminin kullanıldığı güvenlik sistemlerinde maliyetin yüksek olması ve insan-robot işbirliği istasyonu için kör noktalara müdahale edilememesi zor bir sorun olmaya devam etmektedir.

Robotik teknolojisinin, üretim alanında çeşitliliği ve esnekliği hızla büyümeye devam etmektedir. Dolayısıyla maliyeti çok fazla olan üç-boyutlu kamera sistemleri yerine iki-boyutlu kameralar kullanılarak daha geniş bir açıyı tarayan ve daha uygun maliyetlerle ortaya konan bir sistemin yaratılmaya çalışması önemli bir konu haline gelmiştir.

Türkiye endüstrisi düşünüldüğünde dışa bağımlı ve yüksek maliyetli güvenlik teknolojileri uluslararası pazarda rekabeti zorlaştırdığı gerçeği de bu yönde çalışmalar yapılması için bir motivasyon sağlamaktadır.

Kamera teknolojilerinin gelişimi, az fonksiyonlu ama işe yarar kameraların maliyetlerinin günümüzde daha düşük oluşu birden fazla kamerayla uygun maliyetli bir güvenlik sisteminin tasarlanabileceği umudunu doğurmuştur. Fakat, iki-boyutlu kameraların konum ve sayıları belirlenerek görüntüleri birleştirme, derinlik kazandırma veya görüntü işleme yöntemlerine entegre edilebilir hale getirmek maksadıyla en uygun kamera konum ve sayısının belirlenmesi konusunda literatürde bir çalışma bulunmamaktadır. Bu eksiklik bir optimizasyon problemi olarak ele alındığında, birden fazla kamera kullanılan güvenlik sistemlerinde kamera sayısı, kamera yerleşim konum ve pozisyonları, kamera çözünürlük

değişkenine ve robot kol boyut değişkenlerine bağlı olarak optimum bir çözüm sunulmaktadır.

Çözümün doğrulanması için iki kamera kullanılan bir deney düzeneği kurulmuştur. Bu düzenek üzerinden optimal ve optimal olmayan parametrelerden alınan görseller, bir derin öğrenme algoritması olan ResNet-152 kullanılarak karşılaştırılmıştır.

Kör noktaları ortadan kaldırmak için, yaygın olarak kullanılan stereo görüntüleme yöntemlerine alternatif olarak yeni bir iki boyutlu görüntü birleştirme tekniği önerilmiştir.

Kurulan sistemin doğrulama deneyleri, değişen aydınlatma koşullarında iki farklı grafik işlemci ve iki farklı kamera çözünürlüğü kullanılarak gerçekleştirilmiştir. Güvenlik sistemine yüksek hızlı girişlerin güvenilir bir şekilde ve 0,1 saniye yanıt süresiyle algılandığı doğrulanmıştır. Ayrıca, ticari sistemler 400 lux altında çalışamazken, tasarlanan sistemin minimum 120 lux aydınlatmada etkili bir şekilde çalıştığı kanıtlanmıştır.

Ek olarak, sistem denetleyicisi, üç alternatif derin öğrenme (ResNet-152, AlexNet, SqueezeNet) ve üç makine öğrenme modülü (destek vektör makinesi, rastgele orman ve karar ağacı) ile birleştirilmiştir. Bu modüller, alarm bölgesine giren on farklı yabancı nesnenin fotoğraf görüntüleri kullanılarak eğitilmiştir. On farklı sınıf belirlenmiştir ve toplamda 5500 kamera görseli veri seti olarak kullanılmıştır. SqueezeNet derin öğrenme modeli, %95'lik en iyi doğruluk oranını vermiştir. Buna rağmen, makine öğrenimi tabanlı modellerin, derin öğrenme tabanlı modellerden 100 kat daha hızlı tahmin süresine sahip olduğu bulunmuştur. Böylece güvenlik sistemi olası her türlü değişikliğe hızla adapte olabilmekte ve çalışma koşullarından kaynaklanabilecek gürültünün önüne geçilerek endüstriyel üretimde oluşabilecek zaman kayıplarının önüne geçilebilmektedir.

Ortaya konulmuş olan tez çalışması ile var olan yüksek maliyetli sistemlere kıyasla daha verimli çalışması sağlanacak şekilde optimize edilmiş uygun maliyetli bir alternatif sunulmaktadır. Mevcut iş istasyonlarında kör noktanın kalmaması iş ve

işçi güvenliğini pozitif yönde etkileyeceği gibi maliyetlerin düşürülmesi Türkiye üretim sanayisi için de bir katma değer sağlayacağı düşünülmektedir.

1.6 Araştırma Soruları

İlk araştırma sorusu, robot kol kullanılan iş hücrelerinde mevcut yüksek maliyetli kameralı güvenlik sistemlerine alternatif, uygun maliyetli ve güvenirliliği arttırılmış bir sistem ortaya koyulabilir mi?

İkinci araştırma sorusu, uygun maliyetli olan alternatif güvenlik sistemlerinde güvenirliliğe etkisi olan sistem değişkenleri nelerdir?

Bu araştırmanın üçüncü araştırma sorusu ise, sistem değişkenleriyle bir matematiksel model kurulup optimizasyon çalışması yapılarak, en uygun sistem tasarımı ortaya koyulabilir mi?

Bu çalışmayı motive eden bir diğer araştırma sorusu, endüstriyel üretim alanında iş ve işçi güvenliği kontrol sistemlerinin maliyetlerin düşürülmesi ülkemiz üretim sanayi için de bir katma değer sağlayabilir mi?

Bu tez çalışmasında ele alınan özel araştırma soruları aşağıdaki gibidir:

-Kamera sayısının, kamera yerleşim yüksekliğinin, kamera yerleşim açısının, kamera çözünürlüğünün güvenlik sistemi üzerindeki etkisi nedir?

-Kullanılan robot kol boyutlarının güvenlik sisteminin tasarımındaki etkinliğinde farklılıklar var mı?

-Kamera çözünürlüğü ve yerleşim konumu arasındaki ilişkiler nelerdir?

-İki boyutlu görüntülerin birleştirilmesinde işlem maliyeti nasıl düşürülebilir?

-Geleneksel görüntü işleme yöntemleri yerine yapay zeka tabanlı mimarilerin etkisi nedir?

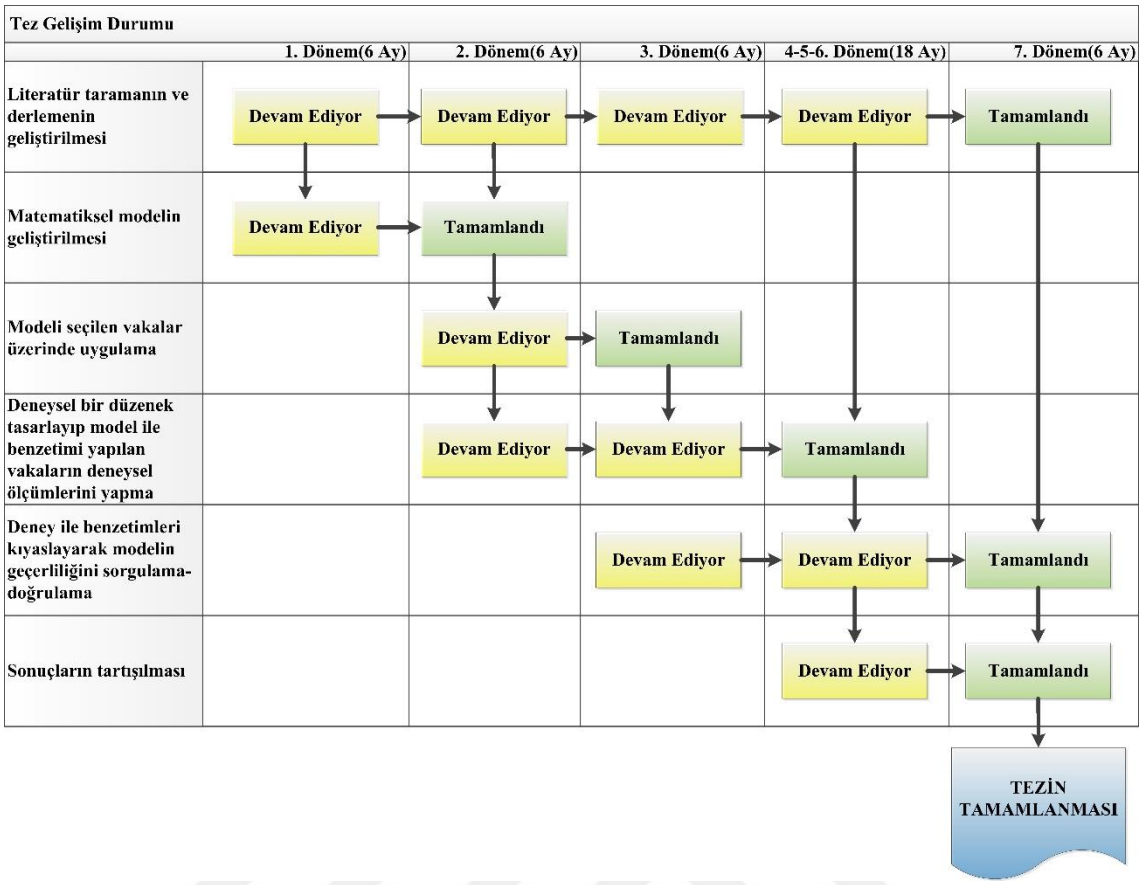
1.7 Tezin Değeri

Robot kol kullanılan mevcut iş istasyonlarında iş güvenliğinin sağlanması konusunda kör noktanın kalmaması, iş ve işçi güvenliğini pozitif yönde etkileyeceği gibi maliyetlerin düşürülmesi sorununa bir alternatif sunulmaktadır.

Yapılmış olan doktora çalışması, kameralı güvenlik sistemlerinin daha verimli ve uygun maliyetli kullanımını genişletmektedir. Bu çalışma aslında neden her küçük ayrıntının endüstriyel alanda bir mühendislik problemi olarak ele alındığında verime, güvenirliliğe ve maliyete pozitif yönde etkisi olduğunu ortaya koymaya yardımcı olmaktadır. Bu çalışmanın sonucu öncelikle Türkiye'deki endüstriyel üretim alanında faaliyet gösteren yatırımcılar için yararlı olacağı, sonrasında ise Dünya'daki endüstriyel üretim alanında faydalı olacağı düşünülmektedir.

1.8 Genel Bakış

Bu bölüm de dâhil olmak üzere yapılmış olan çalışmanın ne olduğu ve ne amaçla yapıldığı yukarıda verilen bölümlerde açıklanmaktadır. Robot hücrelerinde kameralı güvenlik sisteminin optimal olarak tasarlanması isimli çalışmanın ana hatları verilmektedir. Bu çalışma altı ana başlıkta tamamlanmıştır. Başlık 1, literatür taramanın ve derlemenin geliştirilmesidir. Başlık 2, matematiksel-modelin geliştirilmesidir. Başlık 3, sistem tasarımıdır. Başlık 4, matematiksel modelin seçilen vakalar üzerinde uygulanmasıdır. Başlık 5, yapay zekâ yaklaşımı ile modelin geçerliliğini sorgulama ve doğrulama adımıdır. Başlık 6, deneysel bir düzenek tasarlayıp model ile benzetimi yapılan vakaların deneysel ölçümlerinin yapılmasıdır Başlık 7, bulguların tartışılması ve Başlık 8, sonuç bölümüdür. Bu adımlar ve gerçekleştirilme süreleri aşağıdaki Şekil 1.1'de görülmektedir.

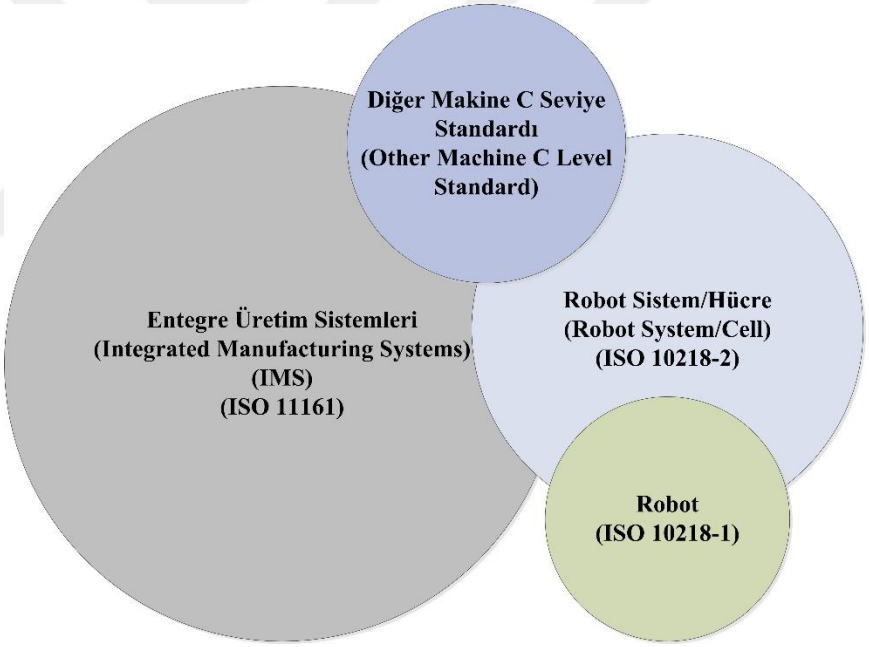


Şekil 1.1 Tezin gelişim durumu

2
MATEMATİKSEL MODEL

2.1 Genel Yaklaşımlar

Tez çalışmasında tasarlanan sistemin genel açıklaması robot kol kullanılan iş hücrelerinde robot kolun ulaşabileceği çalışma uzayını kapsayacak şekilde kamera görüntüsü alarak bu görüntü içerisine girdiği algılanan yabancı nesnelere göre acil durum durması yapacak bir güvenlik sistemi tasarımıdır. Bunun için standartlar çerçevesinde robot kolun çalışma uzayını ve bu çalışma alanın emniyet katsayısı çarpımı güvenlik mesafelerini hesaplamak gerekmektedir. Bunun için yararlanılan güvenlik standartları ve kapsadıkları konu itibariyle Şekil 2.1'de görülmektedir.



Şekil 2.1 Robot sistemi/hücresi ile ilgili standartlar arasındaki ilişkilerin grafiksel görüntüsü [8]

Robot kolların yer aldığı iş hücrelerinde emniyet mesafesi, ilgili standartlarda tanımlanmıştır [38]. Doktora tezinde ortaya konulmuş olan sistemin en temel noktası, robot kol kullanılan iş hücrelerinde robot kolun ulaşabileceği çalışma uzayını kapsayacak şekilde birden fazla kamera görüntüsü alarak bu görüntü içerisine girdiği algılanan yabancı ve istenmeyen nesnelere göre acil durum durması

yapacak bir güvenlik sistemi tasarımıdır. Bunun için güncel standartlar çerçevesinde robot kolun çalışma uzayını ve bu çalışma alanın emniyet katsayısı çarpımı ile güvenlik mesafelerini hesaplamak gerekmektedir. Giriş bölümünde anlatılmış olan standartlara göre Robot kolların yer aldığı iş hücrelerinde Emniyet Mesafesi(S), ilgili standartlarda [38] Denklem 2.1'de görüldüğü gibi tanımlanmıştır.

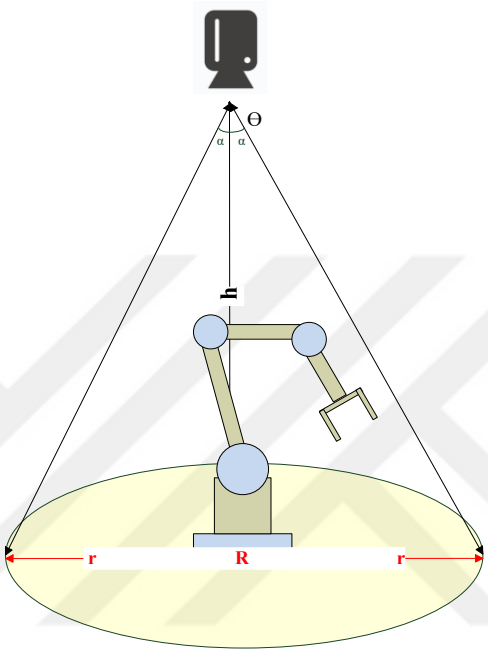
$$S = K * (t1 + t2) + C \quad (2.1)$$

Bu ifadede verilmiş olan parametreler;

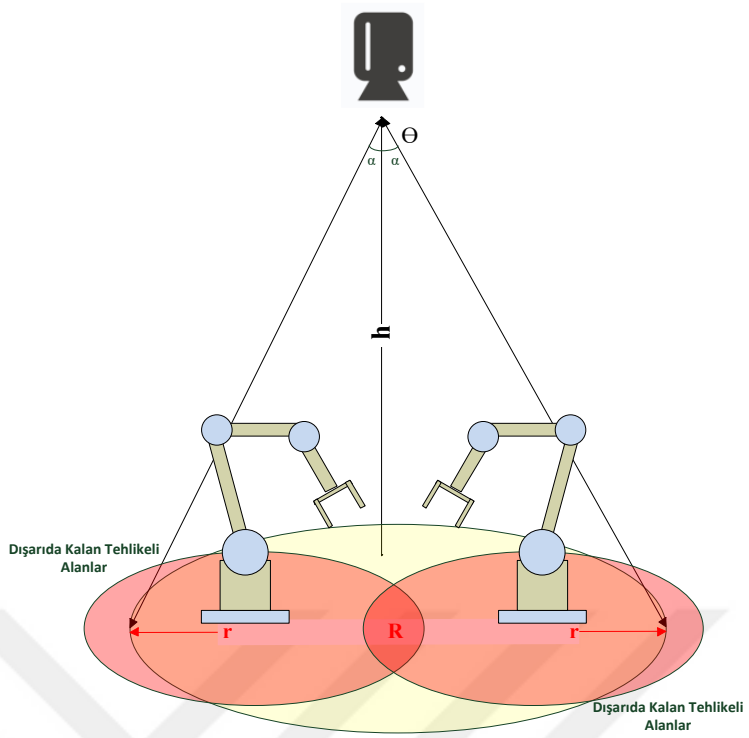
- t1, Tedbirin reaksiyon süresi,
- t2, Makinenin reaksiyon süresi,
- C; Tedbir tarafından saptanmamış tehlike alanına doğru olası yaklaşma,
- K; İnsan vücudunun ya da vücut bölümlerinin tahmin edilen yaklaşma hızıdır. Bu faktör EN ISO 13855 standardında 1.600 mm/sn yürüme hızı ve 2.000 mm/sn el hızı olarak belirlenmiştir.

Doktora çalışmasında tartışılan kameralı güvenlik sisteminin tek bir robot koldan oluşan iş hücresine tek bir kameranın dik bir açı ile bakışının temsili çizimi Şekil 2.2'de görülmektedir. Şekilde yatayla dik bir açı yapan, zeminden h yüksekliğine yerleştirilmiş bir kamera görülmektedir. Robot kol, çalışma alanının ortasına yerleştirilmiştir ve güvenlik kamerasının taradığı alan sarı bir daire ile gösterilmektedir. Bu dairenin yarıçapı r, çapı ise R olarak ifade edilmiştir. Ticari ürünler Şekil 2.2 de görüldüğü gibi tek bir kamerayla ve iş hücresini ortalayacak şekilde dik bir açı ile emniyet mesafesini taranmaktadır. Bu durum robot kolun çalışma esnasında açısal olarak robot kolun altında kalan alanın gölgelenmesine, dolayısıyla da kör bir alan oluşumuna sebep olmaktadır. Bu durum Şekil 2.4'te kırmızı boyalı daireler olarak tehlikeli alanlar şeklinde belirtilmiştir. Birden fazla robot kolun kullanıldığı durumlarda iş hücresinin standartlarda gerektirdiği emniyet mesafesi artacağından daha geniş bir alanı taraması için daha yüksek bir noktada kameranın konumlandırılması gerekmektedir. Dolayısıyla bu durum kameranın çözünürlüğüne olumsuz etki yapacağı gibi iş hücresinin bulunduğu ortamın maksimum yüksekliğinin de kısıtlı olabileceği durumlarda bir problem teşkil etmektedir. Kameralı güvenlik sisteminin iki robot koldan oluşan iş hücresine

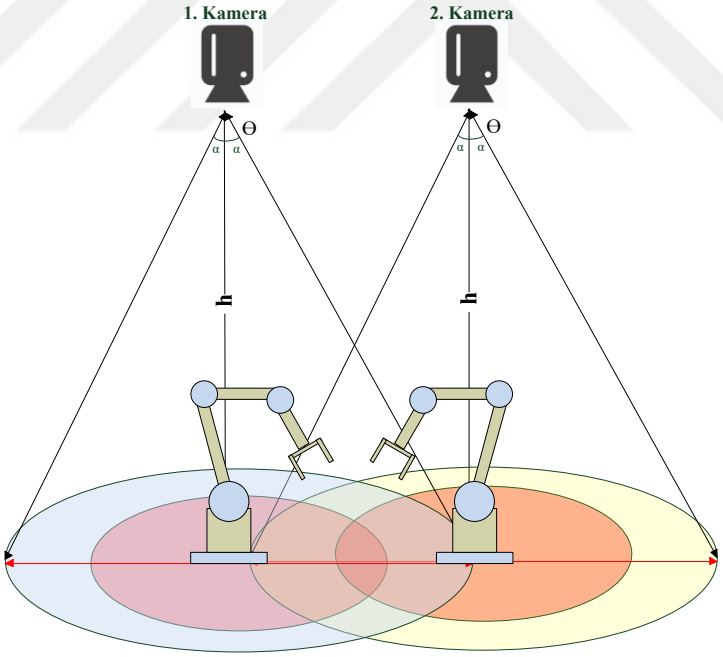
iki kameranın dik bir açı ile bakışının temsili çizimi Şekil 2.4'te görülmektedir. Bu çizimde ise robot kol çalışma alanları kırmızı daireler olarak belirtilmiş ve kameranın taramış olduğu alan standartlarında belirttiği gibi çalışma alanına belli bir emniyet mesafesi bırakacak şekilde daha geniş bir dairesel alan sarı renk ile gösterilmiştir. Böyle bir durum kör nokta olasılığını hala daha barındırmakla beraber maksimum yükseklik kısıtı içinde bir engel teşkil etmektedir.



Şekil 2.2 Bir robot kol ve dik bir açıyla yerleştirilmiş bir kamera ve ihtiva eden sistem görseli

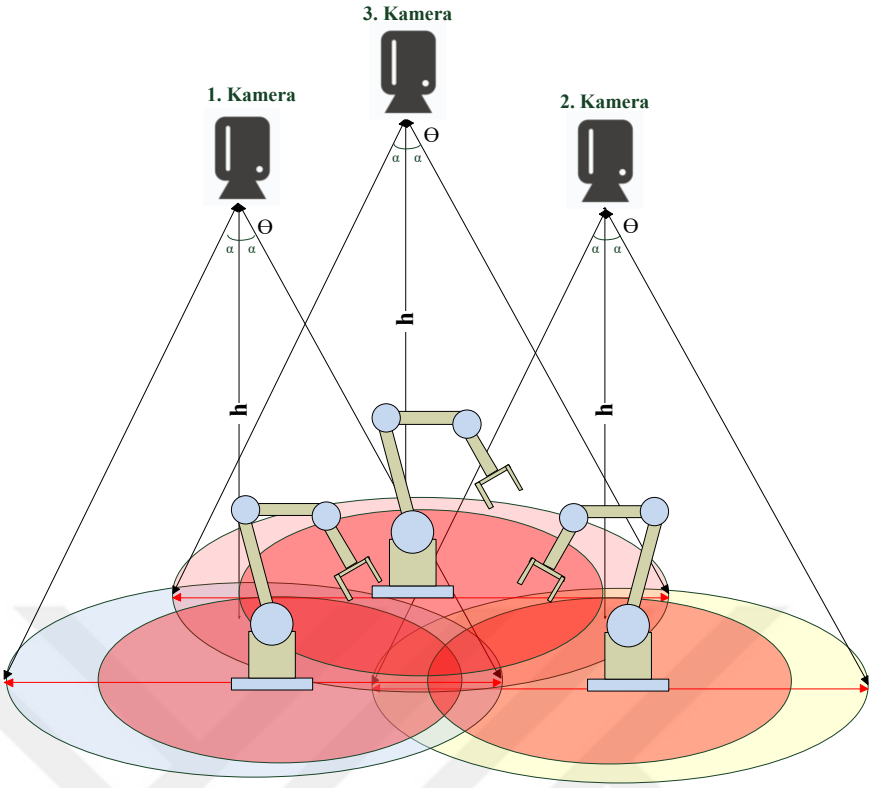


Şekil 2.3 İki robot kol ve dik bir açıyla yerleştirilmiş bir kamera ihtiva eden sistem görseli



Şekil 2.4 İki robot kol ve dik bir açıyla yerleştirilmiş iki kamera ve ihtiva eden sistem görseli

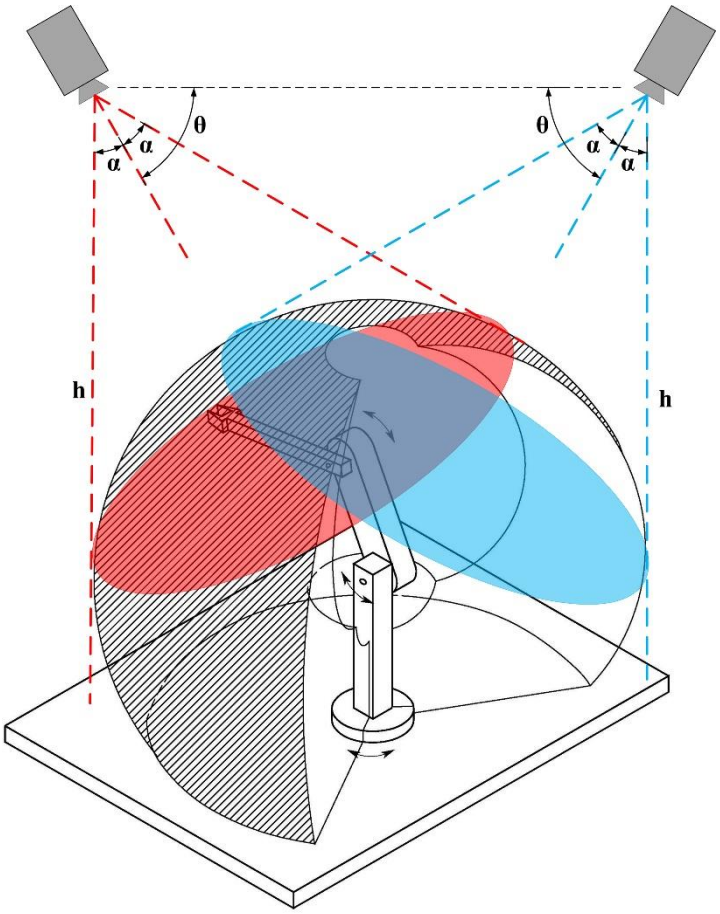
Yukarıdaki şekilde gösterilen durumun sistemin güvenirliliğini düşüreceği bilindiğinden Şekil 2.5'te gösterilmiş olan kamera sayısının arttırılması yapacağımız tasarımda önem arz etmektedir.



Şekil 2.5 Üç robot kol ve dik bir açıyla yerleştirilmiş üç kamera ihtiva eden sistem görseli

Robot kol iş hücresinde kameraların taradığı alanda kör nokta kalmaması için kameraların yatay eksene dik bir açı ile değil daha küçük bir açıyla yerleştirilmesi ve iş hücresine eğimli bir şekilde bakması gerekmektedir. Robot kol sayısına ve boyutlarına bağlı olarak iş hücresi genişleyeceği için kamera sayısının da arttırılması gerekmektedir.

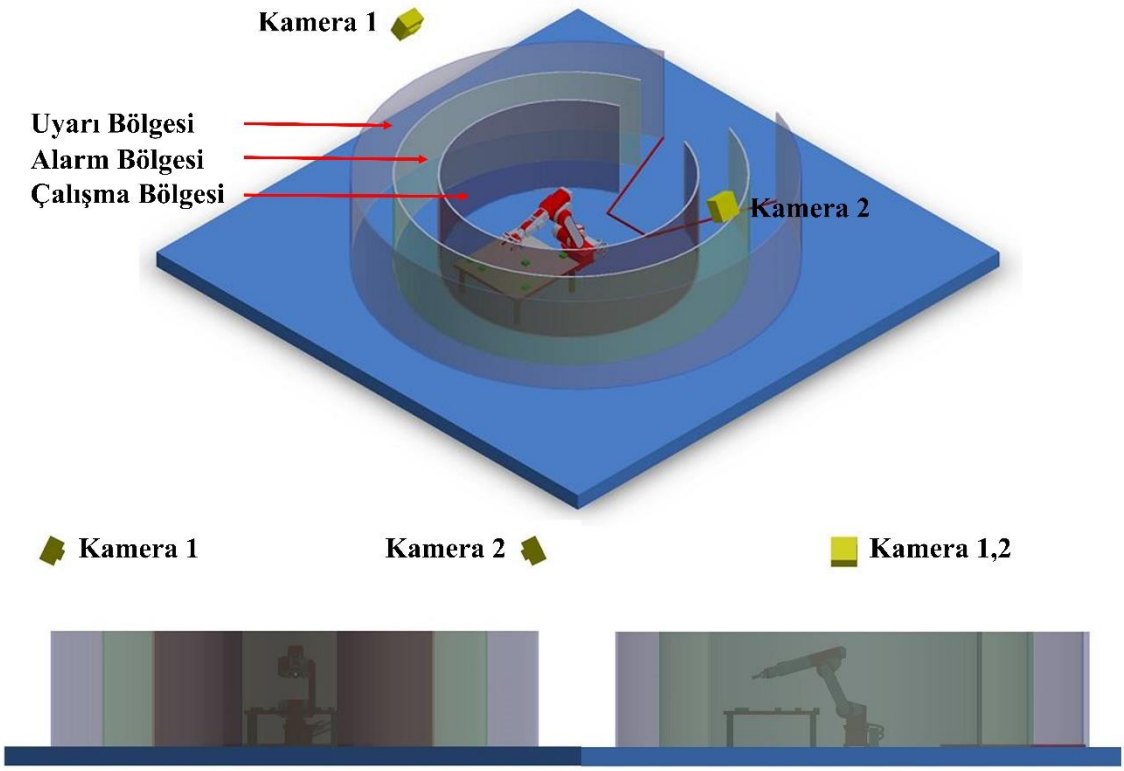
Kullanılacak kameraların konumları, açıları ve sayıları değiştirilmek durumunda kalındığında daha karmaşık bir geometrik yapı ortaya çıkmaktadır. Bu geometrik yapının görseli Şekil 2.6'de görülmektedir. Şekilde iki kamera iş hücresi çalışma alanının dış hattını teğet olarak kesecek şekilde h yüksekliğinde simetrik olarak yerleştirilmiştir. Kameraların yatay eksenle arasındaki açı θ, kameraların lens açısı ise 2α olarak ifade edilmektedir. Kameraların taradığı alanlar mavi ve kırmızı daireler olarak görülmektedir.



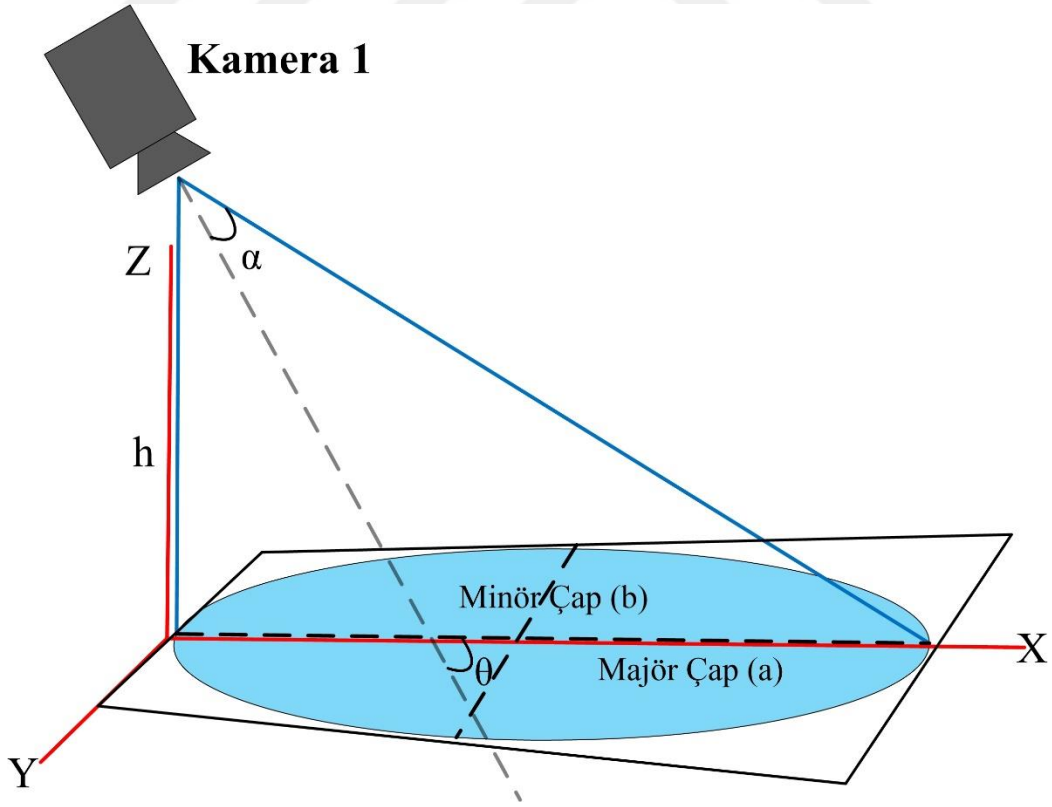
Şekil 2.6 İki kameralı ve açısı değişebilir sistem görseli

Şekil 2.7'da gerçekleştirilmiş olan iki kamera ve bir robottan oluşan sistemin Simülasyon çizimi görülmektedir. Ek olarak, Şekil 2.7 uyarı, alarm ve çalışma bölgeleri olan üç güvenlik alanını göstermektedir. Robot kolunun giriş algılandığında sırasıyla uyarı bölgesinde yavaşlaması, ardından alarm bölgesinde ise durması gerekmektedir

Şekil 2.6 ve Şekil 2.7'de görüldüğü gibi kameraların açılı yerleştirilmesi durumunda kameranın taramış olduğu alan kesik bir koni yapısına benzemektedir. Bu durumda oluşan kamera projeksiyon alanın parametrik olarak ifadesinin görseli Şekil 2.8'de verilmektedir.



Şekil 2.7 İki kamera ve bir robot (üstte) ve önden (sol altta) ve yandan (sağ altta) görünümleri olan bir sistemin çizimi



Şekil 2.8 Kesik konik parametrelerinin şekilsel gösterimi

Yukarıda verilmiş olan şekilde görüldüğü üzere kamera projeksiyon alanı eliptik yapıdadır. Bundan dolayı oluşan geometride bir kısa(minör) çap bir de uzun(majör) çap oluşmaktadır. Bu noktada yukarıdaki şekilde görüldüğü gibi α, kamera lens açıklığının yarısını ifade etmektedir. θ, ise kameranın yerleşimine bağlı olarak yerleştirileceği yatay düzlem ile lensin ortasından geçen normali ile yaptığı açıyı ifade etmektedir. h, parametresi kameranın lensi ile taranmak istenen alan arasındaki dik yüksekliği ifade etmektedir. Değişkenlerimiz olan bu parametrelere bağlı olarak majör yarıçapı a, denklem 2.2'de verilmektedir [39].

$$a = \frac{hsin2\alpha}{2(sin^2\theta - sin^2\alpha)} \tag{2.2}$$

Minör yarıçapı b, ise denklem 2.3'te verilmektedir.

$$b = \frac{hsin\alpha}{\sqrt{(sin^2\theta - sin^2\alpha)}} \tag{2.3}$$

Tüm durumlar ele alındığında çözümü yapılmış olan problemin matematiksel modeli şu şekilde özetlenebilir; Amaç taranacak alanın maksimize edilmesi ve h yüksekliğinin minimize edilmesidir. Bunun için değişkenler;

- α, kamera lens açıklığının yarısını ifade etmektedir.
- θ, ise kameranın yerleşimine bağlı olarak yerleştirileceği yatay düzlem ile lensin ortasından geçen normali ile yaptığı açıyı ifade etmektedir.
- h, parametresi kameranın lensi ile taranmak istenen alan arasındaki dik yüksekliği ifade etmektedir.

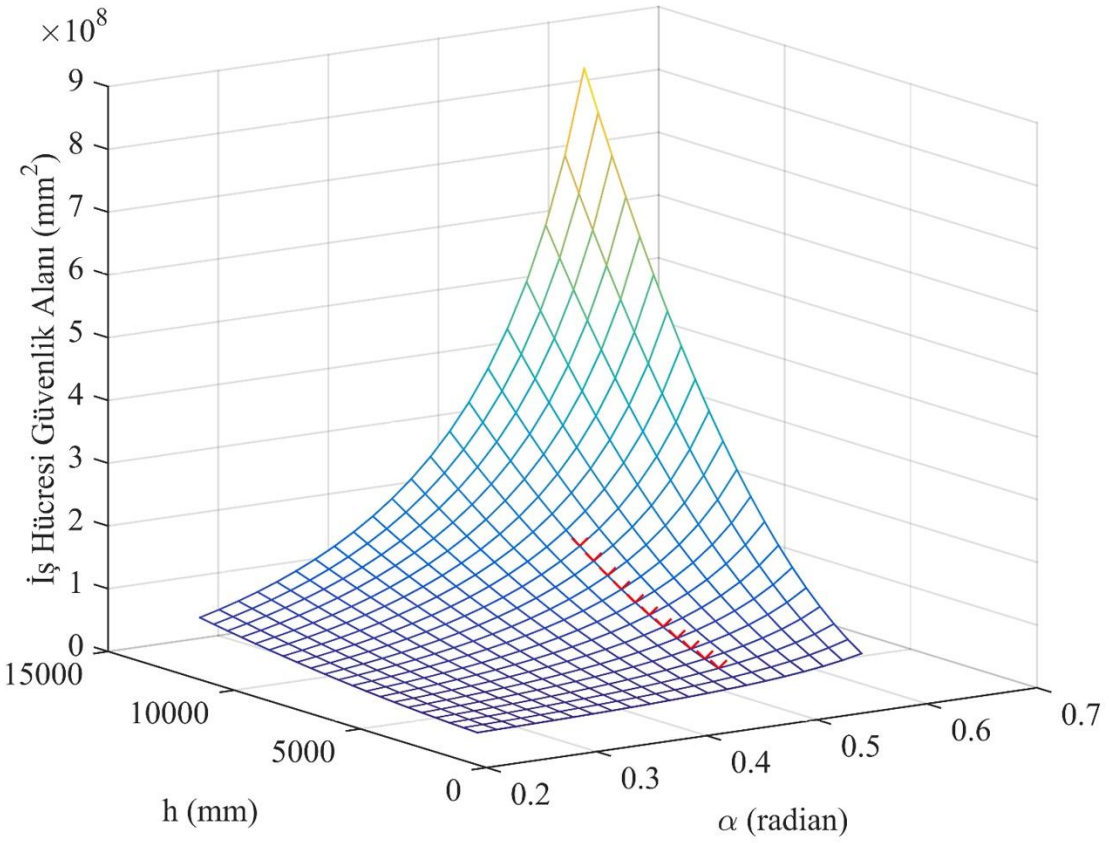
Değişken sayısının azaltılması amaçlandığında, θ, açısı kamera normalinin yatay eksenle yaptığı açı olduğu için içerisinde α açısını da ihtiva etmektedir. Dolayısıyla denklem 2.8'de görüleceği gibi θ açısını, α açısı cinsinden ifade edilebilmektedir.

$$\theta = (90° - \alpha) = (\frac{\pi}{2} - \alpha) \tag{2.4}$$

Denklem 2.2 ve 2.3'te kamera projeksiyon geometrisinin kısa ve uzun kenar yarıçapları verilmektedir. Dolayısıyla Denklem 2.5'te taranacak olan kamera projeksiyon alanı yukarıda verilen değişkenler cinsinden ifade edilebilir.

$$Alan = \frac{\pi h^2 sin(\alpha) sin(2\alpha)}{2(cos^2(\alpha) - sin^2(\alpha))^{3/2}} \quad (2.5)$$

Denklem 2.5'te verilen alan fonksiyonu maksimize edilmesi gereken hedef fonksiyonu olarak seçilmiştir. Bu fonksiyonun 3 boyutlu grafiği Şekil 2.9'da görülmektedir. Şekil amaç fonksiyonunun verilen kısıtlamalar için taradığı alanın grafiksel temsilini göstermektedir. Bu grafikte güvenlik bölgesinin alanı h ve α ile doğru orantılıdır. Dolayısıyla grafikte gösterilen sarı alan güvenlik bölgesi için en yüksek alana sahip olsa da yükseklik kısıtlamaları nedeniyle optimal noktanın bu bölgede aranması gerektiği görülmektedir. Bu nedenle kamera merceğinin optimum açısı ve 4000 ila 10.000 mm arasındaki h yükseklik değerleri için en uygun bölge kırmızı renkle işaretlenmiştir.



Şekil 2.9 Amaç fonksiyonunun üç boyutlu grafiği

2.2 En Uygun Çözümün Bulunması (Optimizasyon)

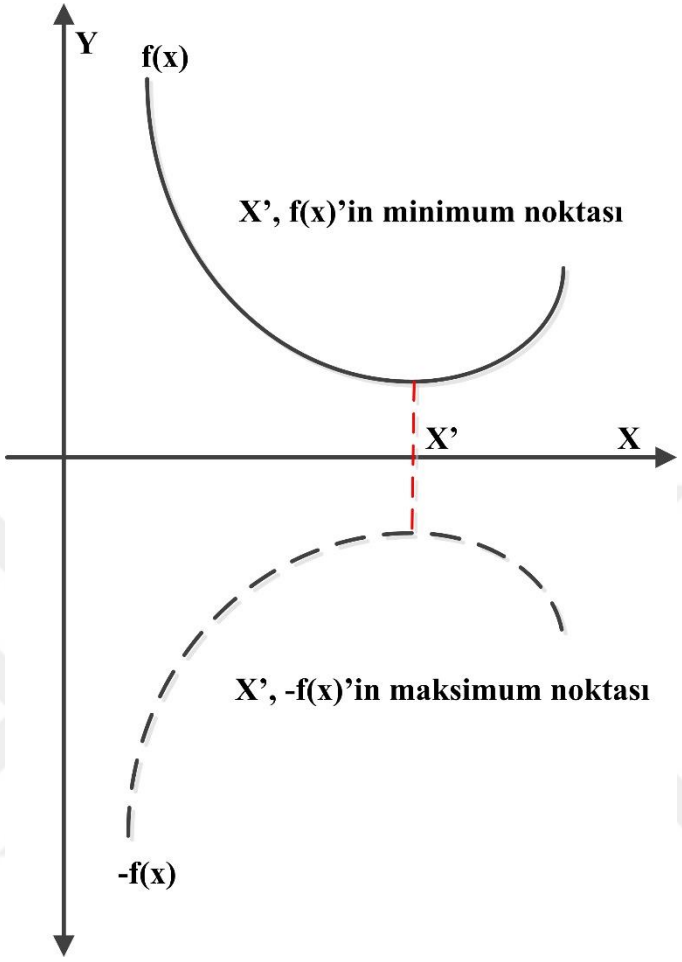
Optimizasyon, verilen koşullar altında en iyi sonucu elde etme eylemidir. Herhangi bir mühendislik sisteminin tasarımında, yapımında ve bakımında, mühendisler çeşitli aşamalarda birçok teknolojik ve yönetimsel karar almak zorundadır. Tüm bu kararların nihai hedefi ya gerekli çabayı en aza indirmek ya da istenen faydayı en üst düzeye çıkarmaktır. Herhangi bir pratik durumda gerekli olan çaba veya fayda istenen karar değişkenlerinin bir fonksiyonu olarak ifade edilebildiğinden, optimizasyon, bir fonksiyonun maksimum veya minimum değerini veren koşulları belirleme süreci olarak tanımlanabilir.

Şekil 2.10' da görülebileceği gibi, X' değeri f(x) fonksiyonunun minimum değerine karşılık gelirse, aynı nokta aynı zamanda fonksiyonun negatif değerinin −f(x) maksimum değerine de karşılık gelir. Bu nedenle, genellik kaybı olmaksızın, aynı işlevin negatifinin minimumunu arayarak bir fonksiyonun maksimum bulunabileceğinden, optimizasyon en aza indirgemek için alınabilir.

Tüm optimizasyon problemlerini çözmek için tek bir yöntem mevcut değildir. Bu nedenle, farklı optimizasyon problemlerinin çözümü için bir dizi optimizasyon yöntemi geliştirilmiştir. Optimum arama yöntemleri, matematiksel programlama teknikleri olarak da bilinmektedir. Yöneylem araştırması, karar verme problemlerine bilimsel yöntem ve tekniklerin uygulanması ve en iyi veya optimal çözümlerin oluşturulması ile ilgili bir matematik dalıdır. Matematiksel programlama teknikleri, belirlenmiş bir dizi kısıtlama altında birkaç değişkenin bir fonksiyonunun minimumunu belirlemede kullanılmaktadır. Stokastik süreç teknikleri, bilinen olasılık dağılımlarına sahip bir dizi rasgele değişken tarafından tarif edilen problemleri analiz etmek için kullanılabilir. İstatistiksel yöntemler, deneysel verileri analiz etmeyi ve fiziksel durumun en doğru temsilini elde etmek için ampirik modeller oluşturmayı mümkün kılmaktadır [40].

Birçok mühendislik sisteminin tasarımı oldukça karmaşık bir süreç olabilmektedir. Mevcut yöntemlerle analize tabi tutulabilen modeller geliştirmek için birçok varsayım yapılabilir ve modeller deneylerle doğrulanabilir. Problem formülasyonu aşamasında birçok olasılık ve faktör dikkate alınmaktadır. Çoğu uygulama için tüm tasarım projesi, daha sonra bağımsız olarak ele alınabilen birkaç alt probleme

ayrılmaktadır. Alt problemlerin her biri, optimum tasarım problemi olarak ortaya konulmaktadır [41].



Şekil 2.10 f(x) foksiyonunun minimum noktası ve –f(x) fonksiyonunun maksimum noktası [40]

Bir optimizasyon probleminde ilk adım, sistem için kesin özelliklerin belirlenmesidir. Bunlar belirlendikten sonra, sistemin tasarlanması görevi başlayabilir. Sürecin ikinci adımı, sistemin bir ön tasarımını geliştirmektir. Sistem için çeşitli kavramlar çalışılmaktadır. Bunun nispeten kısa bir sürede yapılması gerektiğinden, basitleştirilmiş modeller kullanılmaktadır. Bunun için alt başlıklarda doktora çalışmasında ele alınmış olan problem ve çözümü için yaklaşımlar sunulmaktadır.

IP kameralar robotik çalışma hücrelerinde güvenliği sağlamak için son birkaç on yıldır kullanılmaktadır. Bunlardan bazıları vücut duruşu ve pozisyonunu ölçerek insan operatörünün güvenlik izleme sistemine dayanmaktadır [42]. Diğerleri robot

kolu uç efektörünün konumunu ve hızını izlemekle ilgilidir [43]. Bununla birlikte, istenmeyen nesnelerin görsel olarak izlenebildiği sistemler de literatürde meevcuttur [44]. Bu sistemlerin farkındalığı ve kamera teknolojisinin gelişmesi ile 3D kamera teknolojisini kullanan sistemler de görülmektedir [45]. Bu sistemlerin geldiği noktada, çalışma hücrelerinde kör noktaların kalması durumunda birden fazla kamera kullanılması istendiğinde kullanılacak en uygun kamera sayısını bulmak için bir çalışma olmadığı görülmektedir. Bu nedenle, bu yüksek fiyatlı üç boyutlu kameralar yerine, daha uygun fiyatlı iki boyutlu kameralar kullanarak robot kolunun sayısına ve boyutuna uyum sağlayabilen bir optimizasyon problemi öngörülmüştür. Bu sisteme eklenecek robot kollarının sayıları, konumları ve sayısı optimizasyon değişkenleri olarak belirlenebilir. Ancak, farklı çözünürlükteki kameraları kullanma olasılığına rağmen, kamera çözünürlüğü de hesaplama sonuçlarını etkilemektedir.

2.2.1 Optimizasyon Kısıtları

Bir önceki bölümde optimizasyon problemi için bir hedef fonksiyon elde edilmiştir. Problemde yer alan değişkenlerin birtakım sınırlamalara sahip olması gerektiği görülmektedir. Aksi halde sonuç sonsuza gidecektir. Bu sınırlamalar, optimizasyon probleminin kısıtları olarak adlandırılmaktadır. Bu noktada kısıt seçimi olarak öncelikle endüstriyel alanda kullanılan gerçek bir robot kolun ölçülerine uymak elde edilecek sonuçların karşılaştırmasını yapmak açısından doğru bir yaklaşım olduğu düşünülmektedir. Dolayısıyla güncel robot kollar araştırıldığında emniyet mesafesi de eklenmiş hali dikkate alınarak çalışma uzayı kısa kenar yarıçapı(b), minimum 3000 milimetre olarak seçilebilir. Aynı şekilde robot kol çalışma alanı çevresinde kalan çalışma alanının dışında fakat, kontrol altında olması gereken alan için bu yarıçap 4000 milimetre seçilebilir. Sonuç olarak birinci kısıt için Denklem 2.3'te verilen fonksiyonun bu kısıtlar arasında kalması istenmektedir.

Benzer sebeplerle sistemin kurulacağı çevresel şartların önemli olduğu bilinmektedir. Dolayısıyla kameranın yerleştirileceği nokta ve zemin arasında bir yükseklik değeri belirlenmelidir. Aksi halde kamera ne kadar yükseğe konumlandırılırsa tarayacağı alan o kadar büyüyecek ve çözüm sonsuza gitme eğiliminde olacaktır. Dolayısıyla bir üretim tesisi için bu yükseklik sonradan

değiştirilebilir şekilde 10,000mm olarak belirlenmiştir. Sonuç olarak ikinci optimizasyon kısıtı, 0 ve 10m arasında bir yükseklikte olacak şekilde belirlenmiştir.

Bir diğer kısıt ise hedef fonksiyon ile doğrudan ilişkisi olması sebebiyle, her zaman en büyük değeri alabilecek fakat, piyasa şartlarında en büyük değerin bulunamayacağı kamera lens açısı değeri ile ilgilidir. Kamera lens açısının piyasada bulunabilen modeller arasında bir kısıt seçilmesi halinde sistemin gerçeklenmesi esnasında birebir karşılaştırmalar yapılmasını kolaylaştıracaktır. Sonuç olarak, üçüncü kısıt olarak kamera lens açısının 30 derece ve 70 derece arasında olması beklenmektedir.

2.2.2 Kamera Çözünürlüğünün Optimizasyon Kısıtlarına Etkisi

Görüntü çözünürlüğü, görüntünün sahip olduğu ayrıntıdır. Bu terim, dijital görüntüler, film görüntüleri ve diğer görüntü türleri için geçerlidir. Daha yüksek çözünürlük daha fazla görüntü detayı anlamına gelmektedir. Görüntü çözünürlüğü çeşitli şekillerde ölçülebilir. Çözünürlük, çizgilerin birbirine ne kadar yakın olabileceğini ve hala görünür şekilde çözülebileceğini belirlemektedir. Çözünürlük birimleri fiziksel boyutlara, bir resmin genel boyutuna veya açısal uzatmaya bağlıdır. Dijital kameraların çözünürlüğü birçok farklı şekilde tanımlanabilmektedir. Bunlar;

- Piksel çözünürlüğü
- Mekânsal çözünürlük
- Spektral (tayfla ilgili) çözünürlük
- Zamansal çözünürlük

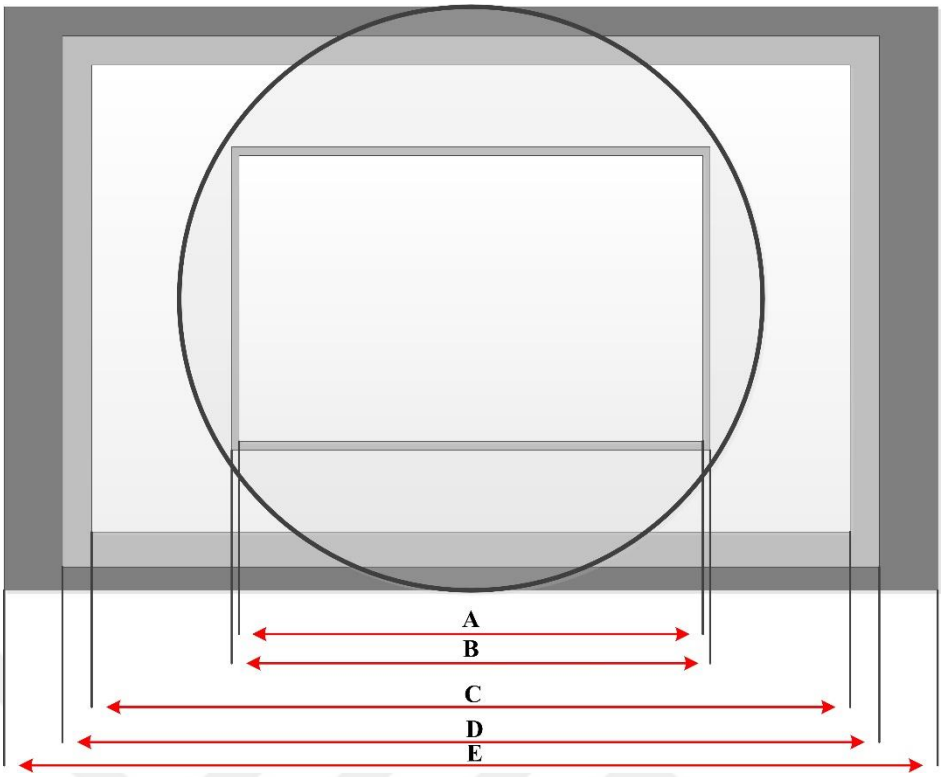
Bu çözünürlük çeşitleri arasından yapılan doktora çalışmasıyla ilgili olan kısım görüntü alanının içinde yer alan objelerin yakalanması olduğu için ve bu durumun kamera yerleşim konumu uzaklığıyla orantılı olarak değişken kabul edildiğinden dolayı piksel çözünürlüğü üzerinde durulmuştur. Her bir piksele tekabül eden gerçek boyut, kamera bakış uzaklığıyla değişeceğinden taranması gereken piksel sayısı da bu uzaklığa bağlı olarak ve kamera çözünürlüğüne bağlı olarak değişmektedir.

Piksel çözünürlüğü; Dijital kamera alanındaki uluslararası standartlar çözünürlük yerine görüntü sensörleriyle ilgili olarak "Toplam Piksel Sayısı" olarak adlandırılması gerektiğini ve bunun için "Kaydedilen Piksel Sayısı" olarak adlandırılmasını belirtmesine rağmen, çözünürlük terimi genellikle dijital görüntülemedeki piksel sayısına eşdeğer kabul edilmektedir [46]. Aynı standartlara göre, bir görüntü sensörünün veya dijital kameranın sahip olduğu "Etkili Piksel Sayısı", son görüntüye katkıda bulunan piksel sayısıdır. Kenarlarda kullanılmayan veya ışık korumalı pikseller içeren toplam piksel sayısı dikkate alınmamaktadır.

N piksel yüksekliğinde ve M piksel genişliğinde bir görüntü, resim yüksekliği başına N satırından veya televizyon satırından daha düşük bir çözünürlüğe sahip olabilir. Ancak piksel sayılarına "çözünürlük" adı verildiğinde, kural piksel çözünürlüğünü iki pozitif tamsayı kümesi ile tanımlamaktır, burada ilk sayı piksel sütunlarının sayısı (genişlik) ve ikincisi piksel sıraları (yükseklik), örneğin 7680x6876 gibi. Bir başka popüler kural, çözünürlüğü görüntüdeki piksel sayısı piksel satırlarıyla çarparak ve bölünerek hesaplanabilen toplam megapiksel sayısı olarak verilen toplam piksel sayısı olarak belirtmektir. Diğer kurallar arasında uzunluk birimi başına piksel sayısı veya inç başına veya inç kare başına piksel gibi alan birimi başına piksel sayısı açıklanmaktadır. Bu piksel çözünürlüklerinin hiçbiri gerçek çözünürlük değildir, ancak yaygın olarak bu şekilde adlandırılmaktadır [47].

Örneğin, genişliği 2048 piksel, yüksekliği 1536 piksel olan bir resim toplam 2048 × 1536 = 3.145.728 piksel veya 3.1 megapikseldir. Buna 2048x1536 veya 3.1 megapiksel görüntü denilebilir. Dolayısıyla kamera seçiminde kamera çözünürlüğünün anlamları bu şekilde değelendirilmelidir. Aşağıda yer alan şekil 2.11'de piksel ve çözünürlük arasındaki ilişki gösterilmektedir.

Şekil 2.11'deki görselde verilmekte olan parametreler; A: Etkili piksel sayısı (Halka pikselsiz görüntü alanı), B: Etkili piksel sayısı (Halka piksellerle), C: Maksimum kullanılabilir piksel alanı (Optoelektrik dönüşüm elemanı satıcısı tarafından garanti edilen pikseller), D: Görüntü sensöründe etkili piksel, E: Toplam piksel sayısı olarak ifade edilmektedir.



Şekil 2.11 Piksel ve çözünürlük arasındaki ilişkinin benzetimi [46]

Etkili Piksel Sayısı A (halka pikselsiz) ile Kaydedilen Pikseller ve Çıktı Piksel Sayısı aşağıdaki gibidir;

- A=Kaydedilen piksel sayısı ve çıkış piksel sayısı (enterpolasyon veya kırpma olmadan)
- A<Kaydedilen piksel sayısı ve çıktı piksel sayısı (enterpolasyonlu)
- A>Kaydedilen piksel sayısı ve çıktı piksel sayısı (kırpma ile)[46]

Sonuç olarak, robot kolun çalışma alanı taranmak istediğinde, bu alanın tam bir çember oluşturacağı varsayılır. Bu varsayıma dayanarak robot güvenlik alanı da dâhil olan bu çember çapı kamera etkili piksel alanının kısa kenarı ile eşitlenir. Bu eşitleme ile çalışma alanının çapının mesafesinin gerçek değeri bilindiğinden kamera çözünürlüğündeki piksel sayısına bölündüğünde her bir pikselin taradığı uzunluk bilinmektedir. Dolayısıyla bu noktada görüntü mesafesi sabit tutulduğu sürece yakalanmak istenen objenin boyutunun kaç piksele denk geleceği bilinebilmektedir.

Örneğin VGA çözünürlüğündeki bir kameradan görüntü aldığımız düşünülüyorsa, bu kameranın 640x480 boyutunda piksele sahip olduğu düşünülebilir. Robot kol

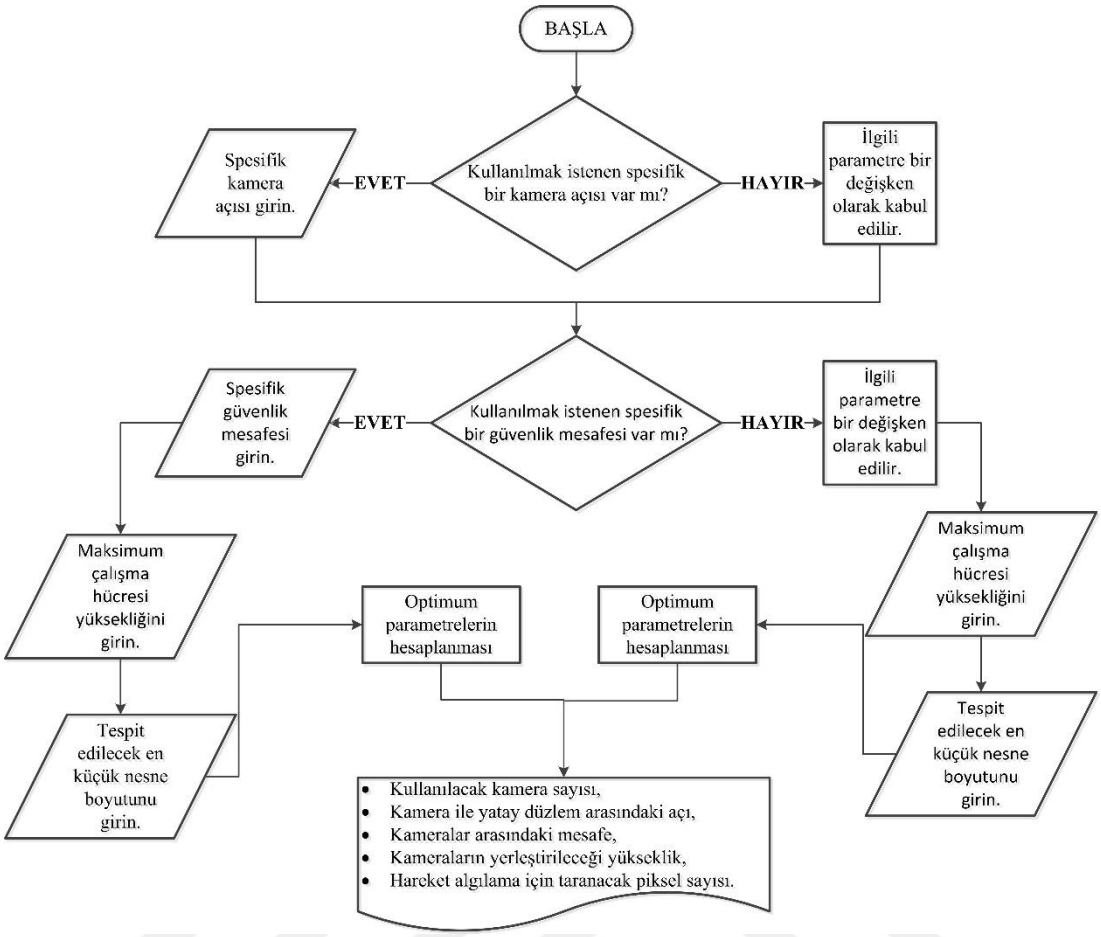
emniyet alanı da dâhil olmak üzere çalışma alan çapı 1000mm olduğu düşünülürse, bu çapın kamera çözünürlüğü kısa kenara bölümünden her bir pikselin 2,08mm olduğu ortaya çıkmaktadır. Güvenlik tedbirlerinde tehdit yaratacak objenin çapı bu piksel değerine bölünüp, çıkan küsürat yukarı doğru en yakın tam sayıya yuvarlanır ve karesi alınır. Çıkan sonuç, objenin kaç piksel kapladığını bize sunmaktadır. 5mm çapa sahip bir obje yakalanmak istendiğinde bu objenin 9 pikselde tarandığı ortaya çıkmaktadır.

3
SİSTEM TASARIMI

3.1 Matematiksel Modelin Uygulaması

Giriş bölümünde tanımlanmış yöntem, çözülmesi gereken bir endüstriyel problem olarak yapılandırıldığında, yöntemin uygulanması gereken problem şu şekilde özetlenebilir; Endüstriyel alanda robot kol kullanılan iş hücrelerinde kameralı güvenlik sistemlerinin kullanıldığı bilinmektedir. Bu güvenlik sistemleri görüntü işlemeye dayalı olarak iki boyutlu veya üç boyutlu olarak kullanılabilmektedir, fakat, kör nokta oluşması açısından tek kameranın yeterli olmadığı durumlar ortaya çıkmaktadır. Literatürde birden fazla kamera kullanılan uygulamalarda, bu kameraların mevcut çalışma alanı kısıtları olan, kameranın lens açısına ve çözünürlüğüne göre, en uygun konum ve yerleşimlerinin bulunmasına dair bir çalışma yoktur. Dolayısıyla güvenlik sistemi, kamera sayısı, kamera lens açısı, kamera çözünürlüğü ve kamera yerleşim açısının en uygun olduğu durumu bulmaya dair bir optimizasyon problemi olarak ele alınmıştır. Giriş bölümünde bahsedilen problem yöntem ve kısıtlarıyla beraber ele alındığında, değişkenlere bağlı olarak matematiksel model olarak türetilmiştir..

Görüntü tabanlı güvenlik sistemi için tüm kritik değişkenleri içeren genel ve dinamik bir matematiksel modele ihtiyaç duyulmaktadır. Bu sistem modelinin sektördeki çeşitli durumlara ve kısıtlamalara pratik olarak uyum sağlaması beklenmektedir. Kamera merceği açısı ve algılanacak minimum nesne boyutu ana girdi değişkenleridir. Güvenlik mesafesi çalışma hücresinin boyutlarına bağlı olarak bağımsız bir değişken olabilir veya olmayabilir. Maksimum kamera yüksekliği ana kısıtlamadır. Önerilen güvenlik sisteminin dinamik modelinin akış şeması Şekil 3.1'de verilmiştir. Bu nispeten basit ama etkili algoritma, kameraların sayısı, yerleşimi, kameraların açılarını ve iş hücresinde taranacak piksel sayısını içeren sistem tasarım parametrelerini vermektedir.



Şekil 3.1 Sistem tasarımını parametrelerini belirleyen optimizasyon algoritmasının akış şeması

Denklem 2.6'da yer alan hedef fonksiyonu elde edildikten sonra, optimizasyon kısıtları olarak, güncel robot kollar araştırıldığında emniyet mesafesi de eklenmiş hali dikkate alınarak çalışma uzayı kısa kenar yarı çapı(b), minimum 3000 milimetre olarak seçilmiştir. Aynı şekilde robot kol çalışma alanı çevresinde kalan çalışma alanının dışında fakat, kontrol altında olması gereken alan için bu yarıçap 4000 milimetre seçilmiştir. Sonuç olarak birinci kısıt için Denklem 2.3'te verilen fonksiyonun bu kısıtlar arasında kalması beklenmektedir. Benzer sebeplerle sistemin kurulacağı çevresel şartların önemli olduğu bilinmektedir. Dolayısıyla kameranın yerleştirileceği nokta ve zemin arasında bir yükseklik değeri belirlenmelidir. Aksi halde kamera ne kadar yükseğe konumlandırılırsa tarayacağı alan o kadar büyüyecek ve çözüm sonsuza gitme eğiliminde olacaktır. Dolayısıyla bir üretim tesisi için bu yükseklik sonradan değiştirilebilir şekilde 10,000mm olarak belirlenmiştir. Sonuç olarak ikinci optimizasyon kısıtı, 0 ve 10m arasında bir

yükseklikte olacak şekilde belirlenmiştir. Bir diğer kısıt ise hedef fonksiyon ile doğrudan ilişkisi olması sebebiyle, kamera lens açısı değeri ile ilgilidir. Ancak, her zaman en büyük değeri alabilecek fakat, piyasa şartlarında en büyük değere sahip olan kameranın bulunamayabileceği göz önünde bulundurulmalıdır. Kamera lens açısının piyasada bulunabilen modeller arasında bir kısıt seçilmesi halinde sistemin gerçeklenmesi esnasında birebir karşılaştırmalar yapılmasını kolaylaştırmaktadır. Sonuç olarak, üçüncü kısıt olarak kamera lens açısının 30 derece ve 70 derece arasında olması beklenmektedir.

İki adet kamera kullanılması planlanan senaryoda, bir fonksiyonun maksimum olduğu nokta türevin sıfır olduğu yer olduğundan, denklem 2.6'da verilmiş olan alan fonksiyonu, f(α,h) şeklinde iki değişkene bağlı olarak tanımlanabilir. Bu durum denklem 3.1'de görülmektedir. Sonrasında ise sırasıyla bu fonksiyonun α'ya ve h'ye göre türevleri alınmıştır.

$$f(\alpha, \mathrm{h}) = \frac{\pi \mathrm{h}^2 \sin(\alpha)\sin(2\alpha)}{2(cos^2(\alpha) - sin^2(\alpha))^{3/2}} \tag{3.1}$$

Denklem 3.1'de verilen fonksiyonun h değişkene göre türevi alındığında, denklem 3.2'de verilen fonksiyon elde edilmektedir.

$$\frac{\partial f(\alpha, \mathrm{h})}{\partial \mathrm{h}} = \frac{2\pi \mathrm{h} sin^2(\alpha)\cos(\alpha)}{cos^{2/3}(2\alpha)} \tag{3.2}$$

Denklem 3.2'de elde edilen fonksiyonun h ifadesi yani kamera yüksekliği için çözüm kümesi 0 olmaktadır. Denklem 3.3'de hedef fonksiyonun α değişkene göre türevi alındığında oluşan durumu verilmektedir.

$$\begin{aligned} \frac{\partial f(\alpha, \mathrm{h})}{\partial \alpha} &= \frac{\pi \mathrm{h}^2 \sin(2\alpha)\cos(\alpha)}{2(cos^2(\alpha) - sin^2(\alpha))^{3/2}} + \frac{3\pi \mathrm{h}^2 sin^2(\alpha)\sin(2\alpha)\cos(\alpha)}{(cos^2(\alpha) - sin^2(\alpha))^{5/2}} \\ &+ \frac{\pi \mathrm{h}^2 \sin(\alpha)\cos(2\alpha)}{(cos^2(\alpha) - sin^2(\alpha))^{3/2}} = \frac{\pi \mathrm{h}^2 \sin(\alpha)(\cos(2\alpha) + 3)}{2cos^{5/2}(2\alpha)} \end{aligned} \tag{3.3}$$

Denklem 3.3'te elde edilen fonksiyonun çözüm kümesi denklem 3.4'de verilmektedir.

$$\alpha_1 = \pi n, n \in Z$$

$$\alpha_2 = \pi n - \frac{1}{2} cos^{-1}(-3), n \in Z \tag{3.4}$$

$$\alpha_3 = \pi n + \frac{1}{2} cos^{-1}(-3), n \in Z$$

Denklem 3.4'te, α değerleri için çözüm kümesi verilmektedir. 3 elemanın yer aldığı bu çözüm kümesinde, birinci çözüm kümesi elemanı için α_1, 0 değerini almaktadır. Üçüncü çözüm kümesi elemanı için ise α_3, negatif değer alacağından, ikinci çözüm kümesi elemanı, α_2 dikkate alınacaktır. Denklem 3.5'te α_2 ifadesinin değeri görülmektedir.

$$\alpha_2 = 0.50505 radyan = 28{,}937° \tag{3.5}$$

Denklem 3.5'te elde edilen α_2 ifadesi için kamera lens açısının elimizdeki kısıtlara göre alabileceği en uygun değer denklem 3.6'da görülmektedir.

$$2.\ Kameralı\ Sistem\ için\ En\ Uygun\ Kamera\ Lens\ Açısı = 1.0101 radyan = 57{,}874° \tag{3.6}$$

Verilen kısıtlar çerçevesinde α açısının 15 ve 35 derece arasında, h yükseklik değerlerinin alabileceği alt ve üst sınırlar hesaplanmıştır. Hesaplanan sınır değerleri Tablo 3.1' de verilmektedir.

Tablo 3.1 İki Kameralı Sistemde Kamera Yüksekliğinin, Lens Açısına Bağlı Olarak Sınır Değerleri

α (°)	α (radyan)	h için alt sınır(mm)	h için üst sınır(mm)
15	0.2618	10787	14382
16	0.27925	10023	13364
17	0.29671	9342,7	12457
18	0.31417	8732,1	11643
19	0.33163	8179,9	10906
20	0.34909	7677,1	10236
21	0.36655	7216,5	9622
22	0.38401	6792,2	9056,3
23	0.40147	6399,2	8532,3
24	0.41893	6033,4	8044,6
25	0.43639	5691,2	7588,3
26	0.45385	5369,7	7159,6
27	0.47131	5066,2	6755
28	0.48877	4778,5	6371,4
28,937	0,50505	4521,5	6028,6
29	0.50623	4504,6	6006,1
30	0.52369	4242,6	5656,9
31	0.54115	3991	5321,4
32	0.55861	3748,3	4997,7
33	0.57607	3512,9	4683,9
34	0.59353	3283,6	4378,1
35	0.61099	3058,8	4078,4

Denklem 3.5'te bulunan açı değerini Tablo 3.1'de yerine koyulduğunda, h yüksekliğinin üst sınırını 6028.6mm olarak bulunmaktadır. Bu değerleri hedef fonksiyonda yerine koyulduğunda, bir kameranın taradığı maksimum alanı 33,9335 m^2 olarak bulunmaktadır.

Tablo 3.1'de kamera lens açısının küçüldüğü durumlarda h yüksekliğinin alt sınırının dahi 10m yükseklikten fazla olduğu görülmektedir. Bu durumdan eğer iş hücresinin tasarımı dar açılı kameralar ile yapılmak istenirse veya 10m yükseklik sınırı daha aşağıda kullanılmak istenirse, bu sistem için iki kameranın yetmediğini ve kamera sayısının arttırılması gerektiği anlaşılmaktadır. Dolayısıyla kamera sayısının arttırılması kısıtlarda değişikliğe gidilmesine sebep olmaktadır. Bu durumda, Denklem 2.6'da ele alınan hedef fonksiyonu aynı kalmakla beraber, optimizasyon kısıtları olarak, iki kameradan fazla kamera kullanılması durumunda, denklem 2.2'de verilmiş olan, çalışma uzayı uzun kenar yarı çapı(a), minimum 3000 milimetre olarak seçilmiştir. Aynı şekilde robot kol çalışma alanı çevresinde kalan çalışma alanının dışında fakat, kontrol altında olması gereken alan için bu yarıçap 4000 milimetre seçilmiştir. Dolayısıyla denklem 3.7'de yeni kısıt değeri bir eşitsizlik olarak görülmektedir.

$$3000\text{mm} \leq \frac{hsin2\alpha}{2(cos^2\alpha - sin^2\alpha)} \leq 4000\text{mm} \tag{3.7}$$

Denklem 3.8'de, denklem 3.7'de verilmiş olan eşitsizliğin sadeleştirilmiş hali yer almaktadır.

$$6000\text{mm} \leq htan(2\alpha) \leq 8000\text{mm} \tag{3.8}$$

Denklem 3.8'de, verilmiş olan iki bilinmeyenli eşitsizlikten, bilinmeyenleri ayırdığımızda aşağıdaki denklemler elde edilmektedir. α için $n \in Z$ olmak üzere, eşitsizlik denklem 3.9'da görülmektedir.

$$\pi n \leq \alpha \leq \pi n + \frac{\pi}{4} \tag{3.9}$$

h için $n \in Z$ olmak üzere, eşitsizlik denklem 3.10'da görülmektedir.

$$h \geq \frac{1500}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} + \frac{1500cot^4(0{,}5\alpha)}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} - \frac{9000cot^2(0{,}5\alpha)}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} \quad (3.10)$$

h için $n \in Z$ olmak üzere, eşitsizlik denklem 3.11'da görülmektedir.

$$h \leq \frac{2000}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} + \frac{2000cot^4(0{,}5\alpha)}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} - \frac{12000cot^2(0{,}5\alpha)}{cot^3(0{,}5\alpha) - \cot(0{,}5\alpha)} \quad (3.11)$$

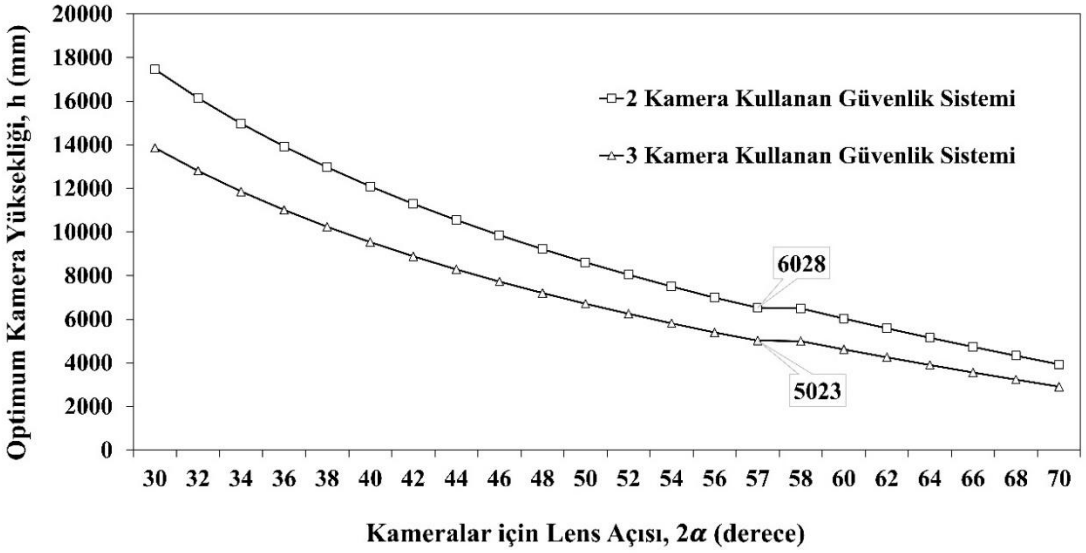
Denklem 3.10 ve 3.11'deki eşitsizlikler α ve h bilinmeyenleri için birleştirildiğinde, f(α,h) için15 ve 35 derece arasındaki değerler Tablo 3.2'de yer almaktadır.

Tablo 3.2 İkiden Fazla Kameralı Sistemde Kamera Yüksekliğinin, Lens Açısına Bağlı Olarak Sınır Değerleri

α (°)	α (radyan)	h için alt sınır(mm)	h için üst sınır(mm)
15	0.2618	10392	13856
16	0.27925	9602	12803
17	0.29671	8895,4	11860
18	0.31417	8258,3	11011
19	0.33163	7679,6	10240
20	0.34909	7150,5	9534
21	0.36655	6663,7	8884,9
22	0.38401	6213,2	8284,2
23	0.40147	5794,1	7725,5
24	0.41893	5402,4	7203,2
25	0.43639	5034,6	6712,8
26	0.45385	4687,7	6250,3
27	0.47131	4359,3	5812,3
28	0.48877	4047,1	5396,1
28,937	0,50505	3767,6	5023,5
29	0.50623	3749,2	4999
30	0.52369	3464,1	4618,8
31	0.54115	3190,3	4253,7
32	0.55861	2926,4	3901,9
33	0.57607	2671,4	3561,8
34	0.59353	2424,2	3232,2
35	0.61099	2183,8	2911,8

Denklem 3.5'te bulunan açı değeri Tablo 3.1'de yerine koyulduğunda, h yüksekliğinin üst sınırı, 3767.6mm olarak bulunmaktadır. Bu değerler hedef fonksiyonda yerine koyulduğunda, bir kameranın taradığı maksimum alan, $23{,}5609m^2$ olarak bulunmaktadır

Tasarım kısıtlamaları için sağlanan tüm hesaplanan α ve h değerleri, farklı kamera merceği açılarına göre yerleştirilecek yükseklikleri grafiksel olarak Şekil 3.2'de verilmiştir. Şekildeki grafik iki eğri içermektedir. Biri "2 Kameralı Güvenlik Sistemi İçin Kamera Yüksekliği", diğeri ise "3 Kameralı Güvenlik Sistemi İçin Kamera Yüksekliği". Daha önceki hesaplamalarda en uygun kamera lens açısının 57.874° olduğu görülmektedir. Bu nedenle iki kameralı sistem için gerekli yükseklik 6028 mm, üç kameralı sistem için gerekli yükseklik 5023 mm olarak hesaplanmıştır. Optimum kamera açısının kullanılmasının istenmediği durumlarda mevcut kameranın lens açısına uygun kamera yükseklikleri de grafiklerle gösterilmiştir. İki kamera yeterli olmadığında (yani kamera yüksekliği 10 m'den büyük olduğunda) üç kameraya ihtiyaç duyulmuştur. Üç kamera kullanılması gerektiğinde kameralar dairesel ve eşit aralıklı olarak konumlandırılmıştır. Üç kameranın kullanıldığı durumda, toplam projeksiyon alanı ve önceki çözümün kısıtlamaları değişmektedir. Bu durumda yeni kısıtlama (htan(2α)) 6000 mm'den büyük ve 8000 mm'den küçük olmalıdır.



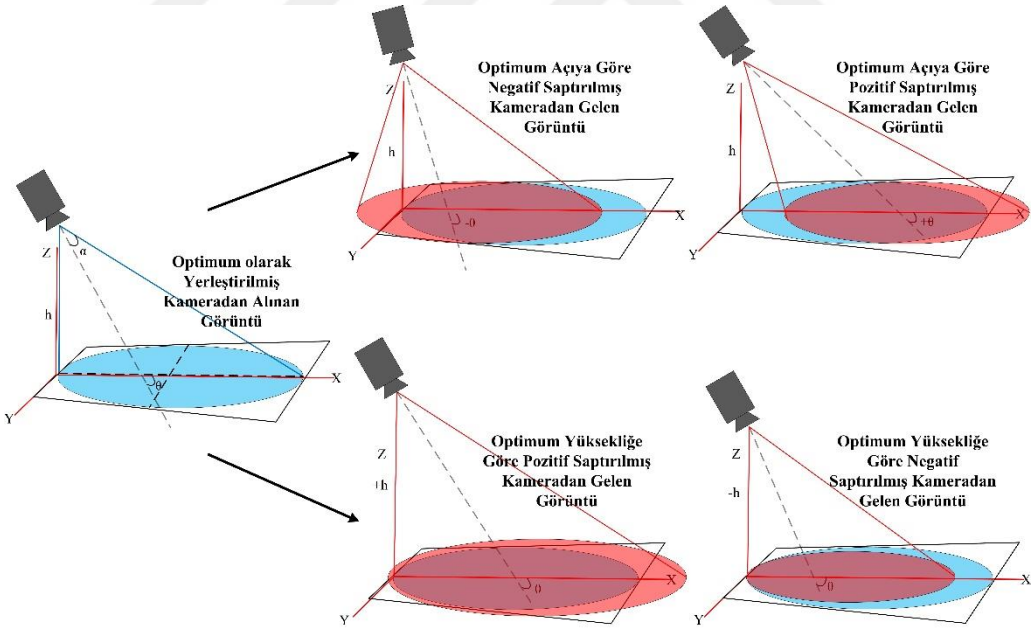
Şekil 3.2 İki ve üç kameralı sistemler için kamera lens açısına göre optimum kamera yüksekliği.

Bir önceki bölümde ele alınmış olan matematiksel model bir sistem üzerinde uygulanmak istendiğinde, öncelikle sistemde kullanılacak olan kamera ve robot kol boyutları belirlenmelidir. Doktora çalışması için ucuz ve kolay bulunabilir olması sebebiyle eklemeli imalat yöntemiyle üretimi yapılmış olan küçük bir robot kol tercih edilmiştir. Bu robot kolun çalışma uzay çapı 22cm olarak belirlenmiştir. Bununla birlikte robot kolun uygun yerleşimlerinin yapılabilmesi için, öncelikle son kullanıcının da rahatlıkla anlayıp kullanabileceği bir yazılım gerekmektedir. Bu yazılım, bir önceki başlıkta ortaya koyulmuş olan matematiksel modelin bir arayüz vasıtasıyla kullanıcıyla buluşturulmasıdır. Ardından uygun parametre değerlerine göre sistem oluşturularak, görüntü işlemeye dayanan güvenlik yazılımı oluşturulmalıdır. Dolayısıyla Matematiksel modelin sisteme uygulanması noktasında yapılacak işlemler şu şekilde sıralanabilmektedir;

- En uygunu bulmak için bilgisayar yazılımının oluşturulması,
- Robot kol kullanılan iş hücresinin tasarımının yapılması,
- İş hücresinin kullanılan robot kol gövde yapısının oluşturulması,
- İş hücresinin kullanılan robot kol kontrolcü yapısının oluşturulması,
- İş Hücresinin Çevresel Yapısının oluşturulması,
- Robot kol kullanılan iş hücresinin görüntü işlemeye dayalı güvenlik sisteminin oluşturulmasıdır.

3.2 En Uygun Çözümün Yapay Zekâ Algoritması Kullanılarak Doğrulanması

Doktora çalışmasında, optimum kamera yüksekliğine ve yönlendirme açısına sahip bir güvenlik sistemi tasarlanmıştır. Bu tasarımın doğrulama deneylerinden önce optimizasyon probleminin çözümünün doğrulanması gerekmektedir. Şekil 3.3'teki en soldaki görüntü, bir kamera tarafından optimum yerleşimde taranan alanı göstermektedir. Sağ üstteki iki resim, kamera oryantasyon açısının pozitif ve negatif değişimi ile oluşan kırmızı alanları göstermektedir. Sağ alttaki iki resim, kamera yüksekliğinin pozitif ve negatif değişimiyle kırmızı alanları göstermektedir. Şekilde göüldüğü üzere taranması gereken mavi alan ile değişiklikler sonucunda oluşan kırmızı alanlar arasında fark oluşmuştur.. Bu farkı teknik bir şekilde ifade etmek için, referans görüntü ile değiştirilen görüntüler arasındaki benzerliği verecek bir yapay zekâ algoritması tasarlanmıştır.

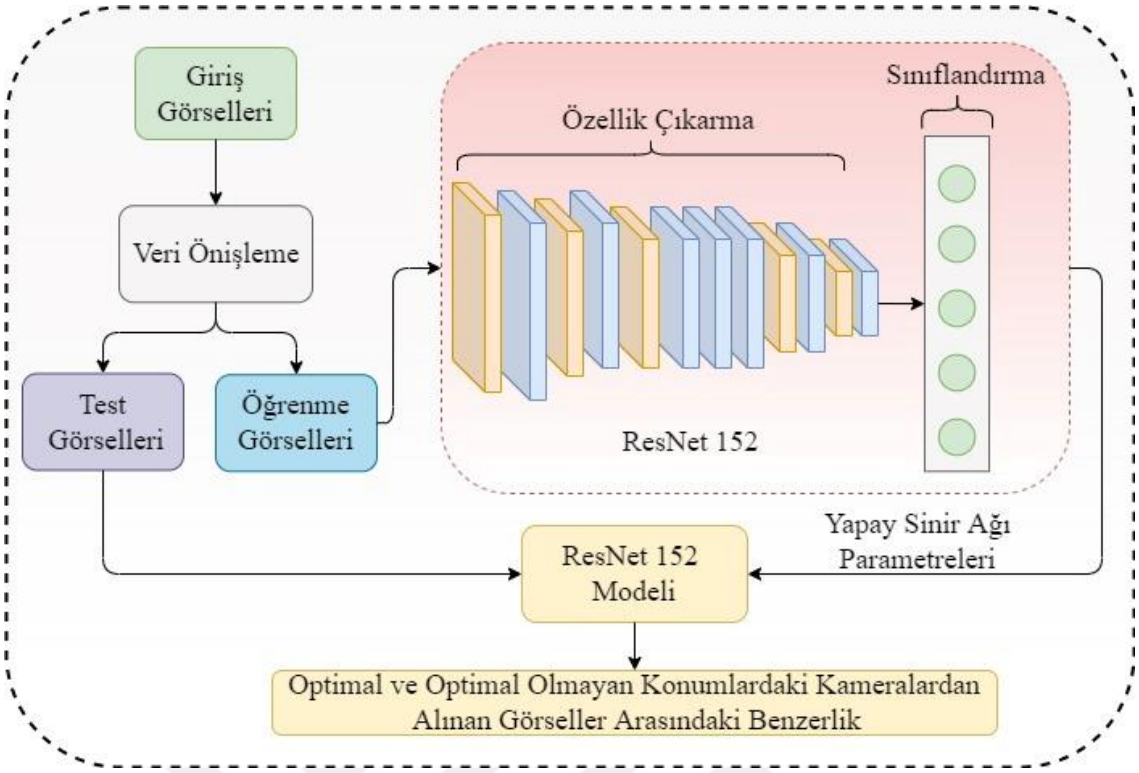


Şekil 3.3 Optimum olmayan oryantasyon açısı ve yükseklikte yerleştirilen kamera tarafından taranan alanların farkı

Değiştirilmiş (artırılmış ve azaltılmış) optimum olmayan kamera yüksekliklerinden ve oryantasyon açılarından alınan görüntülere dayalı olarak sekiz farklı görüntü

sınıfı oluşturulmuştur. Bu sekiz sınıfın her birinden yaklaşık 500 görüntü alınmıştır. Bu görüntüler ResNet-152 derin öğrenme algoritmasını eğitmek için kullanılmıştır.

ResNet-152, optimum ve optimum olmayan görüntü verilerini karakterize etmek için kullanılan popüler bir derin öğrenme algoritmasıdır, böylece optimum olmayan verilerde bulunan gürültünün etkileri ortadan kaldırılmaktadır. Dolayısıyla, uç noktalarda ve açılarda elde edilen tekil görüntü verilerini referans olarak kullanmak yerine, önceden belirlenmiş aralıkta elde edilebilen görüntülerin ortak özellikleri istatistiksel bir yaklaşımla çıkarılmaktadır. ResNet mimarisini diğer mimarilerden ayıran bu özellik ile katman sayısı arttıkça ve tahmin doğruluğu arttıkça optimizasyon daha konforlu hale gelmekte ve kaybolan gradyan sorunu çözülmektedir. Bu nedenle, referans görüntünün eğitilmiş algoritma ile karşılaştırılması için bu mimari tercih edilmiştir. Buradaki amaç, alınan optimum kamera görüntüsünü karşılaştırmak yerine, endüstriyel ortamda titreşim, ışık parlamaları ve kirlilikten etkilenen gürültülü görüntüleri dahil ederek daha gerçekçi sonuçlar elde etmektir. Böylece optimum çözümün robotik iş hücresini tamamen kaplayan en doğru çözüm olduğu ortaya konmuştur. ResNet mimarisi, ileri beslemeli ağ katmanlarına eklenen daha derin bir bağlantı atlamalı ağ sağlamaktadır. Bu nedenle, kısayol bağlantıları ve özellikleri, doğrudan F(x)'ten öğrenmek yerine F(x)'e x (giriş değeri) eklenerek öğrenilmektedir [48]. Blok yaklaşımı ile çıktı olarak, ReLu işleminden x + F(x)'in sonuçları alınmaktadır. Sınıflandırma ve karşılaştırma algoritmasının çalışma prensibi Şekil 3.4'te gösterilmiştir.



Şekil 3.4 Derin öğrenme algoritmasının akış şeması

Son olarak, ilgili sekiz sınıfta eğitilen veri setleri, optimum kurulumda çekilen görsellerle karşılaştırılmıştır. Bu karşılaştırma sonucunda optimum parametrelere yerleştirilen kameralardan alınan görüntüler arasındaki fark yüzde olarak ifade edilmektedir. Bu sekiz sınıf ve farklılıklar Tablo 3.3 ve Tablo 3.4'te gösterilmiştir.

Tablo 3.3, kamera yüksekliğindeki sapmaya bağlı olarak optimum görüntüden sapmanın arttığını göstermektedir. Tablo 3.4, kamera yerleştirme açısının artması ve azalmasına bağlı olarak optimum görüntüden farkın arttığını göstermektedir. Başvurulan görüntü, robot kolunun görünür olduğu istenmeyen alanları hariç tutan ve tüm güvenlik bölgelerini tarayan görüntüdür. Dolayısıyla bu referansa göre farkın artması, istenmeyen bölgelerin taranamaması ve istenen bölgelerin kısmen taranamaması anlamına gelmektedir. Derin öğrenme algoritmaları kullanılarak optimal olmayan parametrelerin karşılaştırılmasında temel optimizasyon yöntemleri ile elde edilen parametrelerin doğruluğu kanıtlanmıştır.

Tablo 3.3 En uygun şekilde yerleştirilmiş kameradan alınan görüntüler ile değiştirilen kamera yüksekliğinden alınan görüntüler arasındaki benzerlik yüzdesi

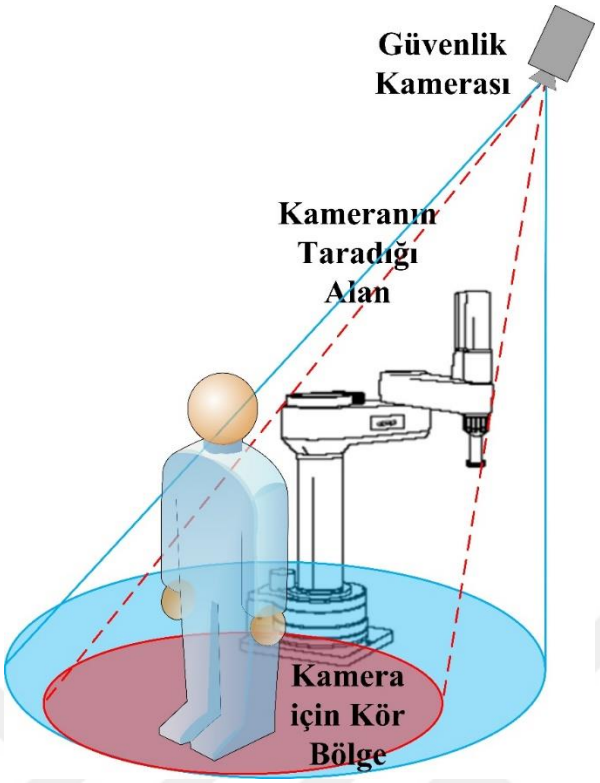
Kamera Yüksekliği Değişikliği	**Optimal Yerleştirilmiş Kamera Görüntüsüne Benzerlik Yüzdesi**
10% Azaltılmış	58%
5% Azaltılmış	60%
5% Artırılmış	55%
10% Artırılmış	41%

Tablo 3.4 En uygun şekilde yerleştirilmiş kameradan alınan görüntüler ile değiştirilen kamera oryantasyon açısından alınan görüntüler arasındaki benzerlik yüzdesi

Kamera Oryantasyon Açısı Değişikliği (Derece)	**Optimal Yerleştirilmiş Kamera Görüntüsüne Benzerlik Yüzdesi**
10^o Azaltılmış	57%
5^o Azaltılmış	59%
5^o Artırılmış	51%
10^o Artırılmış	41%

3.3 2 Boyutlu Kamera Görüntülerinin Birleştirilmesi

Tek kamera ile oluşturulan güvenlik sistemlerinde robot kolun gölgelediği alanlarda kör alanların oluşabileceği öngörülmektedir. Bu durum Şekil 3.5'te görülmektedir. Bu nedenle kurulan tüm sistemlerde birbirine simetrik olarak en az iki kamera kullanılması tavsiye edilmektedir. Böylece kör alanların önüne geçilmektedir.

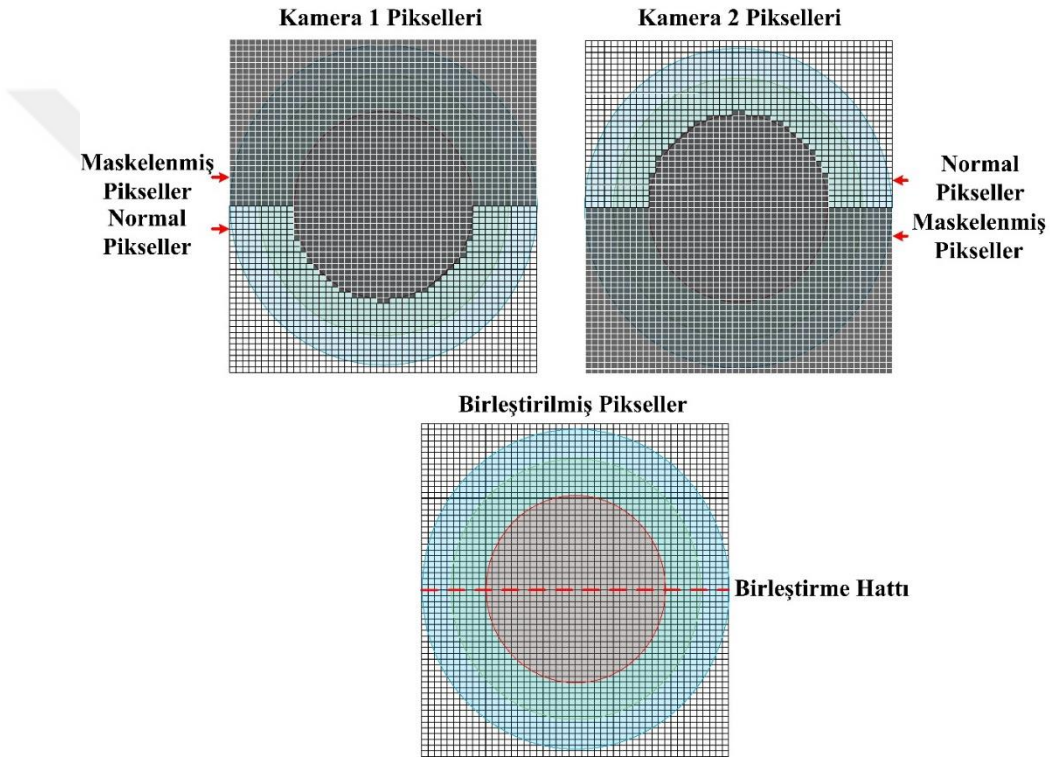


Şekil 3.5 Tek bir kamera kullanıldığında oluşabilen kör bir alan

Birden fazla kameralı güvenlik sisteminde karşılaşılan sorunlardan biri birden fazla görüntünün bir araya getirilmesidir. Birbirinden uzaktaki iki kameradan alınan iki görüntüyü birleştirme yöntemi, nispeten geleneksel bir teknoloji olan stereo görüntüleme olarak adlandırılmaktadır. Stereo görüntüleme, önemli bir hesaplama maliyetiyle üç boyutlu algı avantajını getirmektedir, ancak yine de kör nokta oluşumu kaçınılmazdır. Bununla birlikte, doktora çalışmasında önerilen iki boyutlu görüntü birleştirme, hesaplama açısından daha ekonomiktir ve kör nokta oluşumunun önüne geçmektedir.

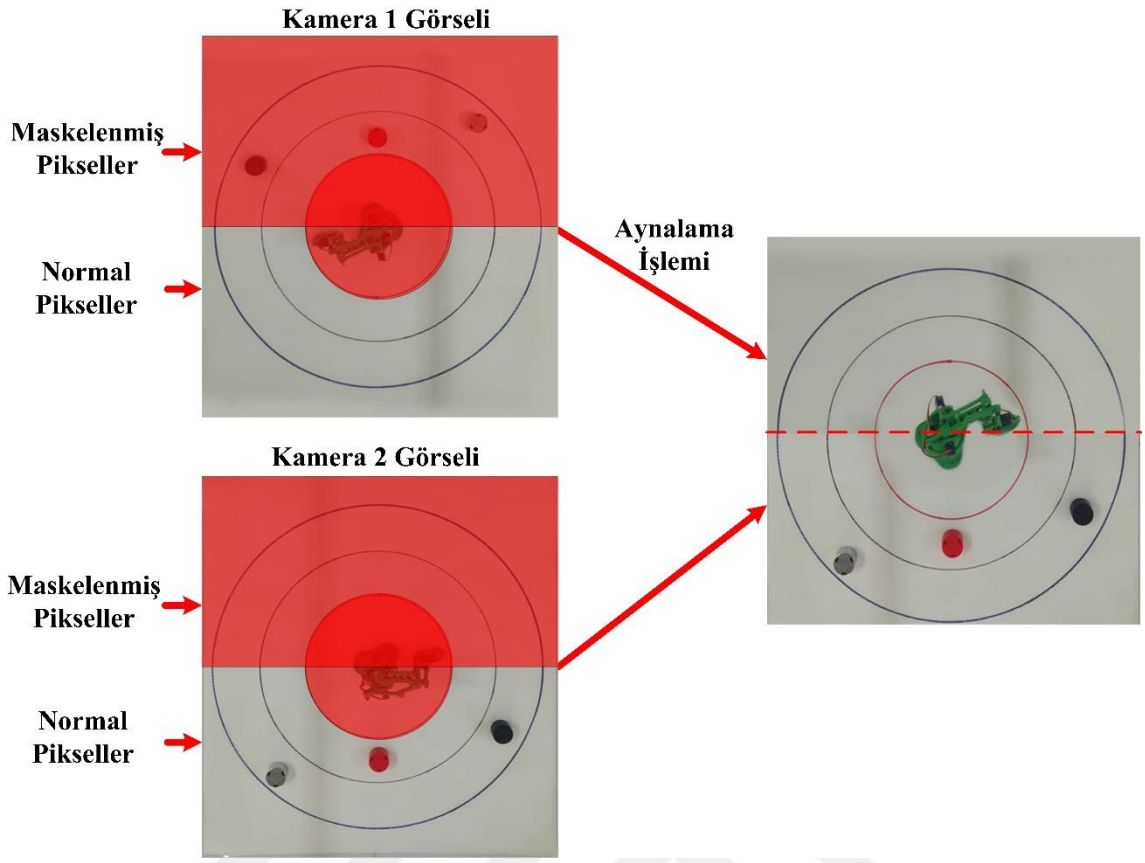
Birden fazla kamera görüntüsünün birleştirilmesinde bir diğer önemli kriter, aynı bölgelerin taranması sırasında güvenlik alanına giren bir cismin birden fazla görüntüsünün oluşmasıdır. Bu, algılanacak piksel sayısının belirlenmesinde karmaşıklık yaratmaktadır. İkinci kriter, görüntü işleme algoritmaları ile aynı alanların birden fazla taranmasının kontrol ünitesi için ek bir işlem yükü gerektirmesidir. Güvenlik sistemi için mümkün olan en hızlı tepki gerekli olduğundan, kontrolör üzerindeki ek yük dezavantajlı olmaktadır. Bu kriterler göz önünde bulundurulduğunda birden fazla kameradan alınan görüntünün kısmi maskelenmesi önerilmektedir. Her kamera için kör noktanın olmadığı bölge

işlenerek, kör noktanın oluşabileceği dış bölgenin maskelenmesi yapılarak kör noktaların önüne geçilmiştir. Sonuç olarak aynı bölgelerin birden fazla kez işlenmesi engellenmiştir. Literatürdeki görüntü birleştirme algoritmaları ile karşılaştırıldığında, güvenlik sistemine entegre edilebilen ve daha verimli çalışabilen özgün bir yöntem kullanılmıştır. Şekil 3.6, iki kameralı bir sistemin görüntü pikselleri üzerine yerleştirilmiş dairesel bir çalışma hücresinin şemasını göstermektedir. Kamera 1 ve Kamera 2'den alınan görüntülerin üst yarısına kısmi maskeleme uygulanmıştır. Maskelenmemiş kısmi görüntüler kör nokta olmadan birleştirilerek tek bir görüntü üretilmiştir



Şekil 3.6 Her piksele dayalı olarak iki simetrik kameranın kısmen maskelenmiş görüntülerinin birleştirilmesi

Şekil 3.7'de önerilen görüntü birleştirmenin daha ayrıntılı bir perspektif şeması gösterilmektedir. Birden fazla görüntünün maskelenmemiş bölümlerinden geliştirilen mono görüntü doğal olarak gerçekçi değildir, ancak bir çalışma hücresinin güvenlik bölgelerine giren yabancı bir nesneyi algılamak için yeterlidir. Üç veya daha fazla kamera kullanıldığında, 120°, 90°, 72°, vb.'deki pasta dilimi görüntüleri de benzer şekilde birleştirilebilir.

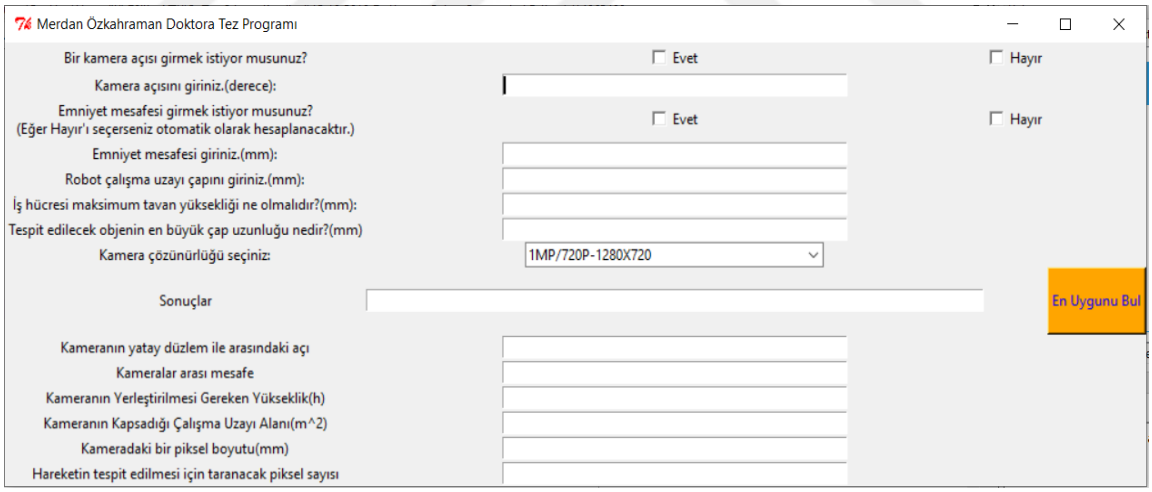


Şekil 3.7 Kısmen maskelenmiş gerçek zamanlı farklı kamera görüntülerinin birleştirilmesi.

4
SİSTEMİN GERÇEKLENMESİ

4.1 En Uygunu Bulmak İçin Bilgisayar Yazılımının Gerçeklenmesi

Ortaya konulmuş olan matematiksel model çerçevesinde optimal bir çözüm geçmiş bölümlerde açıklanmıştır. Bu çözüme dayanarak bir tasarım yapılmak istendiğinde her seferinde ayrı bir hesaplama yapmak yerine uygun bir bilgisayar yazılımının bu işlemi görsel hale getirmesi amaçlanmaktadır. Ayrıca ortaya konuşmuş olan çözüm endüstriyel üretim alanında robot kol kullanan yatırımcılar için kullanılabilir bir çözüm yöntemidir. Bu yöntemin herkes tarafından kullanılabilir olması, özel sektör için pazarlanabilir olması amacıyla bir yazılım geliştirilmiştir. Bu yazılımın arka planında ortaya konmuş olan matematiksel modelin optimum çözüm yöntemi yer almaktadır. Şekil 4.1'de tez çalışması için tasarımı yapılmış olan yazılımın görseli görülmektedir.



Şekil 4.1 En uygunu bulma yazılımının görseli

Şekil 4.1'de verilmiş olan görselde; Birinci satırda; "Bir kamera açısı girmek istiyor musunuz?" ifadesi yer almaktadır. Karşılığında seçebileceğimiz iki seçenek bulunmaktadır. Bunlar; "Evet" ve "Hayır" seçenekleridir. Bu seçeneklerden "Evet" seçildiğinde, kamera lens açısının girilmesi gerekmektedir. "Hayır" seçeneği işaretlendiğinde ise önerilen tasarıma uygun olarak kullanılacak kameranın en uygun lens açısı otomatik olarak hesaplamaktadır.

İkinci satırda; “Kamera açısı giriniz (derece)” ifadesi yer almaktadır. 1. Satırda yer alan “Evet” seçeneği seçildiği takdirde bu ifadenin karşısındaki boşluğa kullanılmak istenen kameranın lens açısı derece cinsinden yazılmalıdır.

Üçüncü satırda; “Emniyet mesafesi girmek istiyor musunuz? (Eğer Hayır’ı seçerseniz otomatik olarak hesaplanacaktır.)” ifadesi yer almaktadır. Karşılığında seçebileceğimiz iki seçenek bulunmaktadır. Bunlar; “Evet” ve “Hayır” seçenekleridir. Bu seçeneklerden “Evet” seçildiğinde, iş hücresi için kullanılmak istenen emniyet mesafesinin girilmesi gerekmektedir. “Hayır” seçeneği işaretlendiğinde ise uluslararası standartlarda öngörülen emniyet mesafesi formülüne göre otomatik olarak mesafe hesaplanmaktadır.

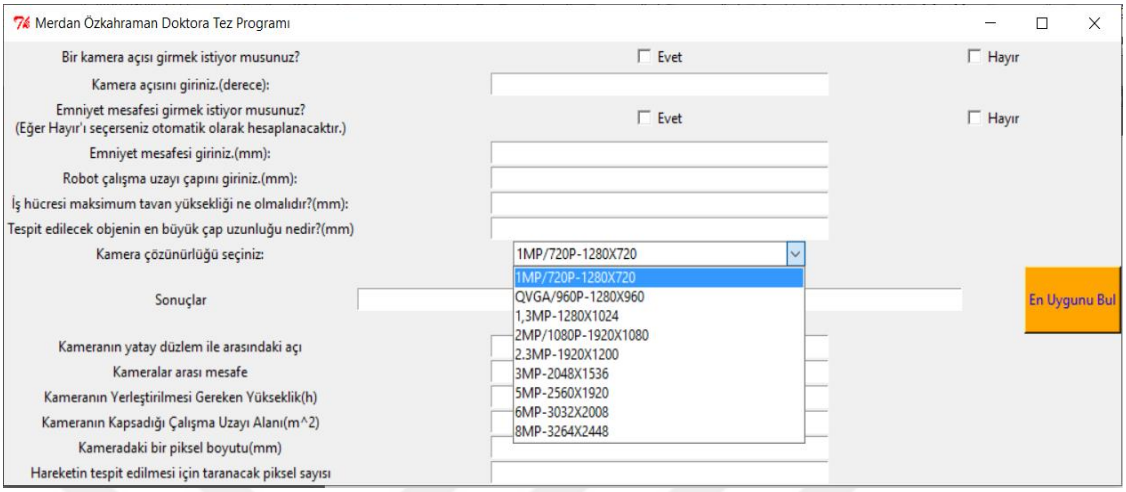
Dördüncü satırda; “Emniyet mesafesi giriniz. (mm)” ifadesi yer almaktadır. 3. Satırda yer alan “Evet” seçeneği seçildiği takdirde bu ifadenin karşısındaki boşluğa kullanılmak istenen emniyet mesafesinin milimetre cinsinden yazılması gerekmektedir.

Beşinci satırda; “Robot çalışma uzayı çapını giriniz (mm)” ifadesi yer almaktadır. Bu ifadenin karşısındaki boşluğa iş hücresinde kullanılmak istenen robot kol çalışma uzayının yatay düzlemdeki çap mesafesinin milimetre cinsinden girilmesi gerekmektedir.

Altıncı satırda; “İş hücresi maksimum tavan yüksekliği ne olmalıdır? (mm)” ifadesi yer almaktadır. Bu ifadenin karşısındaki boşluğa üretim alanındaki iş hücresinin sınırları belirlenerek kullanılmak istenen iş istasyonundaki çatı mesafesinin milimetre cinsinden girilmesi gerekmektedir.

Yedinci satırda; “Tespit edilecek objenin en büyük çap uzunluğu nedir? (mm)” ifadesi yer almaktadır. Bu ifadenin karşısındaki boşluğa güvenlik alanının içerisine girdiğinde robotun otomatik olarak duracağı istenmeyen cismin en küçük boyutunun çap uzunluğunun milimetre cinsinden girilmesi gerekmektedir. Burada eğer obje bir daire değilse en büyük kenarı çap olarak girilmelidir. Bu satırda yer alan giriş görüntü işlemeye dayalı güvenlik kontrol yazılımının eşik piksel değerini belirleyeceğinden önemlidir.

Sekizinci satırda; “Kamera çözünürlüğünü seçiniz” ifadesi yer almaktadır. Bu ifadenin karşısında bir seçenek dizisi ortaya çıkmaktadır. Bu seçeneklerin yer aldığı görsel Şekil 4.2’de görülmektedir.

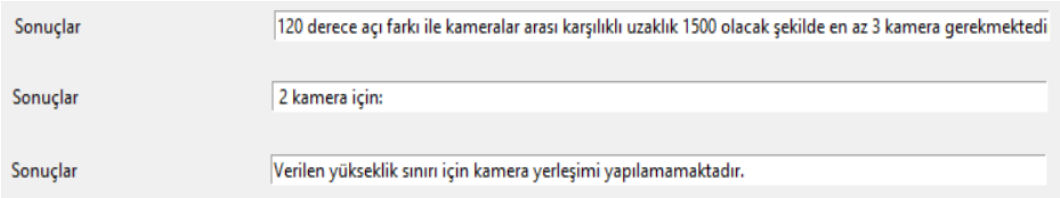


Şekil 4.2 En uygunu bulma yazılımının seçenek dizisi görseli

Şekil 4.2’de yer alan seçenek dizisinden kullanılmak istenen kameranın çözünürlük değerinin seçilmesi gerekmektedir. Bu değer görüntü işlemeye dayalı güvenlik kontrol yazılımının eşik piksel değerini belirleyeceğinden önemlidir. Dizide yer alan seçenekler piyasada bulunan kameraların çözünürlük değerlerini temsil etmektedir. Bunlar;

- 1MP/720P-1280x720
- QVGA/960P-1280X960
- 1.3MP-1280X1024
- 2MP/1080P-1920X1080
- 2.3MP-1920X1200
- 3MP-2048X1536
- 5MP-2560X1920
- 6MP-3032X2008
- 8MP-3264X2448

Dokuzuncu satırda; “Sonuçlar” ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun çözümün sonuçları program çalıştırıldığında yazmaktadır. Burada 3 tane seçenek yazabilir. Bunlar Şekil 4.3’ de görülmektedir.



Şekil 4.3 En uygunu bulma yazılımı sonuçlar sekmesi görseli

Şekil 4.3'te görülen sonuçlar;

- "120 derece açı farkı ile kameralar arası karşılıklı olarak uzaklık "x" olacak şekilde en az 3 kamera gerekmektedir." ifadesinde x üst seçenekte girilen değerlere göre belirlenmektedir. İki kameranın yetmediği yerleşim geometrilerinde sonuç kısmında görülmektedir.
- "2 kamera için;" ifadesinde iki kameranın yeterli olduğu durumlar için hesaplanan değerler verilmektedir.
- "Verilen yükseklik sınırı için kamera yerleşimi yapılamamaktadır." İfadesinde program çalıştırılmadan önce belirlenmiş olan değişkenlerden seçilen yükseklik değerinde ne kadar kamera kullanılırsa kullanılsın robot kol çalışma uzayını tam olarak tarayamadığı durumlarda sonuç olarak verilmektedir.

Onuncu satırda; "Kameranın yatay düzlem ile arasındaki açı" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun kamera ve yatay düzlem arasındaki açı değeri verilmektedir. Bu değer hem derece hem de radyan olarak ifade edilmektedir.

Onbirinci satırda; "Kameralar arası mesafe" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun yerleştirilecek kameraların arasındaki mesafe milimetre cinsinden verilmektedir.

Onikinci satırda; "Kameranın yerleştirilmesi gereken yükseklik(h)" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun yerleştirilecek kamera yüksekliği milimetre cinsinden verilmektedir.

Onüçüncü satırda; "Kameranın kapsadığı çalışma uzayı alanı(m^2)" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere

göre hesaplanan en uygun yerleştirilecek kameranın kapsadığı çalışma uzayı alanı milimetrekare cinsinden verilmektedir.

Ondördüncü satırda; "Kameradaki bir piksel boyutu(mm)" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun kamera görüntüsünde yer alan bir adet pikselin gerçek boyutta kapsadığı mesafe milimetre cinsinden verilmektedir.

Onbeşinci. satırda; "Hareketin tespit edilmesi için taranacak piksel sayısı" ifadesi yer almaktadır. Bu ifadenin karşısındaki boşlukta üst satırda girilmiş olan değişkenlere göre hesaplanan en uygun kamera görüntüsünde yer alan hareketin tespit edilmesi için taranacak piksel sayısı verilmektedir. Bu ifade görüntü işlemeye dayalı güvenlik kontrol yazılımının eşik piksel değerini belirleyeceğinden önemlidir.

Şekil 4.1'de yer alan görselin sağ tarafında yer alan renkli buton ilk sekiz satırda girilen değişkenlerin ardından tıklandığında programın çalışmasını sağlayarak alt satırdaki sonuçların gösterilmesini sağlamaktadır.

Kameranın yatay düzlem ile arasındaki açı	0.5061 radyan - 29.0 derece
Kameralar arası mesafe	3000.0 mm
Kameranın Yerleştirilmesi Gereken Yükseklik(h)	4504.59 mm
Kameranın Kapsadığı Çalışma Uzayı Alanı(m^2)	2.75 m^2
Kameradaki bir piksel boyutu(mm)	4.4 mm
Hareketin tespit edilmesi için taranacak piksel sayısı	23.0

Şekil 4.4 En uygunu bulma yazılımı örnek sonuç görseli

Şekil 4.4'te onuncu satır ve onbeşinci satır arasındaki sonuçlar bölümünün görseli görülmektedir. Üst satırlarda örnek ve rastgele girilmiş olan girişlerin neticesinde programın hesapladığı çözümün örnek sonuçlarını göstermektedir.

4.2 Robot Kol Kullanılan İş Hücresinin Deneysel Tasarımının Gerçeklenmesi

İş hücresinde kullanılabilecek, bir robot kol tasarımı yapılmak istendiğinde, hedefin çalışma uzayı ve güvenliği ile alakalı olduğu noktada uygun maliyetli ve küçük boyutlarda bir robot kol kullanılması uygun görülmüştür. Bunun için 3 boyutlu yazıcıda üretilmiş plastik malzemeden imal edilen bir robot kolun kullanılabileceği düşünülmüştür.

4.3 Deneysel Robot Kol Gövde Yapısı

Doktora çalışmasında kullanılmış olan robot kol gövdesini oluşturacak olan parçalar, ucuz ve kolay bulunabilir olması amacıyla üretimde eklemeli imalat yöntemi tercih edilmiştir. Üretilmek istenen ürün, öncelikle bir CAD programında tasarımı yapıldıktan sonra üç boyutlu yazıcılarda kullanılan STL formatına dönüştürülmektedir. Eklemeli imalat alanında yüksek çözünürlüklü yazdırma bağlantısı için STL formatlı dosya kullanılması bir standarttır. Daha sonra, hata kontrolü ve düzeltimi, üretim alanına yerleşimi ve yapı yönü, ihtiyaç halinde destek yapılarının oluşturulması gibi bazı ön işlemlerden geçmektedir. Bu noktadan sonra 3 boyutlu model, 2 boyutlu olacak şekilde katmanlara ayrılmaktadır. Böylece, imalata hazır hale gelmektedir. İmalat ve destek malzemesi olarak ABS, polikarbonat, PetG ve PLA gibi yüzlerce polimer malzeme kullanılmaktadır.

Eklemeli imalat teknolojileri;

- Steryolitografi (SLA)
- Katı zemin kürleme (SGC)
- Lamine Nesne İmalatı (LOM)
- Eriyik Yığarak Modelleme (FDM)
- 3B Yazıcı (3D Printer)
- Çok Jetli (Polyjet-Multijet) Modelleme
- Seçmeli lazer sinterleme ve ergitme (SLS/SLE)
- Elektron ışınlı ergitme (EBM)

Bu eklemeli imalat türlerinden, Eriyik Yığarak Modelleme (FDM), günümüzde, hammadde ve üretim cihazı bakımında kolay bulunabilir ve uygun fiyatlı olması açısından hassasiyet gerektirmeyen ürünlerde sıklıkla tercih edilmektedir. Doktora çalışmasında kullanılan robot kolların üretim yöntemi FDM olarak belirlenmiştir. İlk olarak 1988'de kullanılmaya başlayan bu üretim yönteminde, ürünün hammaddesi olarak termoplastik malzeme kullanılmaktadır. Makaraya sarılmış olarak ince plastik filament halinde üç boyutlu yazıcıya bağlanmış olan hammadde, kontrollü tahrik yapısı ile erimiş maddenin çıkacağı "nozzle" bölümüne gönderilmektedir. Bu bölüme ulaşan termoplastik madde sıcaklıkla yarı ergiyik hale getirilmektedir.

Bilgisayar destekli üretim yazılımı vasıtası ile hareket edebilen ekstrüzyon ucundaki bir nozzle'dan erimiş plastik malzeme akıtılıp (ekstrüzyon) yüzeye ince bir katman halinde yığılmaktadır. Her katmanda üretim yüzeyi bir adım aşağıya inmekte veya ekstrüzyon ucu bir adım yukarı çıkmaktadır. Böylece parça katmanlar halinde inşa edilmektedir. Bu yöntemle üretilmiş olan robot kol Şekil 4.5'de ve Şekil 4.6' görülmektedir.



Şekil 4.5 3 Boyutlu yazıcıda üretilmiş robot kol gövde parçaları



Şekil 4.6 3 Boyutlu yazıcıda üretilmiş robot kol gövdesi

4.4 Deneysel Robot Kol Kontrolcü Yapısı

Doktora çalışmasında kullanılmak istenen robot kolun, bir yüke maruz kalmayacağı düşülmektedir. Çözümü yapılmış olan problemin küçük boyutlu bir gerçeklemesiyle çözümün doğruluğu sorgulanacağı için düşük fiyatlı bir motor ve kontrolcü tercih edilmiştir. Bu noktada mevcut motor çeşitleri arasında bu tür uygulamalarda sıklıkla kullanılan ve birtakım avantajlara sahip olan servo motorlar kullanışmıştır. Servo motor; geri besleme yöntemiyle hataları denetleyen ve bu hataların giderilebilmesine olanak veren bir motor türüdür. Servo motorlar, geri beslemeli olan doğru akım motorlarını, alternatif akım motorlarını ve step(adım) motorlarını kapsamaktadır.

Servo motorlar, istenilen komutları yerine getirebilen, farklı hızlarda dahi kararlı bir şekilde çalışabilen motor türüdür. Bu motor türü, elektronik bir kontrolcüye ihtiyaç duymaktadır. Yapılan çalışmada, doğru akım Tower Pro SG90 Mini servo motoru kullanılmıştır. Çok yönlü çalışması, darbe genişlik kontrolünün 10µs olması, Vpp: 3-5V kare dalga ile çalışması ve çalışma geriliminin 4,8-6V olması gibi özellikleri sebebiyle bu motor tercih edilmiştir. Kullanılan servo motorun küçük çalışma geriliminde, çalışma hızı yüksüz olarak 0,12s/60° ve torku 1,2-1,6kg/cm'dir.

Servo motorlar, kontrolcüden gelen sinyal ile kontrol edilmektedir. Bunu yaparken darbe genişlik modülasyon (PWM) sinyali kullanılmaktadır. Her bir servo motor, 10-20ms'de ve 0,5-1,5ms'de bir PWM sinyali ile kontrol edilmektedir. Bu sinyalin lojik 1'de kalış süresine (tk) göre motor milinin konumları belirlenmektedir. Bunlar;

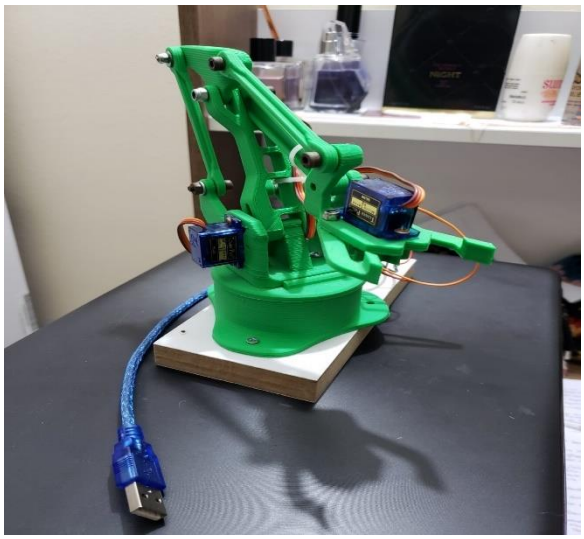
- tk=0,5ms iken motor mili en sona döner,
- tk=0,5-1ms iken motor milinin konumu ortadadır,
- tk=1-1,5ms iken motor mili sağa döner,
- tk=10-20ms iken (tekrar aynı sinyal verildiğinde) eski konumunda kalır,

Servo motor, yukaradaki maddeler şeklinde hareket etmektedir. Şekil 4.7' de anlatılmış olan servo motor görülmektedir.



Şekil 4.7 Servo motor

Şekil 4.8' de gövdesi oluşturulmuş olan robot kola servo motorların monte edilmiş hali görülmektedir. Robot kol eklemlerinde kullanılan motorların kontrolü için gerekli olan elektronik devre kartı, piyasada sıklıkla bu tür projelerde kullanılmış olan Arduio kartı olarak seçilmiştir. Arduino markasının uygulamalara yönelik farklı çeşitlerinden olan Arduino Nano robot kol üzerindeki servo motorları kontrol etmektedir. Arduino Nano, ATmega328 isimli mikrodenetleyiciyi veya Atmega168 mikrodenetleyiciyi barındırmaktadır.



Şekil 4.8 Servo motorlar yerleştirilmiş robot kol gövdesi

ATmega328 ve ATmega168 mikrodenetleyicileri, RX ve TX pinlerinden erişilebilen UART TTL(5V) seri haberleşmeyi desteklemektedir. Kart üzerindeki bir FTDI FT232RL seri haberleşmeyi USB üzerinden kanalize etmektedir. FTDI sürücüleri ile bilgisayardaki yazılıma sanal bir com portu olarak görünmektedir. Kart üzerindeki RX ve TX ledleri FTDI çipi üzerinden USB'den seri çipe ve USB'den bilgisayara veri giderken yanıp sönmektedir. SoftwareSerial kütüphanesi Arduino Nano 'nun digital pinlerinden herhangi biri üzerinden seri haberleşmeye imkân sağlamaktadır. Ayrıca ATmega328 ve ATmega168 mikrodenetleyicileri I2C (TWI) ve SPI haberleşmelerini de desteklemektedir.

Arduinonun mikro denetleyici kartının çeşitli projelerde sıklıkla kullanılma sebepleri, muadillerine göre olan avantajları şunlardır;

- Sürücü kullanımıyla birlikte geliştirme ortamının basit olmasından dolayı bütün platformlarda kullanılabilmektedir.
- Gelişmiş kütüphanesi yardımıyla karmaşık olan işlemlere bile kolaylıkla çözüm bulunabilmektedir.
- Arduino'ya yazılan programlar aracı başka bir platforma ihtiyaç duymadığı için hızlı çalışabilmektedir.
- Arduino ile uyumlu ve birlikte çalışabilecek çok fazla donanım desteği bulunmaktadır.
- Açık kaynak kodlu olduğu için çevresi ile iletişime kolay geçebilmektedir.

- Arduino kullanıcı sayısı fazla olması sebebiyle herhangi bir sorun oluştuğunda kolaylıkla çözüme ulaşılabilir.
- PIC'lerden daha yüksek fiyatlı olmasına rağmen kararlılığı ve kullanım alanlarıyla, muadillerine göre tercih edilebilir olmaktadır.

Şekil 4.9'da Arduino Nano kartını görülmektedir.



Şekil 4.9 Arduino Nano

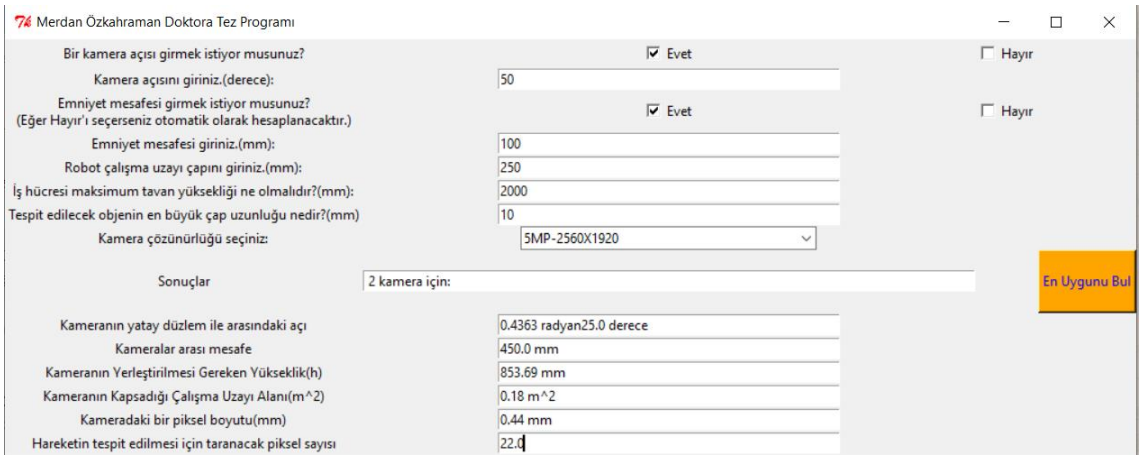
4.5 Deneysel İş Hücresinin Çevresel Yapısı

İş hücresinin çevresel yapısını belirleyebilmek için robot kol kullanılan iş hücrelerindeki değişkenlerimizi bilmek gerekmektedir. Bu değişkenlerimiz;

- Robot kol sayısı ve çeşidi
- Kamera sayısı ve lens açısı
- Kamera yerleşim açısı
- Kamera yerleşim yüksekliği şeklinde sıralanabilir.

Bu kısıtlar çerçevesinde matematiksel modelin ön gördüğü kamera sayısı ve açısı kullanılmalıdır. Fakat, bu değişebileceğinden bir referans olarak en az iki kamera kullanılması gerektiği bilinmektedir. Bununla beraber, önceki bölümlerde ele alınmış olan matematiksel model bir sistem üzerinde uygulanmak istendiğinde, Tez çalışması için ucuz ve kolay bulunabilir olması sebebiyle eklemeli imalat yöntemiyle üretimi yapılmış olan küçük bir robot kol tercih edilmiştir. Bu robot kolun çalışma uzay çapı 25cm olarak belirlenmiştir. Çalışma uzayının küçük boyutlarda kalması amacıyla emniyet mesafesi 10cm olara belirlenmiştir. Bununla birlikte 5MP çözünürlüğe sahip yaklaşık 50 derece lens açısı olan bir kamera kullanılmak

istenmiştir. Dolayısıya bu değişkenleri yapılmış olan en uygunu bulma yazılımında yerine girdiğimizde çıkan sonuç şekil 4.10' da görülmektedir.

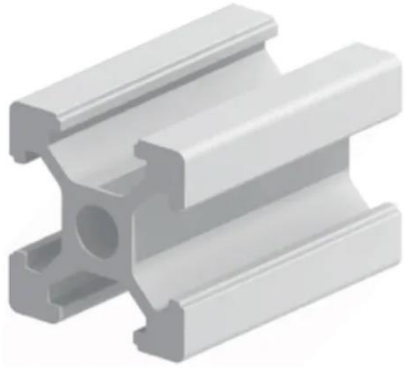


Şekil 4.10 En uygunu bulma yazılımına göre sistem tasarım parametreleri

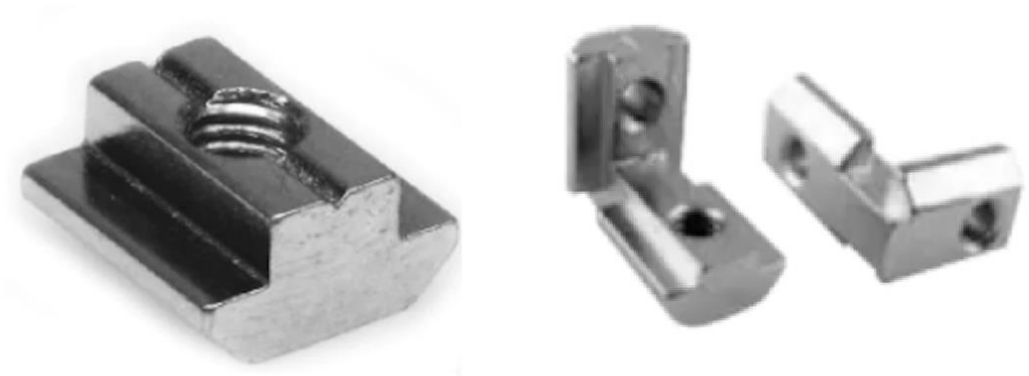
Yukarıdaki şekilde görüldüğü üzere, kameranın yatay düzlem ile arasındaki açı 25 derece, 0,4363 radyandır. İki kamera birbirinden 45cm uzağa karşılıklı olacak şekilde yerleştirilmelidir. Kameranın yerleştirilmesi gereken yükseklik 85cm civarı olarak belirlenmiştir. Kameranın kapsadığı çalışma uzayı alanı 0,18 metrekaredir. Kameradaki bir pikselin gerçekte denk geldiği boyut 0,44mm'dir. Hareketin tespit edilebilmesi için taraması gereken piksel sayısı 22'dir.

Bu çıktılar sonucunda iş hücresinin tasarımında kameranın yerleştirilmesi ve iş hücresinin çerçeve içine alınması işleminde alüminyum sigma profil kullanılması uygun görülmüştür. Alüminyum sigma profil, günümüzde çelik konstrüksiyonun yerini almaya başlamıştır. Sigma profilin modüler yapısı neticesinde çeşitli bağlantı elemanlarının vasıtası ile istenen uzunluktaki parçaların birbirine olan bağlantısı, pratik ve kolay bir şekilde yapılabilmektedir. Sigma profil, alüminyum 6063 serisinden imâl edilmektedir. Sigma Profillerin üzeri 11 mikron eloksal kaplanarak krozyona karşı daha dayanıklı hale getirilmektedir. Bu faktörler sayesinde sigma profil ile yapılan uygulamalarda sonradan taşlama ve boyama gibi işlemlere gerek kalmamaktadır. Sigma profiller standart olarak 6000mm uzunluğunda üretilmektedir. Her ebatta sigma profile uygun bağlantı elemanları ile statik ve dinamik yüklemelere karşı daha sağlam ve emin bağlantılar kurmak mümkün olmaktadır. Planlama zamanları ve malzeme sarfiyatında asgari düzeyde olduğu bilinmektedir. Sigma profillerin, 6, 8, 10mm olarak üç farklı kanal yapısı

bulunmaktadır. Ayrıca sigma profiller, uygun yapısı sayesinde hava, su ve yağ kanalı olarak da kullanılabilmektedir. Sigma profil günümüzde her sektörde kullanılmaktadır. Sigma profiller her türlü şase, kabin, paravan, masa, makine yapımı için uygundur. Sigma profillere uygun bağlantı aksesuarları sayesinde estetik bir görünüm elde edilmektedir. Bu sebeplerden doktora çalışmasında hizmet alımı yapılarak iş hücresinin çerçeve boyutlarına uygun uzunluklarda alüminyum sigma profil ürettirilmiştir. Şekil 4.11'de örnek bir alüminyum sigma profil görülmektedir. Ayrıca farklı uzunluktaki profillerin bağlantıları için çeşitli elemanlar kullanılmaktadır. Bu bağlantıların gizli olması amacıyla kullanılan parçalar, Şekil 4.12' görülmektedir.



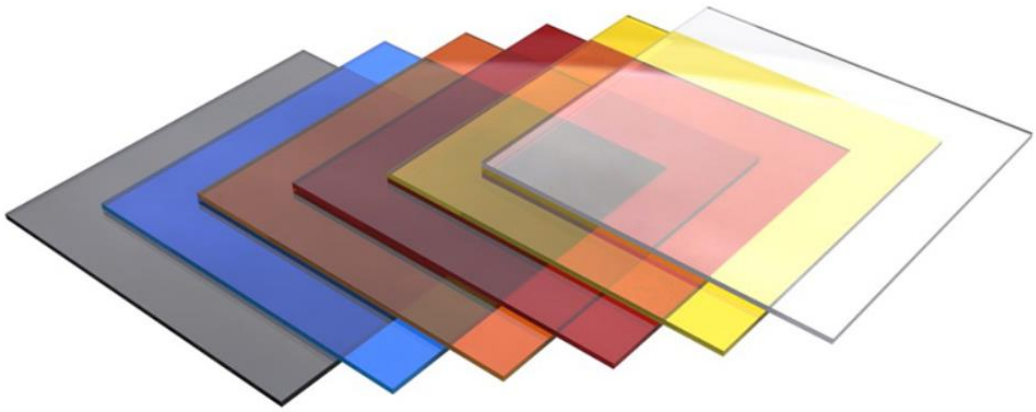
Şekil 4.11 20x20mm 6 kanallı alüminyum sigma profil



Şekil 4.12 Sigma profil bağlantı elemanları

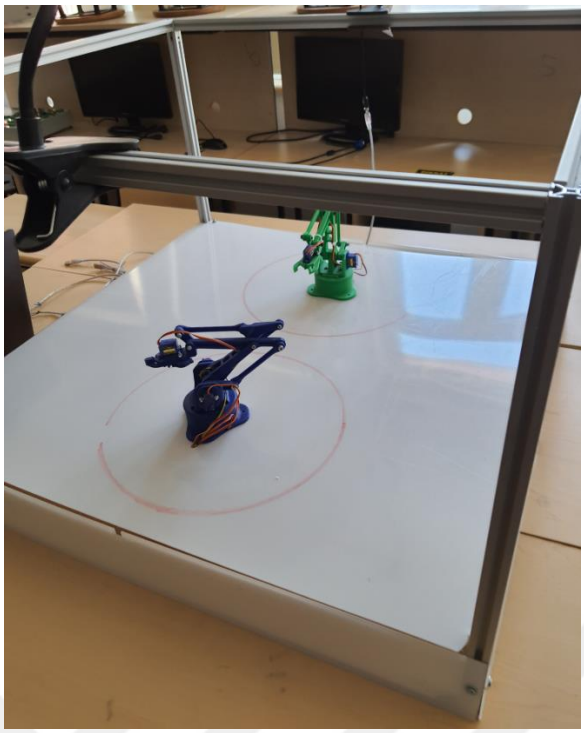
İş hücresi için profilden iskelet tamamlandığında çevresel olarak elektronik devrelerin altında gizlenebileceği ve istenmeyen bölgelerin saklanabileceği bir malzemeninde kaplama olarak kullanılması gerekmektedir. Bu amaçla tasarlanmış

olan iş hücresinin kaplama alanları için pleksi adı verilen şeffaf veya renkli çeşitleri bulunabilen plastik bir cam türü tercih edilmiştir. Pleksi adı verilen malzeme, piyasada genellikle 1,5-2,5mm kalınlığında düz levhalar halinde bulunmaktadır. 90°C sıcaklıkta ya da 90°C-115°C de etüvde ısıtılarak yumuşatılmış bir malzemedir. Böylece kalıplanarak istenilen biçim verilebilir. Camdan daha dayanıklı ve hafif olduğu bilinmektedir. Tek dezavantajı termodinamik bir yapıda olduğundan yanmaya karşı dayanıklılığı daha az olmaktadır. Pleksi malzeme istenilen boyutlarda üretilmesi, ısı kullanılarak rahatlıkla şekil verilmesi, kesme ve delme işlemlerinin kolay olması, kolay kırılmaması sebebiyle doktora çalışmasında hizmet alımı yapılarak iş hücresinin çerçeve boyutlarına uygun uzunluklarda pleksi kaplama ürettirilmiştir. Şekil 4.13'te örnek pleksi malzemeler görülmektedir.



Şekil 4.13 Örnek pleksi malzeme

Matematiksel modele uygun olarak tasarımı yapılmış olan robot kol kullanılan iş hücresi Şekil 4.14'te görülmektedir.



Şekil 4.14 Sigma profilden oluşturulmuş çerçeve tasarımının görseli

4.6 Robot Kol Kullanılan İş Hücresinin Görüntü İşlemeye Dayalı Güvenlik Sisteminin Gerçeklenmesi

Doktora çalışmasında tasarımı yapılmış olan robot kol kullanılan iş hücresinin ana amaç olan iş ve işçi güvenliği kontrolü yapılmak istendiğinde, görüntü işleme teknolojisine dayalı bir algoritmaya ihtiyaç duyulmuştur. Bunun için öncelikle görüntü işleme kütüphanesine sahip güncel ve yenilikçi bir kodlama dili konusunda araştırma yapılmıştır. Bu araştırmaların sonucunda python programlama dili seçilmiştir. Bu programlama dilinin diğer dillere göre bir takım farklı özellikleri ve avantajları olduğu bilinmektedir. Python, dinamik anlambilim ile yorumlanmış, nesne yönelimli, üst düzey bir programlama dilidir. Dinamik yazım ve dinamik ciltleme ile birleştirilmiş yüksek veri yapıları, hızlı uygulama geliştirme için ve mevcut bileşenleri birbirine bağlamak için bir betik veya yapıştırıcı dil olarak kullanılmasını çok cazip kılmaktadır. Python 90'lı yıllarda ortaya çıkmış bir programlama dilidir. Python programlama dili ismini geliştiricisinin en sevdiği gösteri olan MonthyPython isimli bir sahne gösteriden almaktadır.

Python'un basit, öğrenmesi kolay sözdizimi okunabilirliği vurgular ve böylece programın bakım maliyetini düşürmektedir. Python, program modülerliğini ve kod kullanımını yeniden teşvik eden modülleri ve paketleri desteklemektedir. Python yorumlayıcısı ve geniş kapsamlı standart kütüphane, tüm büyük platformlar için ücretsiz olarak kaynak veya ikili biçimde mevcuttur ve serbestçe dağıtılabilmektedir. Derleme adımı olmadığından, düzenleme, test, hata ayıklama döngüsü muadillerine göre hızlı olmaktadır.

4.7 Programlama Dili

Doktora Tez çalışmasında seçilmiş olan python programlama dili diğer diller; Java, JavaScript, Perl, Tcl veya Smalltalk gibi dillerle karşılaştırıldığında bir takım avantajlar sunmaktadır. C++, Common Lisp ve Scheme ile karşılaştırmalar da programla dilinin anlaşılabilmesi için faydalı olmaktadır. Yapılan uygulamalarda, bir programlama dilinin seçimi genellikle maliyet, kullanılabilirlik, eğitim ve önceki yatırımlar veya hatta duygusal bağlanma gibi diğer gerçek dünya kısıtlamaları tarafından belirlenmektedir.

Python programları genellikle eşdeğer Java programlarından 3-5 kat daha kısa olmaktadır. Bu fark Python'un yerleşik yüksek seviye veri tiplerine ve dinamik tiplendirmesine bağlanabilmektedir.

Python'un "nesne tabanlı" altkümesi, kabaca JavaScript'e eşdeğer görülmektedir. JavaScript gibi (ve Java'dan farklı olarak) Python, sınıf tanımlarına girmeden basit işlevleri ve değişkenleri kullanan bir programlama stilini desteklemektedir. Ancak, Python ise, sınıfların ve kalıtımın önemli bir rol oynadığı gerçek nesne yönelimli bir programlama stili ile daha büyük programlar yazmayı ve kodun daha iyi kullanılmasını desteklemektedir.

Python ve Perl, benzer bir arka plandan gelmektedir. Ancak Perl, ortak uygulama odaklı görevlerde kullanılmaktadır. Python ise, veri yapısı tasarımı ve nesne yönelimli programlama gibi ortak programlama metodolojileri için desteğe önem vermektedir.

Python programlama dilinde olduğu gibi, Tcl de bağımsız bir programlama dili olmasının yanı sıra, bir uygulama uzantısı dili olarak da kullanılabilmektedir.

Bununla birlikte, geleneksel olarak tüm verileri dizge olarak saklayan Tcl, veri yapılarında zayıf kalmaktadır. Tcl ayrıca, modüler ad alanları gibi büyük programlar yazmak için gereken özelliklerden de yoksun görülmektedir. Bu nedenle, Tcl kullanan "tipik" bir büyük uygulama genellikle bu uygulamaya özgü olan C veya C++ ile yazılmış Tcl uzantılarını içerirken, genellikle "saf Python" ile eşdeğer bir Python uygulaması yazılabilmektedir. Dolayısıyla, saf Python geliştirme, bir C veya C++ bileşenini yazmak ve hata ayıklamak zorunda kalmaktan çok daha hızlı olmaktadır.

Python ve Smalltalk arasındaki en büyük fark Python'un programcı eğitimi konusunda bir açıklama veren daha "ana" sözdizimi olduğu bilinmektedir. Smalltalk gibi, Python da dinamik yazma ve ciltleme özelliğine sahip ve Python'daki her şey bir nesne olarak yürütülmektedir. Ancak, Python, yerleşik nesne türlerini kullanıcı tanımlı sınıflardan ayırmaktadır. Smalltalk'ın standart veri toplama türleri kütüphanesi daha rafine olmaktadır. Python'un kütüphanesi ise internet ve e-posta, HTML ve FTP gibi WWW gerçekleriyle başa çıkmak için daha fazla olanaklara sahip olmaktadır.

Java için söylenen hemen hemen her şey C++ programlama dili için de geçerli sayılmaktadır. Python kodu eşdeğer Java kodundan 3-5 kat daha kısa olduğunda, eşdeğer C++ kodundan 5-10 kat daha kısa yazılmaktadır. Sonuç olarak Python, C++ ile yazılmış bileşenleri birleştirmek için kullanılan bir tutkal dili olarak görülmektedir.

Sonuç olarak, Python basit ve sade bir dil olarak bilinmektedir. Özetle avantajları şu şekildedir;

- Öğrenilmesi kolaydır.
- Açık Kaynak, kısaca, bu yazılımın kopyaları serbest olarak dağıtılabilir, yazılımın kaynak kodu okunabilir, kaynak kod üzerinde değişiklik yapılabilir, türetilmiş güncel programlara uyum sağlayabilir.
- Makine ile haberleşme konusunda kullanıcı için daha kolaydır.
- Farklı platformlarda yazılmış olan tek python kodu hepsiyle uyumlu çalışmaktadır.
- C veya C++ gibi programlama dilleri derlenerek çalışırken, python derleme gerektirmeden yorumlayarak çalışmaktadır. Dolayısıyla hızlıdır.

- Nesne tabanlı, Python, nesne yönelimli programlamanın yanı sıra prosedür yönelimli programlamayı da desteklemektedir.
- Geniş kütüphanelere sahiptir. Bununla birlikte görüntü işleme kütüphaneleri mevcuttur.

Python programla dili ile şu işlemler gerçekleştirilebilmektedir.

- Bir makineye öğrenme yeteneği vermek. (Machine Learning)
- Web uygulamaları oluşturmak. (Django)
- Bilimsel Fonksiyonlar geliştirmek
- Rest APİ oluşturmak
- Veri analizi yapmak
- Ağ ve Soket Programlamak
- Masaüstü uygulamaları oluşturmak

4.8 Görüntü İşleme Kütüphaneleri

Programlama dilleri yapılacak işlemlere özel birtakım kütüphanelere sahip olmaktadır. Bu kütüphaneler yapılacak olan işe göre birbirlerinden ayrılmaktadır. Tez çalışmasında görüntü işleme teknolojisine dayalı bir sistem tasarlandığı için kullanılacak programlama diliyle uyumlu görüntü işleme kütüphanelerinin araştırılması ve değerlendirilmesi gerekmektedir. En bilinen görüntü işleme kütüphanelerinin başında OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesi gelmektedir. Bu kütüphane 1999 yılında İntel tarafından geliştirilmeye başlanmıştır. Daha sonra Itseez, Willow, Nvidia, AMD, Google gibi çeşitli şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir.

OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, MacOS ve iOS platformlarında çalışabilmektedir. C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla da Visual Basic.Net, C# ve Visual C++

dilleri ile topluluklar tarafından geliştirilen farklı wrapperlar aracılığıyla Perl ve Ruby programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilmektedir. OpenCV kütüphanesi, içerisinde görüntü işlemeye ve makine öğrenmesine yönelik 2500'den fazla algoritma bulunmaktadır. Bu algoritmalar ile yapılabilecekler şu şekildedir;

- Yüz tanıma,
- Nesneleri ayırt etme,
- İnsan hareketlerini tespit edebilme,
- Nesne sınıflandırma,
- Plaka tanıma,
- Üç boyutlu görüntü üzerinde işlem yapabilme,
- Görüntü karşılaştırma,
- Optik karakter tanımlama

Bu görüntü işleme kütüphanelerine ilaveten, tam bir alternaternatif kütüphane olarak sunmak doğru olmasa da görüntü işlemeye dayalı birçok temel algoritmayı bulundurmasından dolayı "Matlab" ismi verilen uygulama da alternatif olarak gösterilebilmektedir. Bu uygulama dilinin dördüncü nesil ve çok amaçlı bir programlama dili olduğu bilinmektedir. Nispeten, performansın çok önemli olmadığı durumlarda temel görüntü işlemleri için kullanılmaktadır. Görüntü işlemeye uygun algoritmalara sahip ticari Matlab programı, OpenCV Kütüphanesi ile etkileşimli olarak da çalışabilmektedir.

Alternatif görüntü işleme kütüphanelerine ilaveten birçok kütüphane daha sayılabilmektedir. Fakat, bu kütüphaneler yapılacak çalışmaya özel geliştirilmesi sebebiyle, amaca yönelik tercih edilmektedir. "Endrov", "ImageJ", "Lead tools", "Pink", "Image Magick", "Boost" isimli kütüphaneler bu amac yönelik geliştirilen görüntü işleme kütüphanelerinden bazıları olarak gösterilebilmektedir.

Sonuç olarak, OpenCV, görüntü işleme kütüphaneleri arasında en popüler ve en çok kullanılan kütüphane olarak yer almaktadır. Yazılım geliştiriciler için bir kütüphanenin, teknolojinin popülerleşmesinin temel sebebi o teknoloji hakkındaki erişilebilecek kaynak çeşitliliği olduğu bilinmektedir. Açık kaynak kodlu olması

sebebiyle doğrudan desteği bulunmayan programlama dilleri için ara katmanlar ilgili geliştiriciler tarafından oluşturularak OpenCV bu dile entegre edilebilir olduğu görülmektedir. Geniş işletim sistemi desteği bulunması itibariyle de geliştiriciler için, platformlar arası uygulama geçişini kolaylaştırmaktadır.

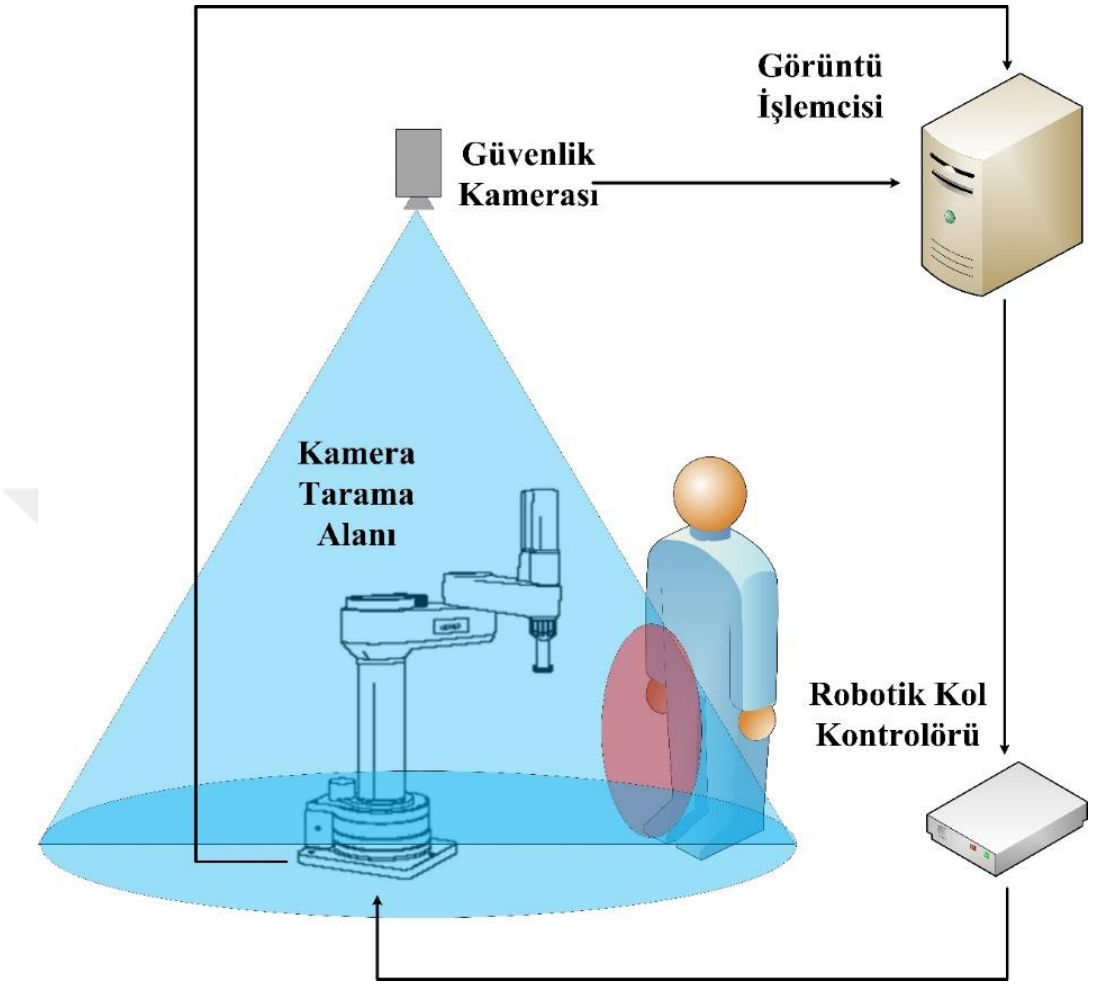
Mevcut programlama dilleri ve görüntü işleme kütüphaneleri düşünüldüğünde, görüntü işleme yazılımları için performansın önemli bir kriter olduğu bilinmektedir. Bu yazılımlar, yüksek işlemci, grafik kartı ve bellek ihtiyacı duymaktadırlar. Bundan dolayı kullanılan programlama dili ve seçilen platform, yazılımın performansı için kritik öneme sahiptir. Önceki dönemlerde görüntü işleme projelerinde C ve C++ programlama dillerinin çalışmalarda yüksek oranda kullanıldığı bilinmektedir. Çünkü bu diller düşük seviyeli olması sebebiyle rakiplerine göre oldukça göstermektedir. Fakat, programlama dillerinin çeşitliliğinin artmasıyla birlikte, birçok dil açık kaynak olmuştur. Ek olarak toplulukların veya firmaların desteği ile yapılan iyileştirmeler sayesinde performanslarında iyileştirilmeler yapılmıştır. Dolayısıyla, Java programlama dili JVM'in (Java Virtual Machine) geliştirilmesiyle yazılan kodu daha iyi optimize edebilir hale gelmiştir. Bu sebeple, Java programlama dili de C++ programlama dili kadar performanslı çalışan kodlar üretmeye başlamıştır.

Java programlama dili, platform bağımsızlığı, orta seviye bir dil olması ve proje geliştirmenin oldukça hızlı olması sebebiyle önce çıkmaktadır. Bu bağlamda OpenCV uygulaması geliştirmede Java tercih edilmektedir. Son birkaç yıldır ise Python yazımının kolay olması, uygulama geliştirme süresini kısaltması gibi özellikleri ile programlamaya yeni başlayan kişiler tarafından kullanılmasının popüler olduğu bilinmektedir.

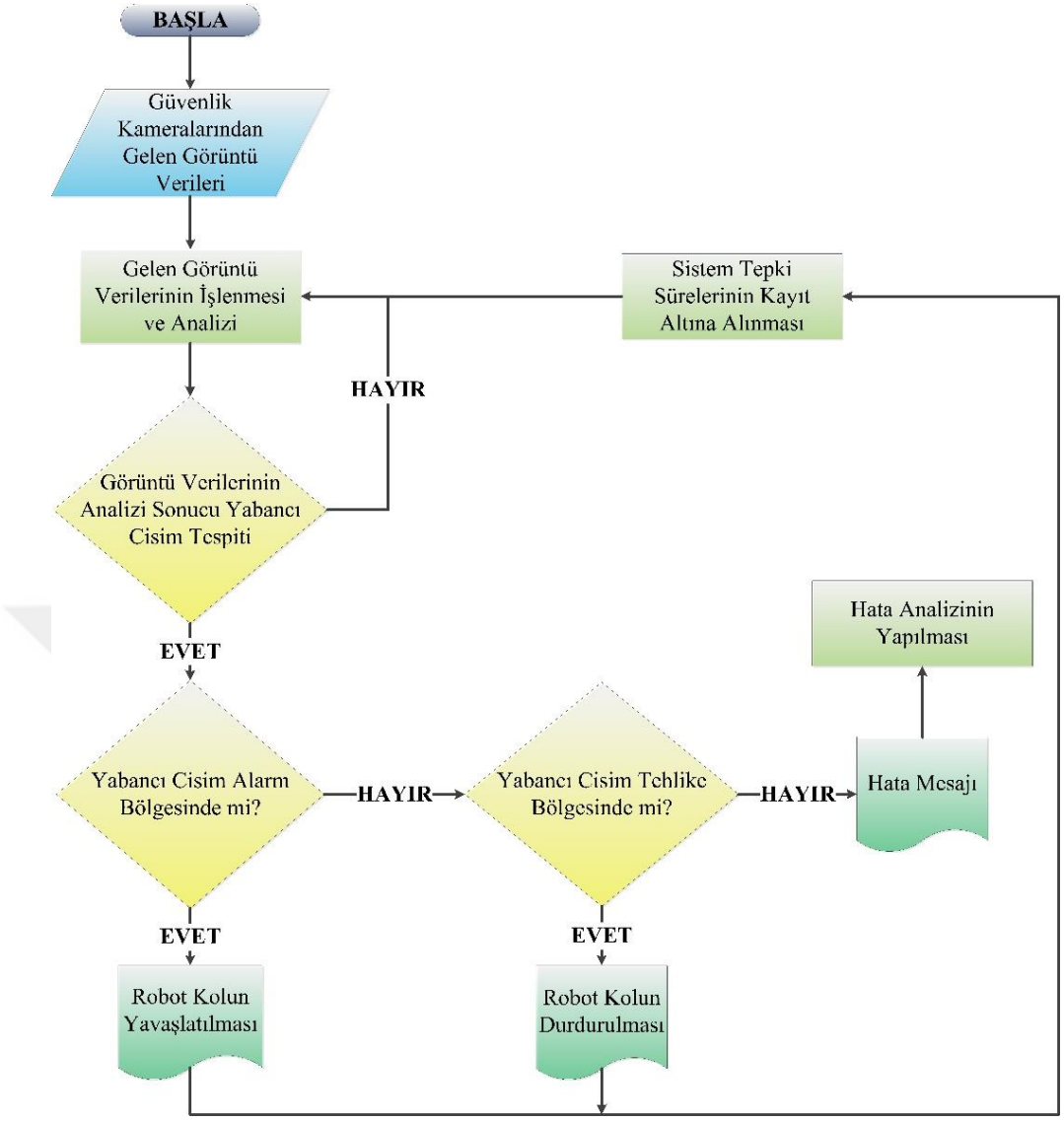
4.9 Görüntü İşlemeye Dayalı Güvenlik Sistemi Yazılımı

Önceki bölümlerde tasarımı yapılmış iş hücresinin matematiksel modele uygun şekilde gerçeklenmesinden sonra görüntü işlemeye dayalı güvenlik yazılımının oluşturulması konusunda öncelik programlama dili ve görüntü işleme kütüphanesine karar vermektir. Programlama dili olarak Python seçilmiştir. Şekil 4.15'de güvenlik sisteminin şematik gösterimi yer almaktadır. Şekil 4.16'de güvenlik

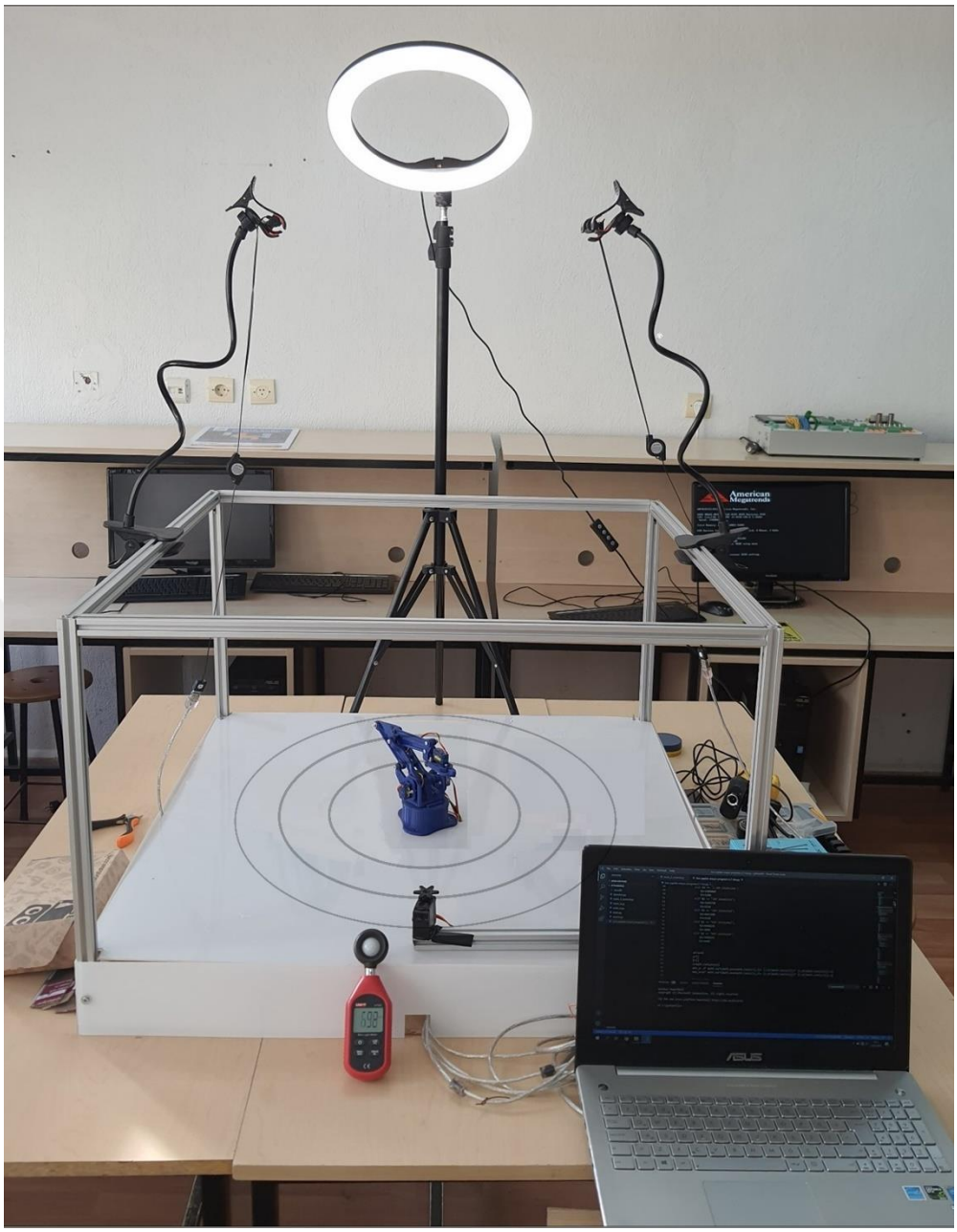
sisteminin akış şeması yer almaktadır. Şekil 4.17'de güvenlik sisteminin gerçeklenmiş deneysel düzeneği görülmektedir.



Şekil 4.15 Güvenlik sisteminin kontrol şeması.



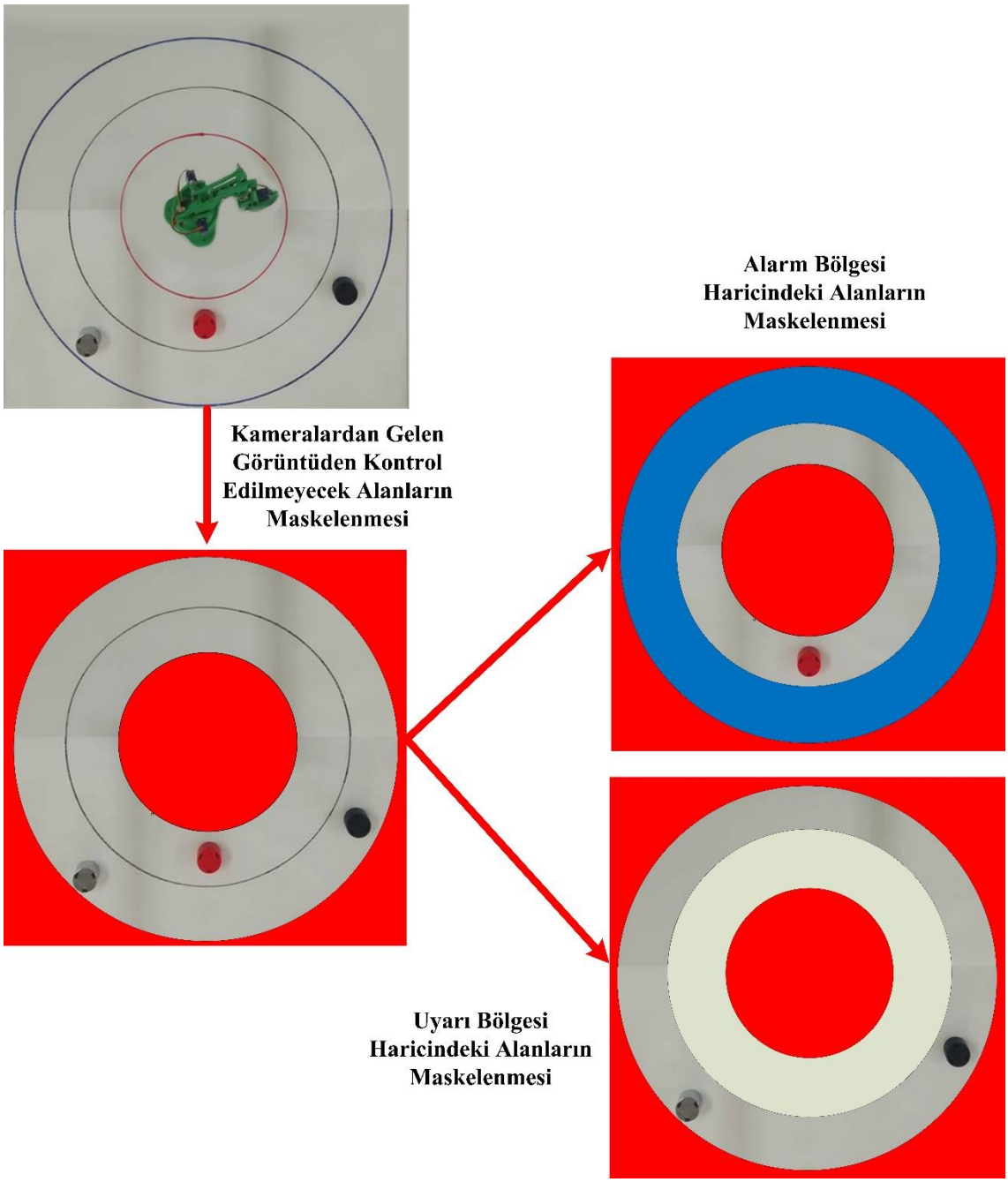
Şekil 4.16 Güvenlik sistemi akış diyagramı



Şekil 4.17 Matematiksel modele göre tasarımı yapılmış iş hücresi

Önceki bölümlerde kameradan gelen görüntülerin maskelenerek birleştirilmesi detaylı olarak anlatılmıştır. Şekil 4.18'de birleştirilmiş olan kamera görüntülerinin güvenli alan haricinde kalan kısımlarının maskelenmesi gösterilmiştir. Görüldüğü üzere iş hücresi görüntüsü kameralardan dikdörtgen şekilde alınmaktadır. Dolayısıyla robot kol çalışma uzayı haricinde gereksiz bölümlerinde anlık görüntüleri dâhil olmaktadır. Bu durum kurulacak güvenli alan mesafesi dışında, güvenliğin gerekmediği alanlarda da değişiklik olması durumunda sistemde bozucu etki olmasına yol açmaktadır. Sonuç olarak bu görüntünün hareket tespit işlemine

alınmadan önce gerekli olmayan bölümlerinin maskelenmesi gerekmektedir. Şekil 4.18'de güvenli alan dışında kalan kısımların maskelenmiş görseli görülmektedir.

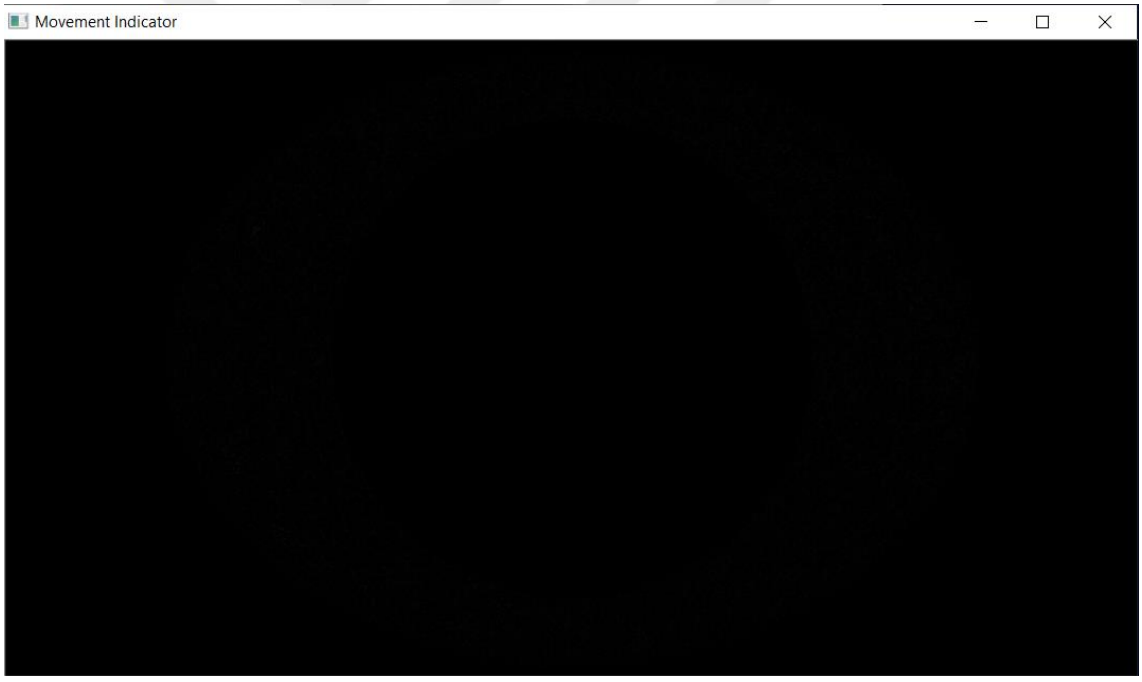


Şekil 4.18 Güvenli alan dışında kalan kısımların maskelenmesi

Yukarıdaki Şekil 4.18'de dış görüntüler görüntü filtreleme kütüphaneleriyle maskelenmiş olsa da ortada yer alan robot kolun anlık çalışması sistem için gereksiz bir bozucu etki olmasına yol açacaktır. Dolayısıyla robot kolun erişebileceği alanlar da sistemden çıkarılarak. Arada kalan güvenlik mesafesinde hareket tespiti

yapılması, sistem için en uygun durum olmaktadır. Sonuç olarak robot kol çalışma uzayının da maskelenmiş hali şekilde verilmiştir.

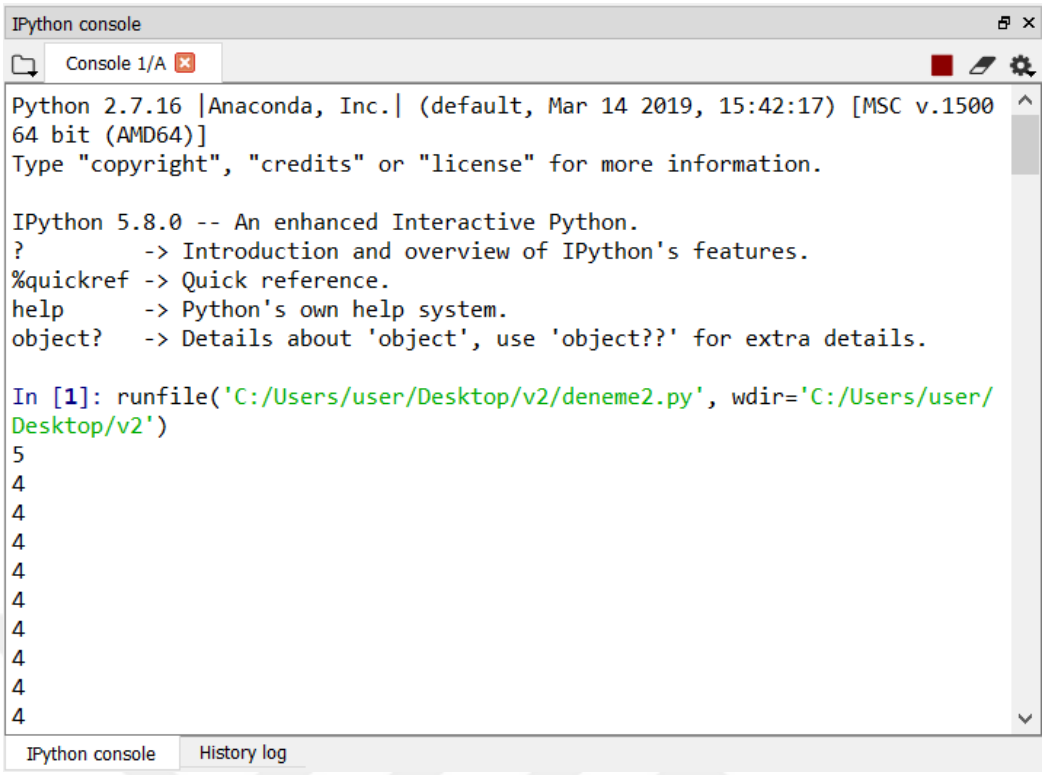
Gerekli maskeleme işlemleri yapıldıktan sonra anlık görüntü belirli aralıklarla sürekli olarak alınmaktadır. Bu görüntüden ardışık olarak 3 adet alınıp birbirleriyle karşılaştırılmasıyla görüntü içerisinde bir hareket olduğu algılaması yapılmaktadır. Bu algılama piksel sayısına bağlı olduğu için ve en uygunu bulma programında taranacak piksel sayısı verilmiş olduğu için buna uygun hassasiyet değeri belirlenmektedir. Sonrasında bu farklar tekrar maskelenerek hareketin tespit edildiği pikseller beyaz, hareketsiz olan alanlar siyah olacak şekilde yeni bir görüntü oluşturulmaktadır. Bu görüntü "Movement Indicator" isimli pencerede verilmektedir. Şekil 4.19'de bu görüntü penceresi görülmektedir. Hareketsiz durumlarda düz siyah olarak verilen görüntü hareket tespiti olduğunda beyaz piksellerle gösterilmektedir.



Şekil 4.19 Hareket tespitinin olmadığı durumlar için "movement indicator" görseli

Hareketin algılandığı durumlarda ise hareketin tespit edildiği pikseller beyaz, hareketsiz olan alanlar siyah olacak şekilde yeni bir görüntü oluşturulmaktadır. Tespit edilen obje belirlenen eşik değerinden büyük olduğu durumlarda robot kolun hareketini önce yavaşlatması sonra durdurması gerekmektedir. Robot kolun hareketinin yavaşlatılabilmesi ve durdurulabilmesi için robot kol eklemlerinde yer

alan motor kontrolcüsüne bir sinyalin gönderilmesi gerekmektedir. Robot kol eklemlerine yerleştirilmiş olan motorların kontrolü ve sürücü devresi olarak Arduino tercih edildiği için seri port üzerinden bir veri paketi gönderilmiştir. Python programında hareket tespit edilmesi durumunda seri port üzerinden eğer tespit edilen cisim uyarı bölgesinde ise yavaşlatma sinyali, eğer alarm bölgesinde ise de durdurma sinyali gönderilmiştir. Seri porta gönderilen bilgilerin anlık takibini yapabilmek için koda ana konsol ekranında da aynı bilgilerin yazılması sağlanmıştır. Böylelikle hareket tespiti durumunda sistemin anlık iletimleri gözle de görülmektedir. Şekil 4.20'de Python programının konsol ekranı görülmektedir. Bu ekranda program çalışmaya başladıktan sonra tek sütun olarak alt alta 5 ve 4 bilgileri akmaktadır. Burada 5 bilgisi uyarı bölgesinde hareket tespit edildiğini, 4 bilgisi ise alarm bölgesinde hareket tespit edildiğini göstermektedir. Şekilde görüldüğü üzere seri port üzerinden 4 bilgisinin Arduino mikrodenetleyicisine iletilmesi üzerine mikrodenetleyici de yer alan kod robot kol eklemlerinde yer alan motorların durdurulmasını sağlamaktadır. Bu noktada programın devamlı kontrol edilebilmesi için duraklama belirli bir süre olarak belirlenebilir veya operatör müdahelesi gelene kadar tamamen durdurulabilmektedir. Görüntü işlemeye dayalı güvenlik yazılımı ve robot kol eklemlerinde yer alan motorların kontrolü için yazılmış olan Arduino kodları tezin ekler bölümünde yer almaktadır.

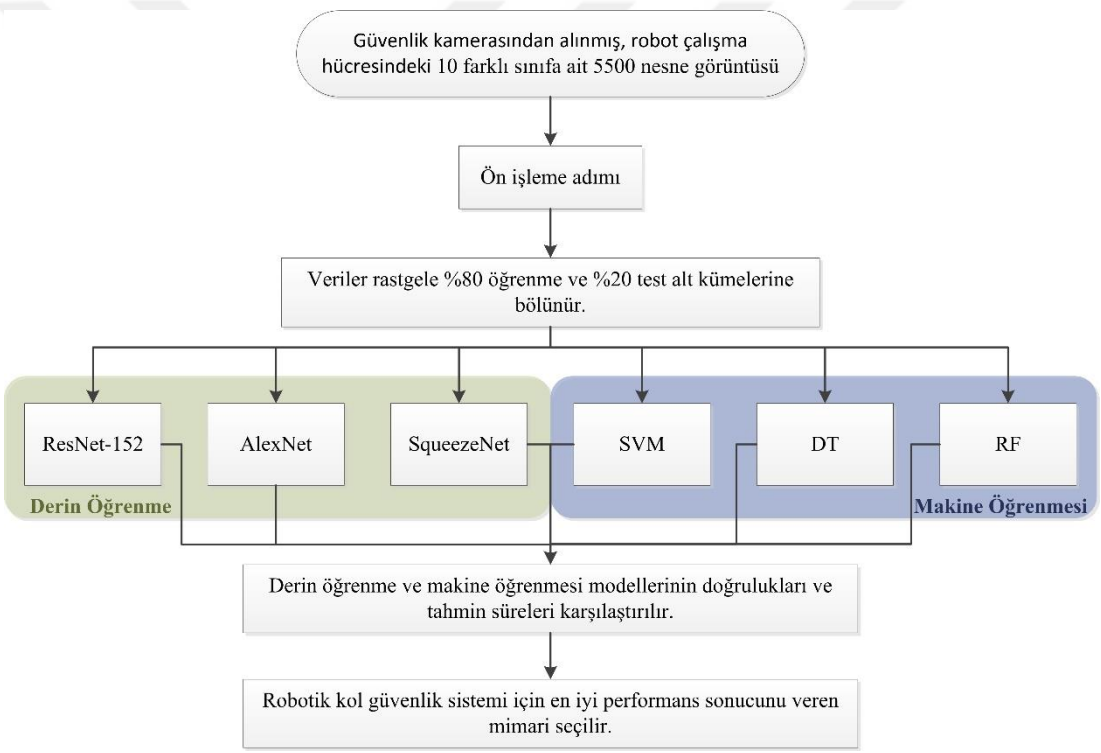


Şekil 4.20 Python programının konsol ekranı

5
GÜVENLİK SİSTEMİNDE YAPAY ZEKÂ YAKLAŞIMI

5.1 Yapay Zekâ Tabanlı Nesne Tanıma ve Sınıflandırma

Veri seti, deneysel bir robot iş hücresinin güvenlik kamerasından alınan görüntülerle oluşturulmuştur. İlk olarak veri seti ön işleme aşamasına tabi tutulmuştur. Daha sonra yapay zeka algoritmalarını eğitmek için kullanılmış ve sonuçlar karşılaştırılmıştır. Tasarlanan görüntü sınıflandırma sisteminin işlem adımları, Şekil 5.1'deki akış şemasında gösterilmiştir.



Şekil 5.1 Tasarlanan görüntü sınıflandırma sisteminin işlem adımları

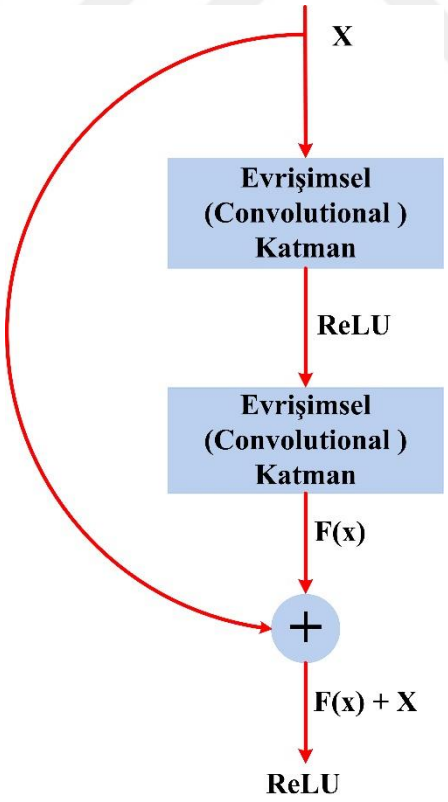
ResNet-152 modeli, 151 evrişim katmanı dahil olmak üzere 152 eğitilebilir katmandan oluşmaktadır [49]. 7x7 çekirdek boyutuna sahip 3x3 üst pooling katmanıyla ImageNet'te sunulan en derin öğrenme mimarilerinden biridir [50, 51]. Klasik CNN mimarilerinde derinlik arttıkça ResNet mimarisinde artık blok yaklaşımı ile aşırı öğrenmenin önüne geçilmektedir.

"Residual unit i" için, y_i girdidir ve f_i eğitilebilir doğrusal olmayan eşlemelerdir. "Residual unit i" çıktısı özyinelemeli olarak denklem 5.1'de tanımlanmıştır [52].

$$y_i = f_i(y_{i-1}.w_i) + y_{i-1} \tag{5.1}$$

Yukarıdaki denklemde, w_ieğitilebilir parametreleri belirtir ve f genellikle iki veya üç yığın evrişim adımıdır. Tam etkinleştirme öncesi sürümde, bir evrişim adımının bileşenleri sırasıyla bir toplu normalleştirme, bir doğrultulmuş doğrusal birim (ReLU) doğrusal olmama durumu ve bir evrişim katmanıdır [53, 54].

ResNet mimarisi, ileri beslemeli ağ katmanlarına eklenen bağlantı atlamalı daha derin bir ağ sağlamaktadır. Başka bir deyişle, kısayol bağlantıları ve özellikleri, F(x) işlevinden öğrenmek yerine, F(x)'e x (giriş değeri) eklenerek öğrenilir. ResNet öğrenme bloğu Şekil 5.2'de [55] gösterilmektedir. "Residual Unit" yaklaşımı ile çıktı doğrudan bloğa verilmez, ancak ReLU işleminden x + F(x) sonuçları verilir. ResNet'i diğer mimarilerden ayıran bu özellik ile katman sayısı arttıkça optimizasyon daha rahat hale gelir, tahmin doğruluğu artar ve kaybolan gradyan sorunu çözülmektedir [56].



Şekil 5.2 ResNet öğrenme bloğu

5.2 Veri Seti

Kurulan sistemin gerçek zamanlı çalışma şartlarının ve sistem tepkilerinin kontrollü bir şekilde incelenmesi gerekmektedir. Çalışmanın yapay zeka aşamasında sistemden on farklı sınıfa ait görüntüler alınmış ve Resnet-152 derin öğrenme mimarisi kullanılarak bir uygulama geliştirilmiştir. Uygulamada ilk olarak üç aşamadan oluşan veri ön işleme aşaması gerçekleştirilmiştir. Veri ön işlemenin ilk aşamasında, görüntüler yeniden boyutlandırılmıştır. Böylece çalışmada kullanılan tüm görseller aynı boyuta getirilmiştir.

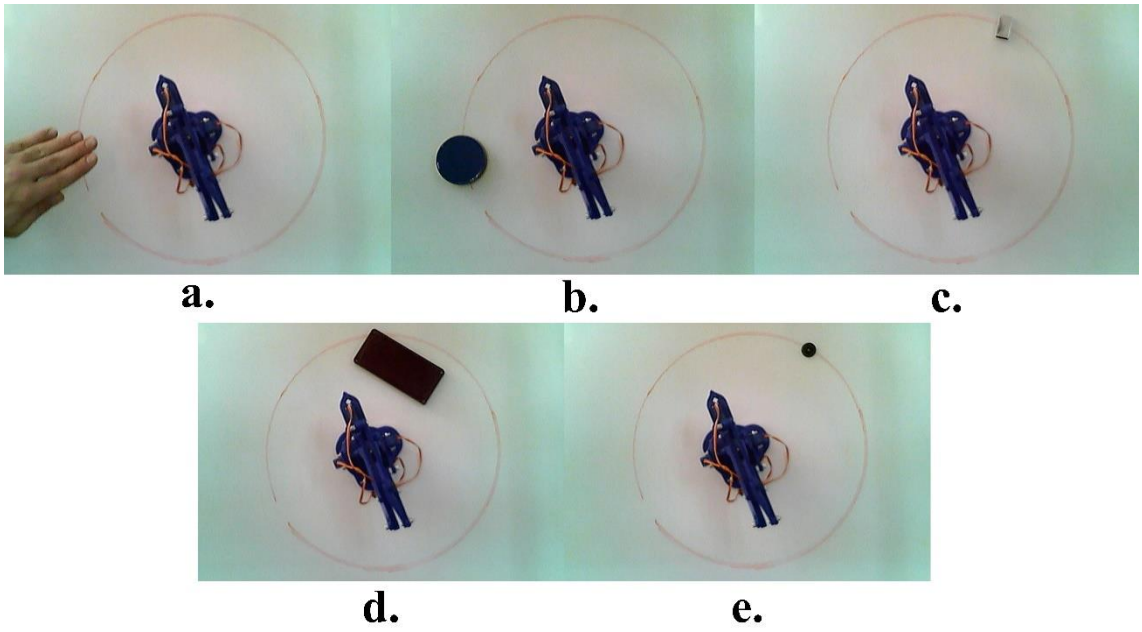
İkinci aşamada toplam 5500 fotoğraf görüntüsü on farklı sınıfa ayrılmıştır. Bu on sınıf kendi aralarında karşılaştırılmıştır. Karşılaştırmalar sırasında aşağıdaki etkilerle ilgili sorular sorulmuştur:

- Nesne boyutu
- Nesne parlaklığı
- Nesne geometrisi
- Net olmayan nesne köşeleri
- Kamera çözünürlüğü
- Nesne ve arka plan renklerinin yakınlığı
- Gölgeler
- Nesneden ışık yansımaları.

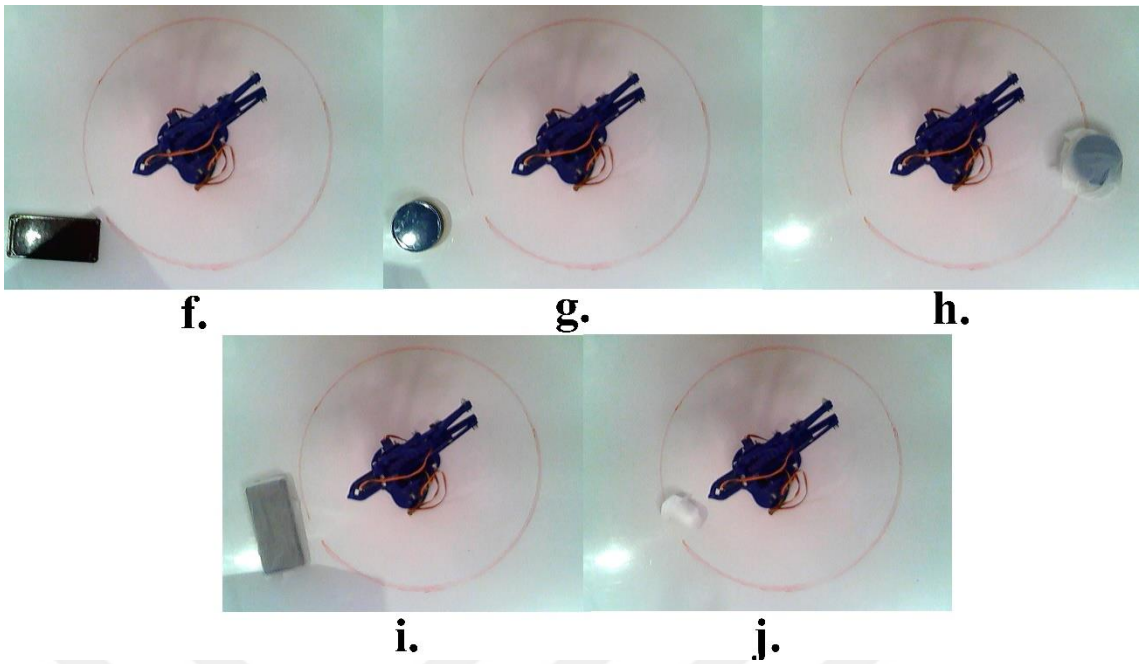
Görüntü çözünürlüğü, görüntünün ayrıntı düzeyini ifade etmektedir. Bu terim dijital görüntüler, hareketli görüntüler ve diğer görüntü türleri için geçerlidir. Görüntü çözünürlüğü birkaç yolla ölçülebilir. Dijital kameraların çözünürlüğü piksel, uzaysal, spektral ve zamansal çözünürlükler gibi çeşitli şekillerde tanımlanabilmektedir [46, 47]. Sistemde kullanılan kamera yerden 90cm yüksekliğe yerleştirilmiştir. Kamera çözünürlüğü 1280x960, ardından bir pikselin gerçek boyutu $0.142 mm^2$'dir. Kamera çözünürlüğünü artırmak, her pikselin gerçek boyutunu azaltır. Analiz edilen görüntüdeki nesne boyutları ve karşılık gelen piksel sayısı Tablo 5.1'de verilmiştir.

Şekil 5.3 ve Şekil 5.4, çalışma alanına giren on yabancı nesnenin örneklerini göstermektedir. Görsel-a, bir operatörün elinin girişini göstermektedir. Görsel-b, büyük, koyu mavi dairesel bir girişi göstermektedir. Görsel-c, metal ve dikdörtgen

nesneleri göstermektedir. Görsel-d, siyah ve büyük bir dikdörtgeni göstermektedir. Görsel-e, küçük, yuvarlak ve siyah bir nesnenin girişini göstermektedir. Görsel-f, gölgeli ve yansıtıcı bir dikdörtgenin girişini göstermektedir. Görsel-g, gölgeli ve yansıtıcı koyu mavi dairesel bir girişi göstermektedir. Görsel-h, düzensiz köşeleri olan yuvarlak bir şekli göstermektedir. Görsel-i mat ve gölgeli bir dikdörtgeni göstermektedir. Görsel-j, arka planla yakın renkte bir nesneyi göstermektedir. Görseller, robotik çalışma alanına girebilecek farklı boyut ve yapılardaki nesneleri simüle etmek için seçilmiştir. Sınıflandırma yapılırken boyut, renk, geometri, parlaklık ve gölgenin etkisinin öğrenilmesi amaçlanmıştır.



Şekil 5.3 Birinci giriş yapan obje sınıfları



Şekil 5.4 İkinci giriş yapan obje sınıfları

Tablo 5.1 Nesnelerin gerçek ve piksel boyutu

Sınıflandırma	**Nesne Türü**	**Gerçek Boyutu (mm)**	**Piksel Sayısı (Yaklaşık)**
Dairesel Nesne	Küçük siyah nesne	Ø10	552
	Büyük siyah nesne	Ø65	23,301
	Büyük parlak yüzeyli nesne	Ø65	23,301
	Büyük mat yüzeyli nesne	Ø65	23,301
	Büyük düzensiz şekilli nesne	Ø65	23,301
Dikdörtgen Nesne	Küçük parlak yüzeyli nesne	25X10	1,756
	Büyük siyah nesne	120X55	46,344
	Büyük parlak yüzeyli nesne	120X55	46,344
	Büyük mat yüzeyli nesne	120X55	46,344
	Küçük düzensiz şekilli nesne	50X10	3,511

Her nesne sınıfı yaklaşık 550 fotoğraf görüntüsü ihtiva etmektedir. Bu fotoğrafların %10'u düşük kaliteli görüntülere aittir. Bu fotoğraflar kasıtlı olarak titreşim, tozlu hava ve flaş altında çekilmiştir. Öğrenme sırasında sınıflandırılacak etki türleri ve karşılaştırılacak görsel kümeler Tablo 5.2'de gösterilmiştir.

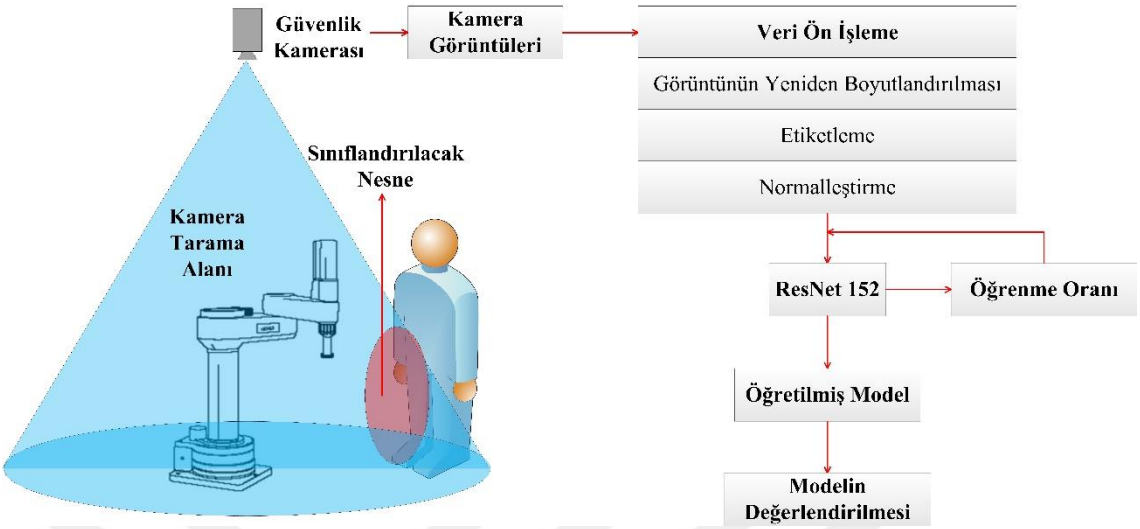
Tablo 5.2 Karşılaştırılan nesnelerin özellikleri

Sınıflandırma	**Etki Türü**	**Karşılaştırılan Nesneler**
Operatörün Eli	nesne geometrisi	a
	Nesne üzerine düşen gölge veya ışık	
Dairesel Nesne	nesne boyutu	b, e, g h
	nesne parlaklığı	
	nesne geometrisi	
	Bozulmuş nesne köşeleri	
	Nesne üzerine düşen gölge veya ışık	
Dikdörtgen Nesne	nesne boyutu	c, d, f, i, j
	nesne parlaklığı	
	nesne geometrisi	
	Bozulmuş nesne köşeleri	
	Nesne renginin ve arka plan renginin birbirine yakınlığı	
	Nesne üzerine düşen gölge veya ışık	

Veri ön işlemenin son aşamasında, tüm görüntüler 0 ile 1 arasına getirilerek hesaplama maliyetini minimize edecek şekilde normalizasyon yapılmıştır. İkinci aşamada veriler rastgele %80 eğitim ve %20 test alt kümelerine ayrılmıştır. Eğitim için en uygun öğrenme oranı değeri belirlenerek optimizasyon yöntemi parti büyüklüğü değeri 64 olarak belirlenmiş ve 50 döngüyle eğitilmiştir. Son aşamada ise eğitim aşamasından elde edilen model test verileri üzerinden değerlendirilmiştir.

Görüntü sınıflandırma algoritmasında, görüntü veri seti kullanılarak üç derin öğrenme ve üç makine öğrenme modülü eğitilmiştir. Derin öğrenme ve makine öğrenimi mimarileri, doğruluk ve yanıt süresi açısından karşılaştırılmıştır. Öncelikle Resnet-152, SqueezeNet ve AlexNet derin öğrenme mimarileri uygulanmıştır. Sonraki aşamada Destek Vektör Makinesi (SVM), Rastgele Orman (RF) ve Karar Ağacı (DT) makine öğrenme algoritmaları kullanılmıştır. Çalışmada kullanılan veri seti görsellerden oluşmaktadır. Bu nedenle makine öğrenmesi algoritmalarında kullanılan görüntüler tek boyutlu bir vektöre (birinci dereceden tensör)

dönüştürülerek eğitim işlemleri gerçekleştirilmiştir. Resnet-152 için eğitim aşaması Şekil 5.5'de gösterilmektedir.

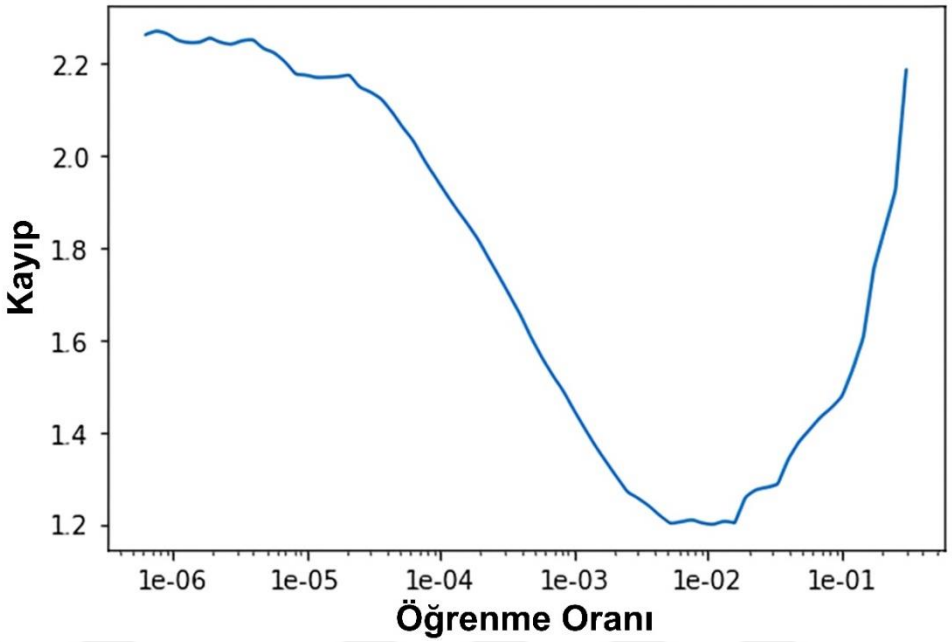


Şekil 5.5 Sınıflandırma için iş akışı diyagramı

Çalışmada yapay zeka algoritmaları Python programlama dilinde yazılmıştır. Eğitim süreci bulut servisleri üzerinde derlenmiştir. GoogleColab sistemindeki GPU işlemci ve Tensorflow kütüphanesi kullanılarak eğitim süreci tamamlanmıştır. En iyi performans gösteren yapay zeka mimarisi keşfedildikten sonra, belirli bir robotik iş hücresinin görüntü tabanlı güvenlik sisteminin GPU'suna entegre edilebileceği öngörülmektedir.

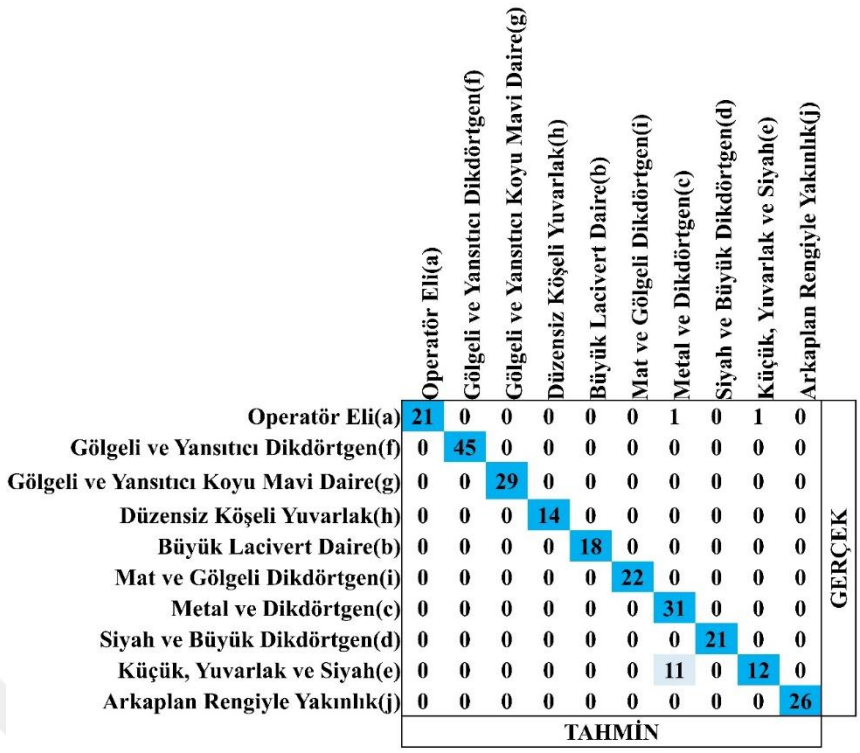
5.3 Yapay Zekâ Mimarilerinin Karşılaştırılması ve Modellerin Sonuçları

Resnet-152 için öğrenme oranı değeri en yüksek doğruluğa sahip modeli seçecek şekilde belirlenmiştir. Şekil 5.6, öğrenme hızı eğrisinin optimal değerlerinin 0,00001 ile 1 arasında olduğunu göstermektedir. Görüntüler üzerinde eğitim, bu aralık içinde minimum kaybı üreten 0,001'de gerçekleştirilmiştir.



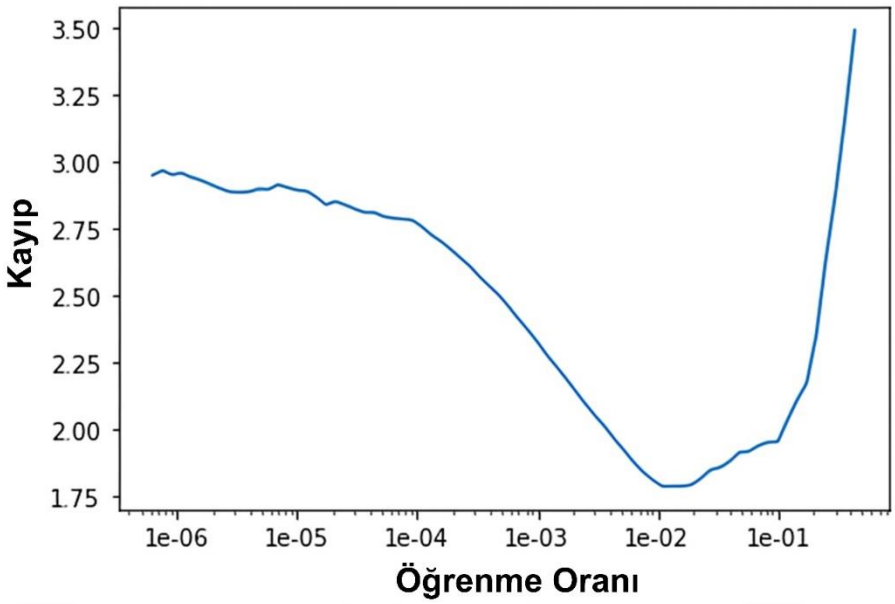
Şekil 5.6 Resnet-152 mimarisi için öğrenme oranı tablosu

Eğitim sonrasında modelin doğruluğunu belirlemek için bir karmaşıklık matrisi oluşturulmuştur (Şekil 5.7). On sınıf görselin kendi aralarında Tablo 5.2'ye göre karşılaştırmasını göstermektedir ve yüksek doğruluk elde edilmiştir. Bu karmaşıklık matrisleri, modellerde aşırı uydurma (over fitting) olup olmadığını belirlemek için kullanılmıştır. Kurulan sistem ve mimari uyumun iş güvenliği açısından yeterli olduğu görülmektedir.

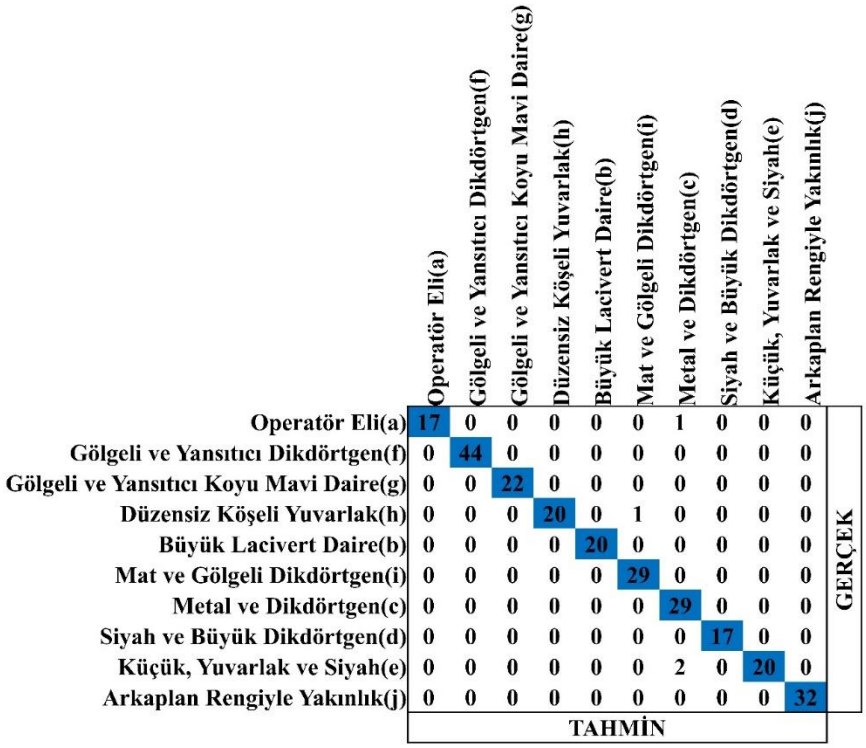


Şekil 5.7 Resnet-152 modelinin on nesneli karmaşıklık matrisi

Daha sonra SqueezeNet derin öğrenme mimarisi kullanılarak eğitim süreci gerçekleştirilmiştir. Öğrenme oranı optimizasyonu yöntemi kullanılarak, küçük bir orandan başlanarak, küçük parti büyüklükleri ve her parti için kayıp değerleri kaydedilen yüksek bir öğrenme oranı elde edilene kadar eğitime devam edilmiştir. Kaydedilen öğrenme hızı kayıtları çizilmiş ve aşağı eğimin en keskin olduğu bölgenin orta noktası alınarak nihai öğrenme hızı değeri belirlenmiştir. Şekil 5.8, öğrenme oranının 0.000001 ile 1 arasında değiştiğini göstermektedir. Model, kaybın sıfıra en yakın olduğu 0,004'lük bir öğrenme oranı ile eğitilmiştir. Eğitim aşamasından sonra modelin doğruluğunu belirlemek için bir karmaşıklık matrisi oluşturulmuştur (Şekil 5.9). Buna göre SqueezeNet mimarisi, veri seti üzerindeki nesneleri %99 doğrulukla algılamaktadır.



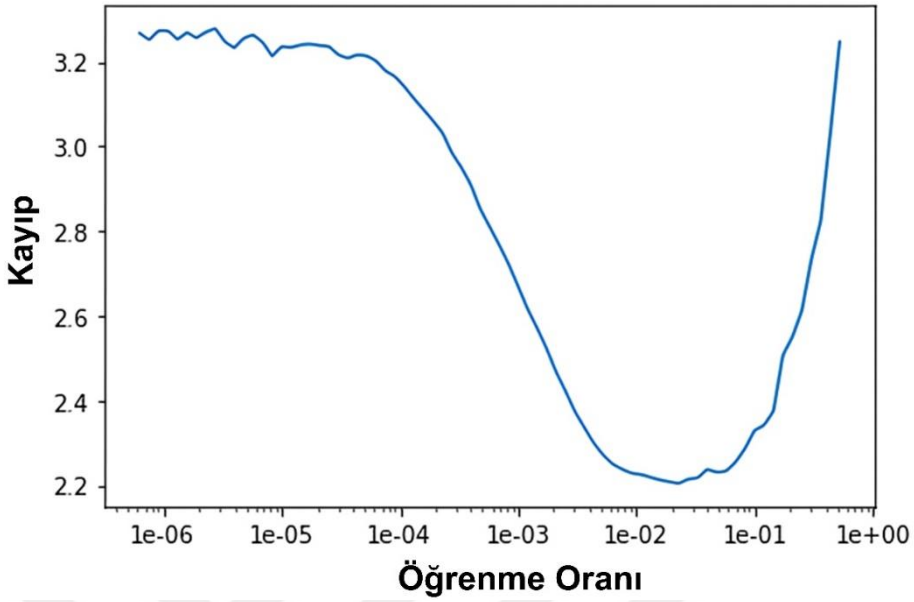
Şekil 5.8 SqueezeNet mimarisi için öğrenme oranı tablosu



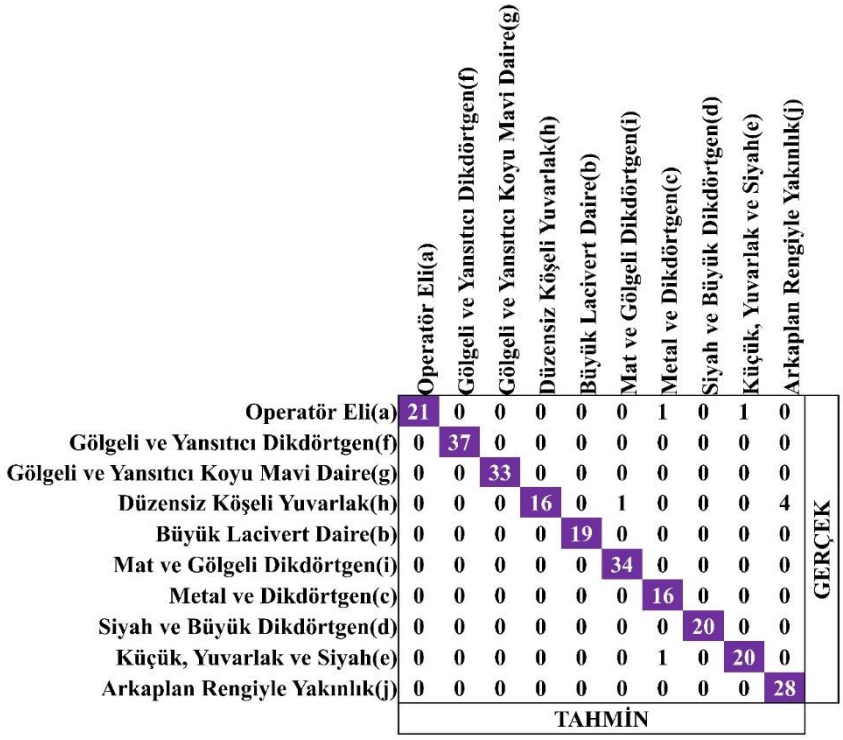
Şekil 5.9 SqueezeNet modelinin on nesneli karmaşıklık matrisi

Eğitim, üçüncü derin öğrenme modeli olan AlexNet ile veri seti üzerinde gerçekleştirilmiştir. Şekil 5.10 öğrenme oranının 0.000001 ile 1 arasında değiştiğini göstermektedir. Model, minimum 0,005 kayıpla bir öğrenme hızında eğitilmiştir. Eğitim sonrasında modelin doğruluğunu belirlemek için tekrar bir karışıklık matrisi

oluşturulmuştur (Şekil 5.11). Sonuç olarak, AlexNet modülü, veri setinde %97 doğrulukla nesneleri tespit etmiştir.



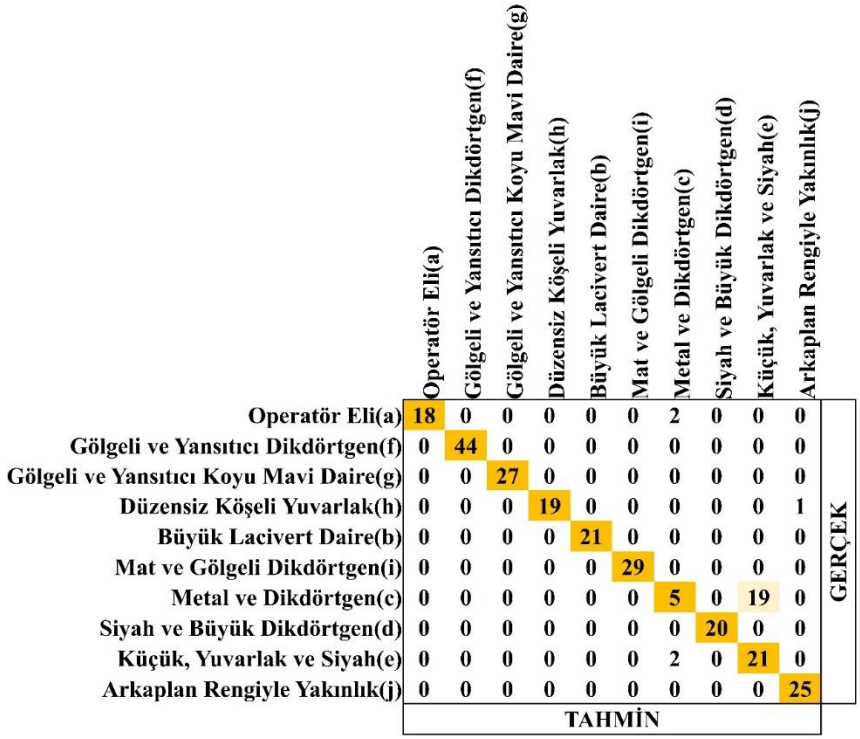
Şekil 5.10 AlexNet mimarisi için öğrenme oranı tablosu



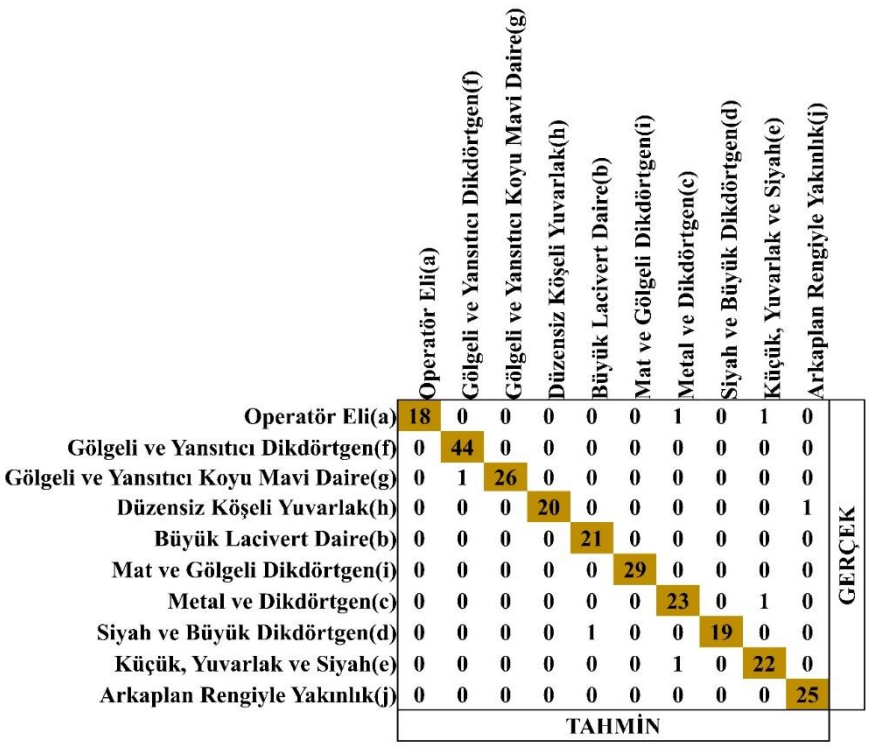
Şekil 5.11 AlexNet modelinin on nesneli karmaşıklık matrisi

Makine öğrenmesi test aşamasında, Destek Vektör Makinesi (SVM), Rastgele Orman (RF) ve Karar Ağacı (DT) modüllerini eğitmek için aynı veri seti kullanılmıştır. SVM modülünün eğitim sonuçlarından elde edilen karmaşıklık matrisi Şekil 5.12'de

verilmiştir. Algoritma, veri kümesindeki nesneleri %91 doğrulukla algılamıştır. RF makine öğrenme modülünün karışıklık matrisi Şekil 5.13'te verilmiştir. Sonuç olarak, RF algoritması veri seti üzerindeki nesneleri %98 doğrulukla tespit etmiştir.



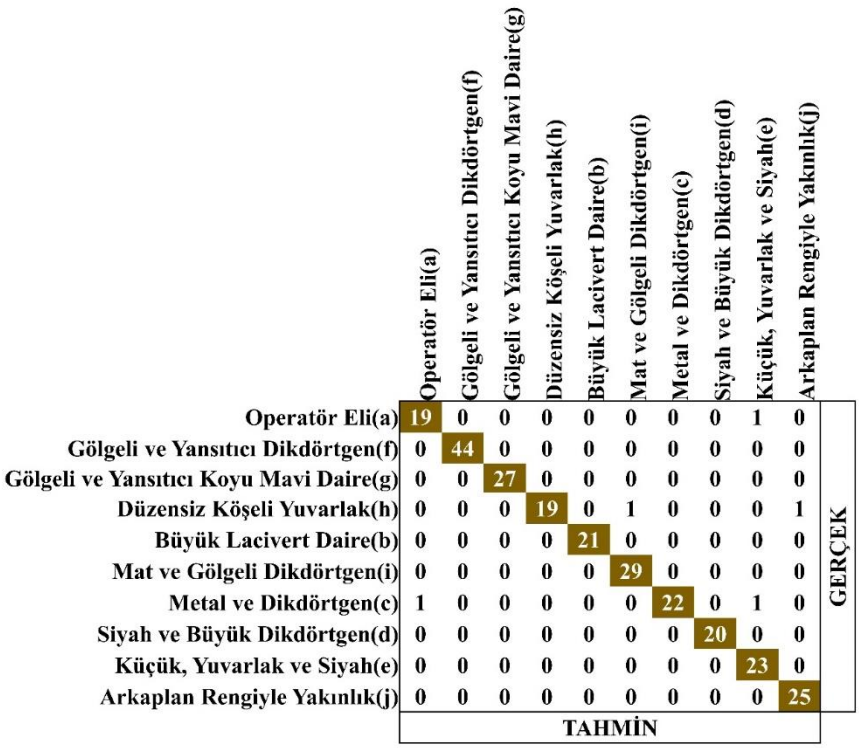
Şekil 5.12 SVM modelinin on nesneli karmaşıklık matrisi



Şekil 5.13 RF modelinin on nesneli karmaşıklık matrisi

Son olarak, DT makine öğrenmesi algoritması test edilmiştir. Eğitilen modülün karmaşıklık matrisi Şekil 5.14'te verilmiş olup, DT algoritması veri seti üzerindeki nesneleri %98 doğrulukla tespit etmiştir.

Çalışmada kullanılan tüm yapay zeka modellerinden elde edilen sonuçlar Tablo 5.3'te karşılaştırılmıştır. SqueezeNet'in %99 doğruluk oranı ile en başarılı algoritma olduğunu göstermektedir. AlexNet bunu %97 ile takip etmiştir. ResNet-152 %95 doğruluk vermiştir. Çalışmada kullanılan her üç derin öğrenme mimarisinin de %95'in üzerinde performans elde etmesi, veri seti ile mimarilerin uyumlu olduğunu göstermektedir.



Şekil 5.14 DT modelinin on nesneli karmaşıklık matrisi

Tablo 5.3 Kullanılan mimarilerin test sonuçları

Model	**Doğruluk (Accuracy)**	**Hassasiyet (Precision)**	**Keskinlik (Sharpness)**	**F1-Puanı (Score)**
Resnet-152	%95	%96	%95	%95
AlexNet	%97	%97	%96	%96
SqueezeNet	%99	%99	%99	%99
SVM	%91	%91	%91	%90
RF	%98	%98	%98	%98
DT	%98	%97	%97	%98

Makine öğrenimi algoritmaları arasında RF ve DT, %98 doğrulukla eşit derecede en başarılı mimariler olarak bulunmuştur. SVM'nin %91 doğruluk çalıştığı görülmüştür. RF'nin DT ve SVM'den daha başarılı olmasının temel nedeni daha entegre bir yapıya sahip olmasıdır.

Çalışmanın bir sonraki aşamasında, derin öğrenme ve makine öğrenmesi algoritmalarının test verileri üzerinde tahmin süreleri hesaplanmış ve sonuçlar Tablo 5.4'te karşılaştırılmıştır.

Tablo 5.4 Veri seti üzerinde derin öğrenme ve makine öğrenmesi algoritmalarının tahmin süreleri

Model	**Süre** (sn)
ResNet-152	4.39
AlexNet	3.91
SqueezeNet	3.94
SVM	2.39
RF	0.035
DT	0.015

Öğrenme algoritmaları, en başarılı derin öğrenme modeli olan SqueezeNet'in tahmin süresi 3,94 saniye iken, makine öğrenmesi algoritmaları arasında en başarılı DT algoritması, 0,015 saniye ile test verilerine en az sürede tahmine dayalı yanıt veren modeldir. Bu nedenle çalışmada kullanılan veri setinin makine öğrenmesi algoritmalarından biri olan DT ile en başarılı sonucu verdiği tespit edilmiştir.

Çalışmada kullanılan derin öğrenme mimarilerinin yanıt sürelerinin makine öğrenmesi algoritmalarına göre çok daha yüksek olmasının temel nedeni, derin öğrenme algoritmalarında kullanılan mimarilerin yanıt sürelerinin hesaplama maliyetlerinin yüksek olmasından kaynaklanmaktadır. ResNet-152, derin öğrenme mimarileri arasında en yüksek tahmin süresine sahiptir. Bunun nedeni, AlexNet ve SqueezeNet mimarilerinden daha fazla katman sayısıdır. Makine öğrenmesi algoritmalarından SVM makine öğrenmesi algoritmasının RF ve DT mimarilerine göre daha yüksek yanıt süresine sahip olmasının temel nedeni vektörel hesaplamalardan kaynaklanan hesaplama maliyetidir.

Derin öğrenme tabanlı SqueezeNet modeli, daha başarılı özellik çıkarımı nedeniyle daha yüksek doğruluğa ulaşmıştır. Ayrıca, makine öğrenimi tabanlı RF ve DT algoritmaları, %1'lik önemsiz bir farkla benzer doğruluğa yaklaşmıştır. Bununla birlikte, makine öğrenmesi tabanlı algoritmaların, derin öğrenme tabanlı algoritmalardan 100 kat daha hızlı olduğu kanıtlanmıştır. Bu nedenle, DT'nin verilen görüntü verileriyle en iyi performansı gösterdiği bulunmuştur.

Görüntü tanımada sürekli olarak en yüksek doğruluk, kesinlik ve keskinliği veren RF modeli, 0.035 saniye ile en hızlı ikinci modeldir. Bu yanıt süresi, daha iyi bir F1 puanı ile daha sağlam bir davranışın beklendiği endüstriyel bir ortamda da kabul edilebilir. Öte yandan, SVM, derin öğrenme modülleriyle aynı büyüklük sırasına göre bir hızda, ancak %90'lık çok daha düşük bir F1 puanıyla performans göstermiştir.

6
DENEY DÜZENEĞİ TASARIMI

6.1 Deney Düzeneğinin Oluşturulması

Kurulan sistemin gerçek zamanlı çalışma şartlarının ve sistem tepkilerinin kontrollü bir şekilde incelenmesi gerekmektedir. Dolayısıyla kontrollü bir deneme sistemi kurulması gerekmektedir. Bunun için öncelikle kontrol edilmek istenen niceliğe karar verilmelidir. Bizim sistemimizde en önemli sorulardan bir tanesi; Yabancı cismin kontrol edilen alana girmesi durumunda robot kolun yavaşlama veya durma işlemleri için tepki süresinin ne olduğudur, fakat, bu durum kontrol alanına giren cisim ile doğrudan ilişkili olduğu için giren cisimin hızının kontrol altında olması gerekmektedir. Dolayısıyla bu sistem için kontrol edilebilir hızda bir yabancı cisim tasarımının yapılması gerekmektedir.

Bu tezde gerçekleştirilmiş olan uygulama için kontrol edilebilir hız için bir servo motor tercih edilmiştir. Dönel hareket gerçekleştiren bir servo motorun sürülmesi esnasında yazılımsal olarak her saniyede kaç derecelik bir dönme yapabileceği belirlenebilmektedir. Dolayısıyla aşağıdaki denklemden de görülebileceği gibi açısal değişimin zamansal değişime oranı bize açısal hızı vermektedir.

$$\omega = \frac{\Delta\theta}{\Delta t} \tag{5.1}$$

Yukarıda görülen denklemde; ω, açısal hızı, Δθ açısal değişimi, Δt, zamansal değişimi ifade etmektedir. Bununla birlikte açısal hızı bilinen ve dairel hareket yapan cismin dairesel hareketin merkezine olan uzaklığı biliniyorsa, bu cismin çizgisel hızı hesaplanabilmektedir. Aşağıda görülen denklemler çizgisel hızın denklemini vermektedir.

$$v = \frac{2\pi r}{T}, \omega = \frac{2\pi}{T}, v = \left(\frac{2\pi}{T}\right) r = \omega . r \tag{5.2}$$

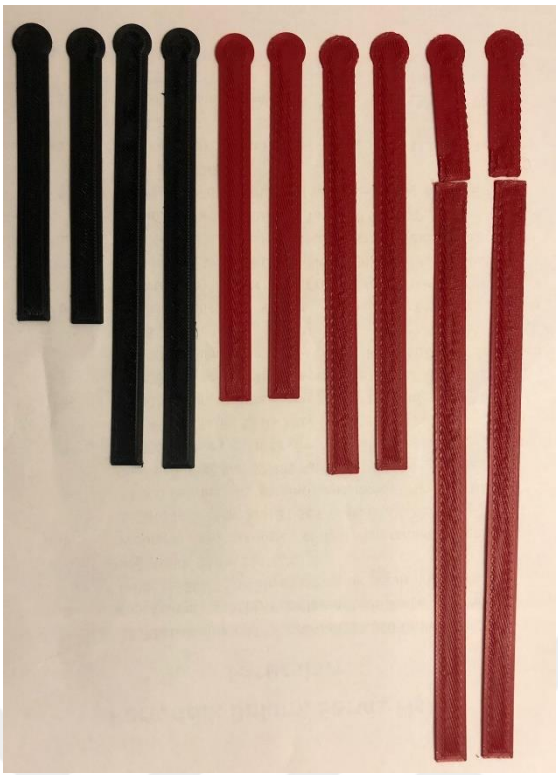
Yukarıda görülen denklemde; T, periyodu, v, çizgisel hızı, r, dairesel hareketin merkezine olan uzaklığı ifade etmektedir. Dolayısıyla bu denklemlerden

yararlanarak yazılımsal olarak motorun bir derecelik hareketi için geçecek süre ayarlanabilmektedir. Kontrol edilmek istenen ana parametre cismin çizgisel hızı olduğu için hangi hızlarda işlem yapacağımızı belirlememiz gerekmektedir. Ardından bu cismin dairesel hareketin merkezine olan uzaklığı belirlenmelidir. Bu uzaklık 5cm olarak ayarlanmıştır. Aşağıdaki tabloda motor sürücü yazılımında bir derecelik açı değişimi için gerekli olan süreler listelenmiştir.

Tablo 6.1 Deney düzeneğinde kullanılacak cismin çizgisel hız, açısal hız ve kontrol programındaki karşılık süreleri

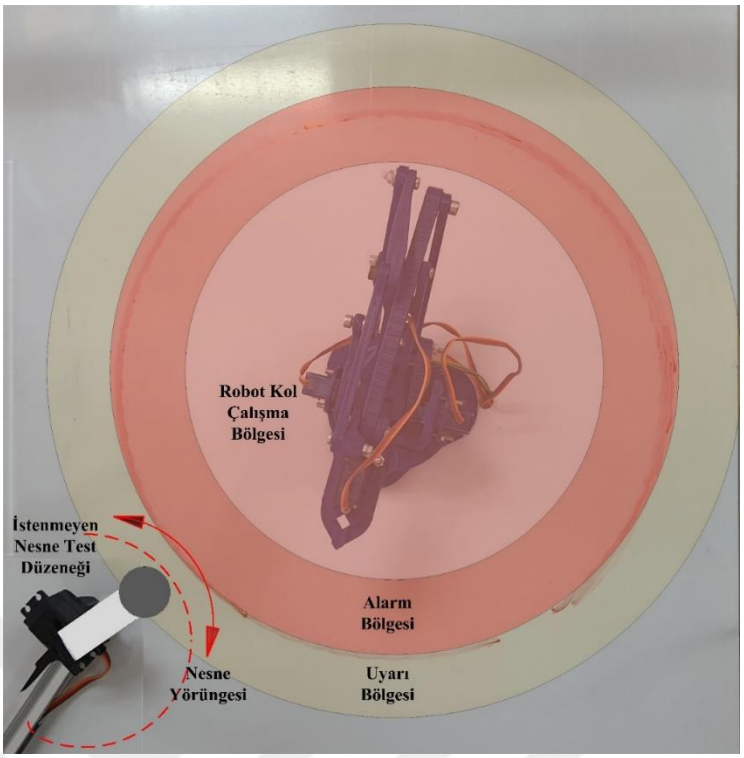
Yabancı cisim alana girme hızı (v) (m/s)	**Dairesel hareket yapan cismin açısal değişimi ($\Delta\theta$) (rad)**	**İstenen çizgisel hız için açısal değişime karşılık gelen zamansal değişim (Δt) (ms)**
0.1	1°, 0.01745	8.7
0.05	1°, 0.01745	17.4
0.025	1°, 0.01745	34.8
0.01	1°, 0.01745	87
0.005	1°, 0.01745	174
0.0025	1°, 0.01745	349
0.001	1°, 0.01745	872

Yukarıda verilmiş olan tabloda yeralan zamansal değişim motor sürücü yazılımında kullanılarak cisim için istenen hızda kontrollü bir hareket sağlanabilmektedir. Bunun için motor tahrik mili merkezinden 5cm uzaklıkta bir cisim yerleştirmek gerekmektedir. Bu cismin yapacağı hareketi robot kol kontrol alanının sınırına yerleştirildiğinde yabancı cisim istenen hızda kontrol alanına girmesi sağlanmaktadır. Daha yüksek hızlar istendiği durumlarda motor miline bağlı olan parçanın uzunluğu arttırıldığında, bu parça ucunda oluşacak çizgisel hız da artacaktır. Bunun için aşağıdaki Şekilde 6.1'de görüldüğü gibi kontrollü çizgisel hız için farklı boyutlarda motor miline bağlanacak aparatlar üç boyutlu yazıcıda basılmıştır.

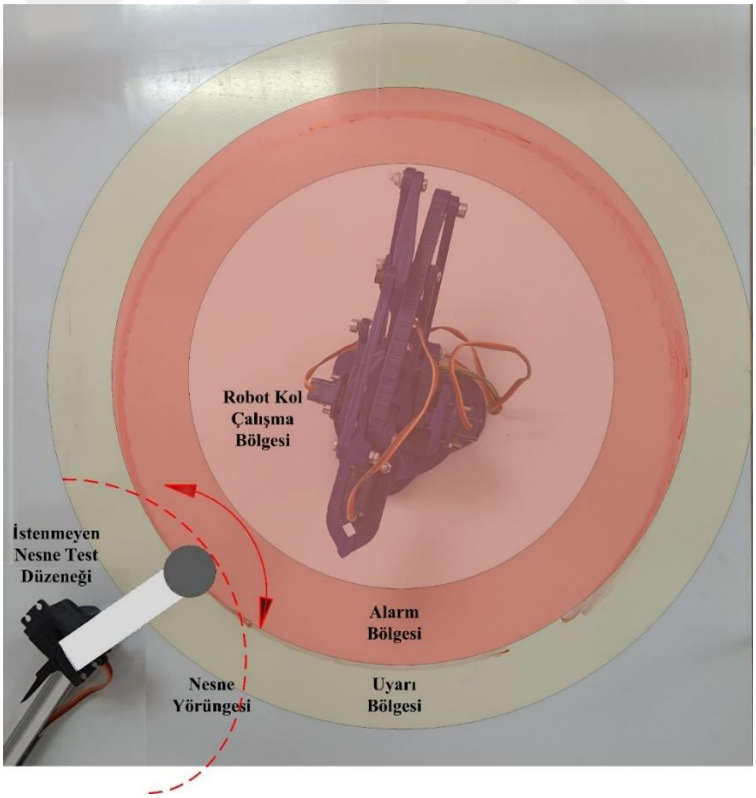


Şekil 6.1 Kontrollü hız deneyi için üç boyutlu yazıcıda basılmış farklı boyutlardaki aparatlar

Şekil 6.1'de tasarlanmış olan aparatlar bir servo motor miline bağlanmış ve hız kontrolü sağlanmıştır. Şekil 6.2'de yer alan hız kontrollü cisimin tamamı kontrol alanına girmemektedir. Cisimin tam uç noktasında istenen çizgisel hıza ulaşılmış ve bu uç kısmın kontrol alanına girmesi sağlanmıştır. Şekil 6.2 ve Şekil 6.3'de robot kol çalışma bölgesi, alarm bölgesi ve uyarı bölgesi dairesel renkli alanlar olarak gösterilmiştir. İstenmeyen hız kontrollü nesnenin bu alanlara girme senaryosu görsellerin sol altına yerleştirilmiş ve dairesel bir hareket yaparak çizilen yörüngede ilgili bölgeye girmesi sağlanmıştır. Şekil 6.2'de uyarı bölgesine giren cisim görülmektedir. Bu girme hareketi ile beraber robot kolum yavaşlaması sağlanmıştır. Şekil 6.3 de alarm bölgesine giren cisim görülmektedir. Bu bölgede giriş olduğu takdirde robot kolun durdurulması amaçlanmış ve gerçekleştirilmiştir.



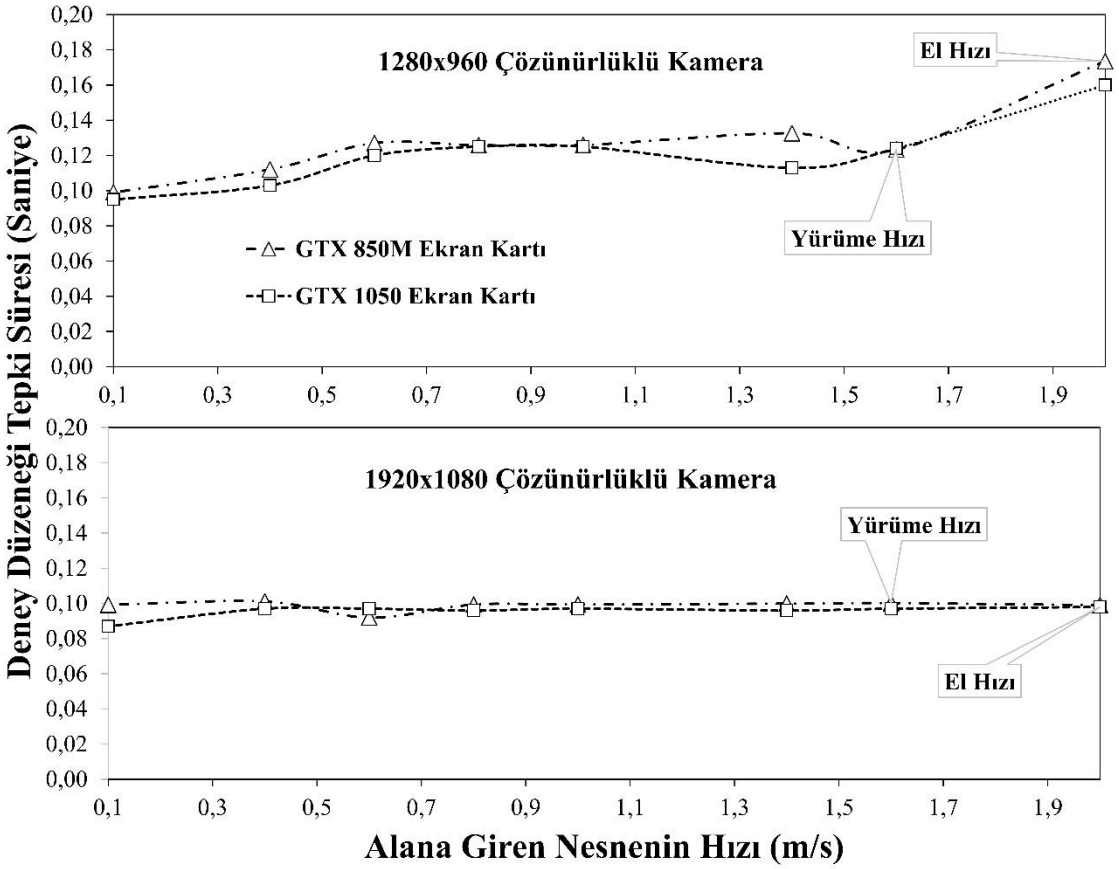
Şekil 6.2 Hız kontrollü objenin uyarı bölgesine giriş yörüngesi ve emniyet alanları



Şekil 6.3 Hız kontrollü objenin alarm bölgesine giriş yörüngesi ve emniyet alanları

6.2 Hız Kontrollü Yabancı Cismin Sisteme Etkisi

Deney sistemi tasarlanmış olup hızı kontrol edilebilen bir mekanizmanın robot kolun kamerayla kontrol edilen çalışma uzayına girmesi durumunda vereceği tepkinin süresi ölçülmüştür. Farklı hızlarda kontrol edilen alana giriş yapan yabancı nesneye karşılık sistem farklı sürelerde cevap vermiştir. Bu kontrollü hız ve sistem tepki süresi Şekil 6.4'de verilen grafiklerde gösterilmektedir. Şekillerdeki tüm obje, izlenen alanların içine girmemiştir. Cismin tam uç noktasında istenilen lineer hıza ulaşılmış ve bu uç parçanın kontrol alanına girmesine izin verilmiştir. Robot kol kamera kontrollü çalışma alanına girdiğinde hızı kontrol edilebilen bir mekanizmanın tepki süresi ölçülmüştür. Bu grafiklerdeki eğriler üzerinde "Yürüme Hızı" (2m/s) ve "El Hızı" için gösterilen iki işaretli nokta bulunmaktadır. (1,6m/s) Bu hızlar uluslararası standartlarca tanımlanmış hızlardır.



Şekil 6.4 İki farklı grafik işlemci ve kamera çözünürlüğü kullanılan deneysel sistemin istenmeyen kontrollü hıza sahip nesnelere karşı tepki süresi grafiği

Yukarıdaki ve aşağıdaki grafik, sırasıyla düşük ve yüksek çözünürlüklü kameralar kullanan güvenlik sistemlerinden elde edilmiştir. Her iki grafikte de NVIDIA Geforce GTX 850M ve GTX 1050 dahil olmak üzere iki farklı grafik kartının performansı karşılaştırılmıştır. GTX 1050'nin kıyaslama testlerine göre GTX 850M'den yaklaşık %60 daha yüksek performansa sahip olduğu görülmektedir. Tüm deneyler, 300lux sabit aydınlatma seviyesinde gerçekleştirilmiştir.

Kamera çözünürlüklerinin sistem yanıtına etkisinin yüksek olduğu görülmektedir. Düşük çözünürlüklü bir kamera kullanırken, sistem yanıtı farklı hız girişlerinde daha kararsızken, yüksek çözünürlüklü bir kamerayla daha kararlı çalıştığı görülmektedir. Öte yandan, grafik işlemcinin yüksek çözünürlüklü kameralarla sistem yanıtı üzerinde önemli bir etkisi (0,01s'den az) yoktur. Düşük çözünürlüklü kameraların kullanıldığı sistemde, yüksek performanslı grafik işlemcinin düşük performanslı olana göre ortalama %10 daha kısa sürede tepki verdiği gözlemlenmiştir. Bu nedenle önerilen sistem 1080p çözünürlüklü kameralar kullanıldığında düşük performanslı işlemcilerde bile verimli sonuçlar vermiştir. Dolayısıyla güvenlik sistemi için en az 1080p çözünürlüğe sahip bir kamera kullanılmalısı önerilmektedir. Optimum değerlere sahip bir güvenlik sistemi kurulduğunda, sistem yanıt süresi 0,08 ile 0,1s arasında değişmektedir. Bu değerler etkin bir çalışma için yeterli kabul edilmektedir. Grafik işlemciler karşılaştırıldığında sistem belleği (RAM) ve sabit diskin de etkili olduğu göz ardı edilmemelidir. Bu çalışmadaki karşılaştırmalarda her iki bilgisayar da 16GB RAM ve 256GB SSD sürücüye sahiptir.

6.3 Aydınlanma Şiddetinin Sisteme Etkisi

Kurulan sistemin gerçek zamanlı çalışma şartlarının ve sistem tepkilerinin kontrollü bir şekilde incelenmesi gerekmektedir. Dolayısıyla kontrollü bir deneme sistemi kurulması gerekmektedir. Bunun için önemli kriterlerinden bir diğeri; Yabancı cismin kontrol edilen alana girmesi durumunda robot kolun durma veya yavaşlama işlemlerinde kamera algılamasının hangi aydınlanma şiddetinde tepki süresinin ne olduğudur. Dolayısıyla bu sistem için kontrol edilebilir aydınlanma şiddetleri altında denemelerin yapılması gerekmektedir. Bunun için aydınlanma şiddetini belirlemek gerekmektedir. Bir metre yarıçaplı kürenin merkezinde bulunan bir

mum ışığı (candela) şiddetindeki aydınlanma kaynağının küre yüzeyinde oluşturduğu aydınlanma şiddetine lux adı verilmektedir [57, 58]. Aydınlanma şiddeti ölçen bir sensörle (Şekil 6.5) kurulmuş olan deney sisteminde, farklı aydınlanma şiddetlerinde tepki süresi ölçümleri yapılmış olup, bu değerler Şekil 6.6'de grafik olarak sunulmuştur.

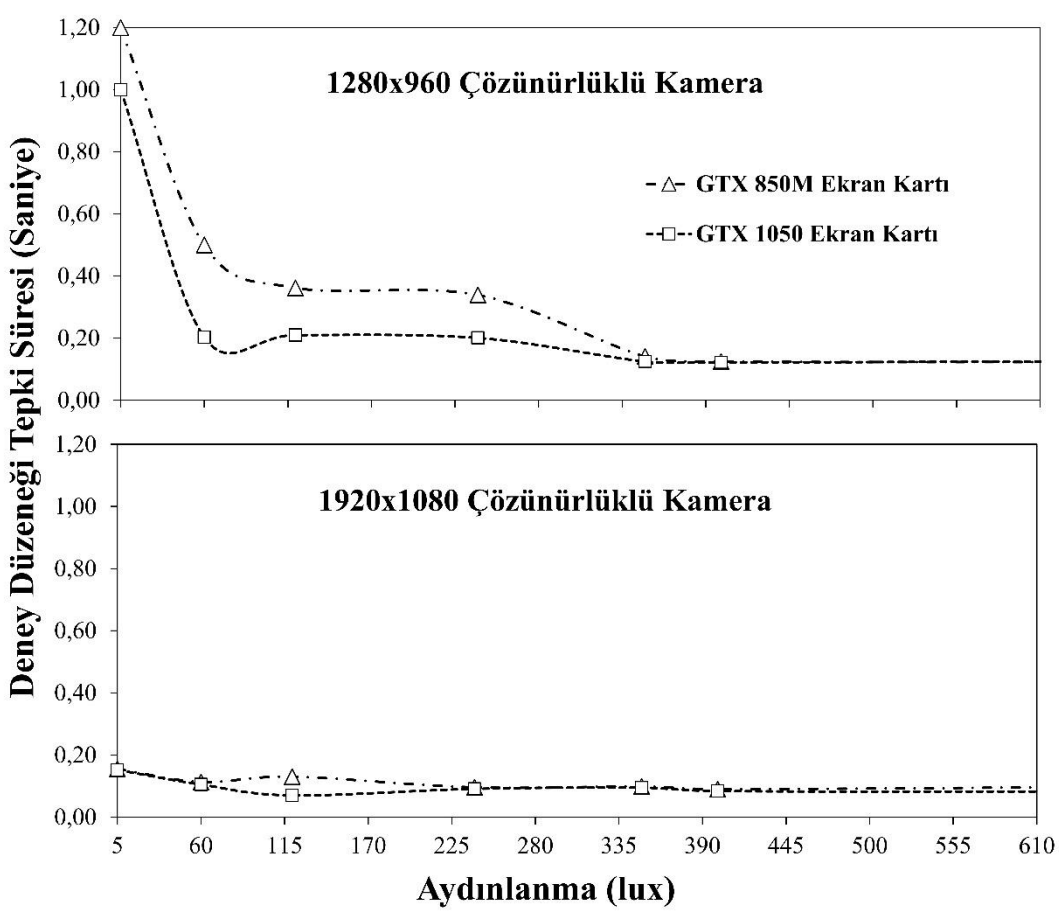


Şekil 6.5 Luxmetre

Şekil 6.6'daki yukarıdaki ve alttaki grafikler, sırasıyla düşük ve yüksek çözünürlüklü kameralar kullanan güvenlik sistemlerinden türetilmiştir. Her iki grafikte de NVIDIA Geforce GTX 850M ve GTX 1050 dahil olmak üzere iki farklı grafik kartının performansı karşılaştırılmıştır. GTX 1050'nin kıyaslama testlerine göre GTX 850M'den yaklaşık %60 daha yüksek performansa sahip olduğu bilinmektedir. Grafik işlemcisi, yüksek aydınlatma seviyelerinde sistem yanıt sürelerini önemli ölçüde etkilemediği gözlemlenmiştir. Aynı zamanda şekildeki grafiklerde iki farklı çözünürlüğe sahip kameralar ile aynı deneyler yapılmıştır. Kamera çözünürlüğünün, farklı aydınlatma seviyelerinde sistem yanıtı üzerinde yüksek bir etkisi olduğu ortaya konmuştur. Bu grafikten de görüldüğü gibi, kameranın yabancı bir cismi algılaması için minimum parlaklığı 5lux'dur. Bu seviyede tespit süresi en üst seviyededir. Düşük çözünürlüklü kamera, algılama süresinin kısaldığını ve aydınlatma 120lux'a yükseldikçe performansın arttığını göstermektedir. Bu seviyeden sonra görülen dalgalanma, sistemin kararlılığı ile ilgilidir. Aydınlık

açısından, mevcut en düşük seviye 120lux'dur. Yüksek çözünürlüklü kamera, sistemin düşük aydınlatma seviyelerinde bile stabil çalıştığını gözlemlemiştir.

Ticari üretim alanlarının en az 300-400lux ışık gerektirdiği bilgisi literatürde yer almaktadır [34]. Aydınlanma seviyesinin daha net anlaşılabilmesi için karanlık ve bulutlu bir günde aydınlanma şiddeti 100lux olarak ölçülmektedir [59]. Bir konutun oturma odasının aydınlanma şiddeti ise 50lux olarak ölçülmektedir [60]. Verilen değerler göz önüne alındığında deneysel düzeneğimiz yeterli seviyededir. Daha düşük aydınlanma şiddetinde çalışılması istenen sistemlerde algılama yapması beklenen kameranın diyafram açıklığının yüksek olmasına dikkat edilerek, istenen sistem kurulabilir.



Şekil 6.6 İki farklı grafik işlemcisi ve kamera çözünürlüğünü kullanılan deneysel sistemin farklı aydınlanma şiddetlerine karşı tepki süresi grafiği

7
BULGULARIN DEĞERLENDİRİLMESİ

Birinci bölümde, doktora çalışmasının başlangıç aşamasında belirlenmiş olan araştırma sorularına çalışmanın sonucunda yanıt verilmiştir. İlk araştırma sorusu, robot kol kullanılan iş hücrelerinde mevcut yüksek maliyetli kameralı güvenlik sistemlerine alternatif, uygun maliyetli ve güvenirliliği arttırılmış bir sistem ortaya koyulabilir mi? Bu soruya mevcut yüksek maliyetli sistemlerin genellikle üç boyutlu kamera teknolojisinden yararlandığı bu sebeple derinlik analizi yapmasından kaynaklı yüksek işlemcilere ve yüksek çözünürlüklü kameralara ihtiyaç duyulması sebebiyle maliyetin arttığı görülmüştür. Dolayısıyla bu tür sistemlerde tek açıdan bakan tek kameranın sıklıkla kullanıldığı görülmüştür. Doktora çalışmasında tasarlanmış olan sistemde ise iki boyutlu ve uygun fiyatlı kameralarla, fakat, birden fazla kullanılarak aynı alanın birden fazla açıyla tarandığı için iş ve işçi güvenliğinin arttığı görülmektedir. Yüksek kapasiteli grafik işlemci devreleri ile çok daha hızlı ve güvenilir bir güvenlik sistemi alternatifinin kurulabileceği bu çalışma vasıtasıyla ortaya koyulmuştur.

İkinci araştırma sorusu, uygun maliyetli olan alternatif güvenlik sistemlerinde güvenirliliğe etkisi olan sistem değişkenleri nelerdir? Bu soru yapılan tez çalışmasında cevap bulunmuştur. Bu değişkenlerimiz;

- Robot kol sayısı ve çeşidi
- Kamera sayısı ve lens açısı
- Kamera yerleşim açısı
- Kamera yerleşim yüksekliği olarak belirlenmiştir.

Bu değişkenler bir matematiksel modelde ifade edilmiş ve optimizasyon yöntemleriyle değişkenlerin en uygun hali bulunmuştur.

Doktora tez çalışmasının üçüncü araştırma sorusu ise, sistem değişkenleriyle bir matematiksel model kurulup optimizasyon çalışması yapılarak, en uygun sistem tasarımı ortaya koyulabilir mi? Bu soruya cevap olarak matematiksel modeli

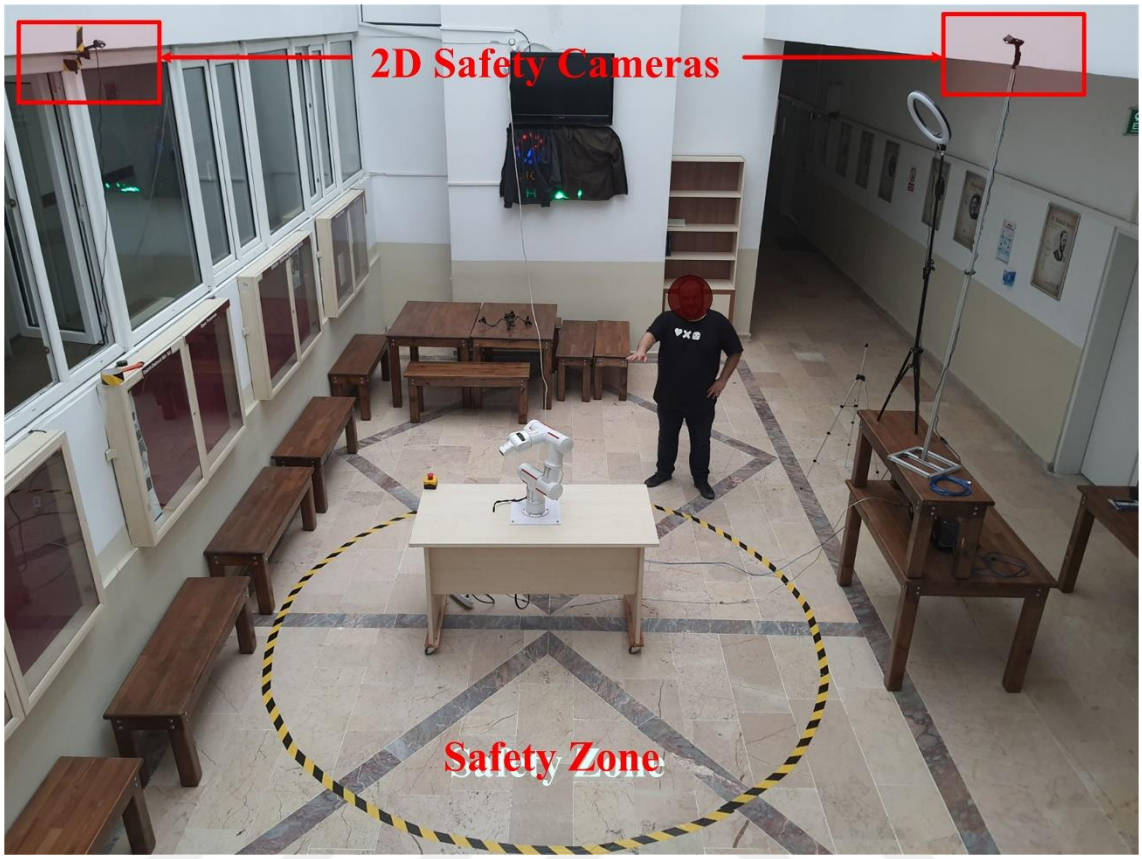
kurulmuş ve çözümü yapılmış olan problemin, gerçeklenmesi yapılarak karşılaştırması önceki bölümlerde yapılmıştır. Böylece en uygun sistem tasarımının ortaya koyulabileceği ispatlanmıştır.

Bu çalışmayı motive eden bir diğer araştırma sorusu, endüstriyel üretim alanında iş ve işçi güvenliği kontrol sistemlerinin maliyetlerin düşürülmesi ülkemiz üretim sanayi için de bir katma değer sağlayabilir mi? Bu soruya doktora tez çalışmasında hazırlanmış olan en uygunu bulma yazılımı ve kullanıcı dostu ara yüzüyle endüstriyel alanda kameralı güvenlik sistemleri kullanan iş hücreleri tasarlamak isteyen son kullanıcıya bir cevap bulunmuştur. Ticari olarak pazarlanabilir ve kullanılabilir olan bu yazılım tasarım yapmak isteyen herkese faydalı olacaktır. İlaveten yapılmış olan doktora tezinden türetilen yayınlar ile bulunmuş olan cevapların herkes tarafından ulaşılabilir ve kullanılabilir olması amaçlanmıştır.

Bu tez çalışmasında ele alınan diğer özel araştırma sorularında; Kamera sayısının, kamera yerleşim yüksekliğinin, kamera yerleşim açısının, kamera çözünürlüğünün güvenlik sistemi üzerindeki etkisi nedir? Kullanılan robot kol boyutlarının güvenlik sisteminin tasarımındaki etkinliğinde farklılıklar var mı? Kamera çözünürlüğü ve yerleşim konumu arasındaki ilişkiler nelerdir? Soruları sorulmaktadır. Bu sorulara da matematiksel model kurularak ve grafiksel olarak çözümlenerek cevap verilmiştir.

İki boyutlu görüntülerin birleştirilmesinde işlem maliyeti nasıl düşürülebilir? Geleneksel görüntü işleme yöntemleri yerine yapay zeka tabanlı mimarilerin etkisi nedir? Soruları da tez çalışmasının başında sorulmuştur. Bu sorulara da tezin ilgili başlıklarında algoritma ve testleri tamamlanarak karşılaştırmalı olarak cevap verilmiştir.

Önerilen güvenlik sistemini doğrulamak için, endüstriyel ölçüde bir robot kol ihtiva eden iş hücresinde güvenlik sistemi yeniden ölçeklendirilmiştir. Acrome-Acrobot marka robot kolun kullanıldığı iş hücresi için iki adet iki boyutlu kamera kullanılmıştır. Sistem Şekil 7.1'de görülmektedir. Sarı-siyah güvenlik şeridi, güvenlik bölgesini göstermektedir. Bu sistem test edildiğinde çok benzer tepki süresi sonuçları elde edilmiştir. Böylece, önerilen tasarımın daha büyük ölçekte de uygulanabilir olduğu kanıtlanmıştır.



Şekil 7.1 Büyük ölçekli bir robot çalışma hücresinde en uygun şekilde tasarlanmış güvenlik sisteminin uygulanması

Sonuç olarak, gerçekleştirilen deney düzeneğinde özetlenen hipotez doğrulanmıştır. Büyük ölçekli endüstriyel uygulamalar için iyileştirme önerileri sunulmaktadır. Ayrıca, ticari ürün ve tasarlanan güvenlik sistemi için maliyet karşılaştırmaları Tablo 7.1'de gösterilmiştir.

Tablo 7.1 Ticari bir örneğin ve tasarlanan deneysel düzeneğinin özelliklerinin karşılaştırılması

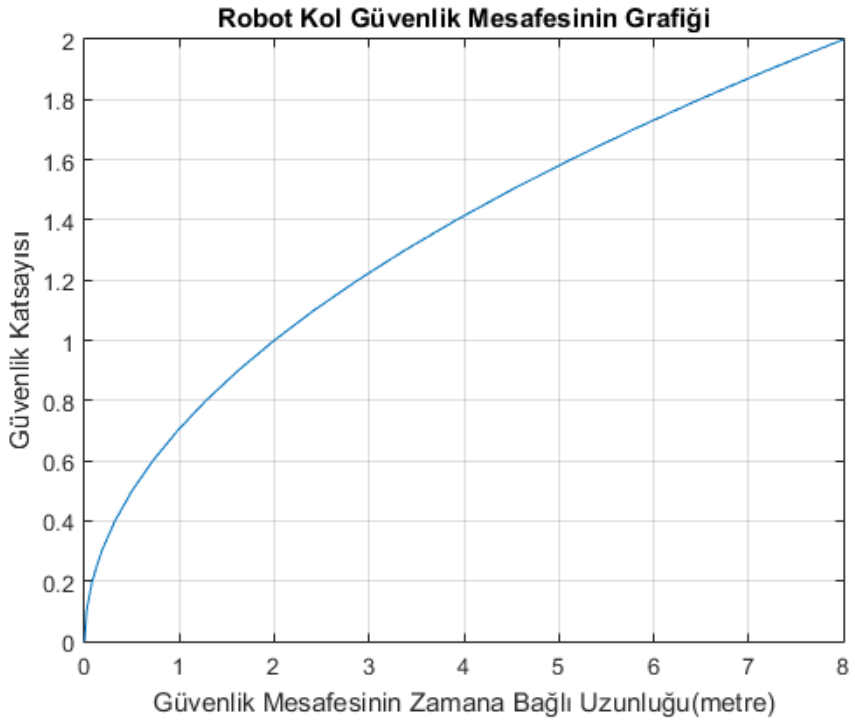
	Ticari Sistem	**Deneysel Sistem**
Manipülatör Sayısı	Değişken	1
Manipülatör Erişimi (mm)	2800	270
Çalışma Uzayı Yarıçapı (mm)	3000	300
Maksimum Manipülatör Hızı (derece/sn)	260	500
Alarm Bölgesi Yarıçapı (mm)	3500	300
Uyarı Bölgesi Yarıçapı (mm)	4000	350
Kamera Sayısı	1	2
Kamera Çözünürlüğü	Açıklanmamıştır	1280X960 1920X1080
Kamera Açısı (derece)	80	50 and 70
Grafik Kartları	PSEN se AU AM3	Nvidia Geforce GTX 1050 GTX 850M
Giriş Yapan Nesne Yarıçapı (mm)	İnsan vücudu	5
Gereken Minimum Aydınlanma Seviyesi (lux)	300	120
Gereken Minimum Saniyedeki Kare Sayısı	Açıklanmamıştır	23 and 28

Ticari bir örneğin, doktora çalışmasında tasarlanmış olan iki boyutlu ve üç boyutlu güvenlik sistemleriyle olan maliyet karşılaştırması Tablo 7.2'de verilmiştir. Maliyetler karşılaştırıldığında önerilmiş olan güvenlik sistemin fiyat performans oranının ticari ürüne kıyasla yüksek olduğu görülmektedir. Buna rağmen doktora çalışmasında tasarımı yapılmış olan sistemin maliyetleri ticari olarak hesaplanmamıştır. Dolayısıyla ticari bir örnekle karşılaştırılması halinde işletme maliyetleri, kâr, mühendislik maliyeti vs. nicelikler hariç tutulmuştur.

Tablo 7.2 Ticari bir örneğin ve tasarlanan güvenlik sistemlerinin maliyet karşılaştırması (2022 yılı için yaklaşık fiyatlar)

	Ticari Sistem (USD)	**İki Kameralı Tasarlanan Güvenlik Sistemi (USD)**	**Üç Kameralı Tasarlanan Güvenlik Sistemi (USD)**
3D Kamera	9500	-	-
2D Kamera	-	120	180
Analiz Ünitesi	14,000	360	540
Kontrol Sistemi	12,000	1700	2550
Toplam	35,500	2180	3270

Ek olarak yapılmış olan literatür araştırmalarında robot kol çalışma uzaylarına iş ve işçi güvenliğinin sağlanması amacıyla standartlarda tanımlanmış olan güvenlik mesafesi tezin önceki bölümlerinde detaylı olarak verilmiştir, fakat, tasarlanan sistemin gerçeklenmesinde bu emniyet mesafesinin küçük boyutlu robot kollar için çok fazla olduğu gözlemlenmiştir. Şekil 7.2'de standartlarda tanımlanmış olan ve Denklem 2.1'de verilmiş olan güvenlik mesafesi grafiği görülmektedir. Denkleme göre güvenlik mesafesi reaksiyon sürelerine ve emniyet katsayısına bağlı durumdadır. Şekildeki grafikte görüldüğü üzere düşey eksende reaksiyon süresinin 1 saniye seçilmesi durumunda, yatay düzlemde görülebileceği gibi en az emniyet mesafesi 2 metre olmaktadır. Bu doktora tez çalışmasının deneysel düzeneğinde kullanılan robot kol 22cm çapında bir çalışma uzayına sahiptir. Dolayısıyla bu çalışma uzayı için 2 metrelik bir güvenlik mesafesinin çok fazla olduğu görülmektedir. Bu süreye bağlı olan denklemin robot kol boyutlarıyla da ilişkilendirilerek optimize edilip standartların güncellenmesi gerektiği ön görülmektedir.



Şekil 7.2 Robot kol güvenlik mesafesinin grafiği

8
SONUÇ VE ÖNERİLER

Bu doktora tezi, çitsiz robot hücreleri için iki boyutlu kameralarla optimize edilmiş bir güvenlik sistemi önermektedir. Kamera lens açısı, kamera oryantasyon açısı ve kamera yüksekliği optimizasyon probleminin değişkenleri olarak kabul edilmiştir. Kör noktaları ortadan kaldırmak için, yaygın olarak kullanılan stereo görüntüleme yöntemlerine alternatif olarak yeni ve daha basit iki boyutlu görüntü birleştirme tekniği önerilmiştir. Optimal ve optimal olmayan seviyelerdeki parametrelere göre tasarımlar, bir derin öğrenme algoritması olan ResNet-152 kullanılarak karşılaştırılmıştır.

Doğrulama deneyleri iki kamera çözünürlüğü ve iki grafik işlemci kullanılarak sekiz aydınlatma seviyesinde yapılmıştır. Bu testlerde grafik işlemci performansı sistem tepki süresini önemli ölçüde etkilemezken kamera çözünürlüğünün oldukça etkili olduğu görülmüştür. Dolayısıyla bir güvenlik sistemi için en uygun iki boyutlu kamera çözünürlüğü en az 1080p ve görüntü yenileme hızı 23FPS olarak bulunmuştur. Kameralar en uygun şekilde yerleştirildiğinde, sistemin en düşük 120lux aydınlatmada etkin bir şekilde çalıştığı kanıtlanırken, ticari sistemlerin 400 lux altında çalıştırılamadığı bilinmektedir.

Önerilen güvenlik sistemini doğrulamak için, endüstriyel ölçüde bir robot kol ihtiva eden iş hücresinde güvenlik sistemi yeniden ölçeklendirilmiştir. Böylece önerilen tasarımın daha büyük ölçekte de uygulanabilir olduğu kanıtlanmıştır. Deneysel güvenlik sistemi, Windows 10 işletim sistemi yüklü bir PC'de çalıştırılmıştır, ancak iki dijital giriş kanalı aracılığıyla ticari bir robot kontrolörüne uyarlanabilir.

Sonuç olarak, ticari alternatiflere kıyasla kör noktalar açısından daha etkili ve fiyat-performans oranı daha iyi olan, nispeten basit iki boyutlu kameralarla kurulmuş bir güvenlik sistemi önerilmiştir.

Doktora çalışmasnın bir sonraki aşamasında, 10-farklı sınıfta 5.500 görüntüden oluşan bir veri seti üzerine altı alternatif yapay zeka modeli kullanılarak kurgulanan

deneysel bir robotik güvenlik sisteminin yabancı cisim algılama kabiliyeti geliştirilmiştir. ResNet-152, AlexNet, SqueezeNet, SVM, DT ve RF modellerini eğitmek için robotik hücre çalışma bölgelerine dışarıdan giren on farklı yabancı nesneden oluşan bir görüntü veri seti kullanılmıştır. Sonuç olarak, modellerin doğruluğu %91 ile %99 arasında değişmektedir. Derin öğrenme tabanlı SqueezeNet modelinin %99 ile en yüksek doğruluğa sahipken, makine öğrenmesi tabanlı RF ve DT modelleri %98 ile çok yakın doğruluk seviyeleri yakalamıştır. Tahmin süresi sıralamasında derin öğrenme tabanlı modelleri arasında en düşük süre 3,91 saniyedir. Makine öğrenmesi tabanlı modeller arasında DT, 0.015sn ile en düşük tahmin süresine sahip modeldir. Bu nedenle, makine öğrenmesi tabanlı bir modelin, özellikle DT ve RF modellerinin, çitsiz bir robot hücre güvenlik sistemi için en uygun alternatifler olduğu söylenebilir. Karmaşıklık matrislerinin değerlendirmeleri, modellerin hiçbirinde aşırı uyum olmadığını kanıtlamıştır. RF ve DT tarafından elde edilen yüksek doğruluk sınıflandırması ile makine öğrenme modeli, çalışma alanına girebilecek yabancı nesneler açısından robot hücresinin güvenliğini iyileştirmiştir.

Geleneksel görüntü işleme yöntemlerine alternatif olarak sunulan bu çalışmada, endüstriyel üretim ortamlarının değişkenlik ve bozucu etkilerinden kaynaklanan problemler ele alınmaktadır. Bunlar arasında titreşim, aydınlanma değişiklikleri, toz, kirlilik gibi bozucu etkenler öğrenme algoritmalarına dahil edilmiştir. Sonuç olarak, güvenlik sisteminin doğruluğunu ve kararlılığını olumsuz yönde azaltan bozucu etkiler ve görüntü öznitelikleri, yapay zekanın istatistiksel sınıflandırma yaklaşımı tarafından giderilmektedir. Ayrıca, değişikliklere uyum süresi de önemli ölçüde azalmıştır. Bu sayede istenilen ve yabancı cisimler belirlenebilir ve çitsiz bir robot çalışma alanına erişim daa yüksek bir güve ile kontrol edilebilir. Bu nedenle, sensör ve geleneksel görüntü işleme tabanlı sistemlere kıyasla daha güvenilir ve esnek bir güvenlik sistemi sunulmaktadır.

Doktora tezi, bilimsel katkı açısından değerlendirildiğinde, robot kullanılan iş hücrelerinde görüntü işleme teknolojisine dayalı bir güvenlik sistemi tasarımında karşılaşılabilecek zorluklar incelenmiştir. Mevcut yüksek maliyetli güvenlik sistemlerine alternatif birden fazla düşük maliyetli 2-D kamera kullanılan bir sistem geliştirilmiştir. Tasarımı yapılan güvenlik sisteminin değişkenleri belirlenerek

matematiksel model oluşturulmuş ve en uygun çözüm temel optimizasyon metoduyla bulunmuştur. Bu aşamada her ne kadar yapılmış olan çalışmalar kameraya dayalı güvenlik sistemi kullanan iş hücrelerinde var olabilecek değişkenler belirlenmiş olsa dahi endüstriyel üretim alanı her geçen gün farklı ürünlere ve üretim metotlarına uyum sağlayacak şekilde değişmekte ve gelişmektedir. Bundan dolayı önerilen modeldeki değişkenler yetersiz kalabilir. Bu doktora tezinde, değişkenlerin sayısı artsa da buna uygun yeni çözüm getirilmesini sağlayabilecek bir yöntem ortaya konulmuştur. Geleneksel görüntü işleme yöntemlerine alternatif özgün bir 2-boyutlu görüntü birleştirme algoritması geliştirilmiştir. Optimum sonuçlar yapay zekâ modelleri kullanılarak test edilmiş ve doğrulanmıştır. Geleneksel görüntü işleme metotlarına alternatif olarak yapay zeka modelleri ile nesne sınıflandırma yapılmış farklı modeller karşılaştırılarak hücreye giren yabancı cisim tespitinde en hızlı ve doğru model önerilmiştir.

Gelecekte tezde önerilen matematiksel model, kamera çözünürlüğü ve aydınlanma şiddeti gibi değişkenlerle genişletilerek optimizasyon probleminin maliyet fonksiyonu genişletilebilir. Tasarım yeni kısıtlar çerçevesinde geliştirilebilir.

Yapay zekâ modelleriyle nesne tanıma yapılması neticesinde dost ve yabancı nesne kavramı güvenlik sistemine eklenebilir. Emniyet uzayında izin verilen ve izin verilmeyen farklı nesnelerin olması uluslararası güvenlik standartlarında da ileriye yönelik güncelleme yapılması gerektiğini de göstermektedir.

Güvenlik sistemi gerçek zamanlı kontrolcülere entegre edilerek tepki süresinde iyileştirmeler ve sistem kararlılığının arttırılması sağlanabilir. Önerilen görüntü işlemeye dayalı güvenlik yazılımı sürekli olarak gelişen programlama dilleri, görüntü işleme kütüphaneleri ve yapay zekâ modelleri ile gelecekte daha hızlı ve daha güvenilir hale evrilebilir. Bu doktora çalışması, bu konuda tasarım yapmak isteyen herkese bir bakış açısı kazandıracaktır. Önerilen çözümlerin endüstriyel üretim alanında iş ve işçi güvenliğinin sağlanması için Türkiye ve dünyada mevcut sistemlerin dışına çıkılarak problemlere bir mühendislik bakışıyla en uygun olanı arayarak bir çözüm bulunabileceği konusunda bir ilham kaynaağı olacağı düşünülmektedir.

KAYNAKÇA

[1] Takata, S. and T. Hirano, *Human and robot allocation method for hybrid assembly systems.* CIRP Annals, 2011. 60(1): p. 9-12.

[2] Koren, Y., et al., *Reconfigurable manufacturing systems.* CIRP annals, 1999. 48(2): p. 527-540.

[3] Krüger, J., T.K. Lien, and A. Verl, *Cooperation of human and machines in assembly lines.* CIRP Annals, 2009. 58(2): p. 628-646.

[4] Krüger, J., et al., *Image based 3D surveillance for flexible man-robot-cooperation.* CIRP Annals-Manufacturing Technology, 2005. 54(1): p. 19-22.

[5] Weitschat, R., et al. *Safe and efficient human-robot collaboration part I: Estimation of human arm motions*. in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018. IEEE.

[6] Gecks, T. and D. Henrich. *Human-robot cooperation: safe pick-and-place operations*. in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*. 2005. IEEE.

[7] Bley, H., et al., *Appropriate human involvement in assembly and disassembly.* CIRP Annals-Manufacturing Technology, 2004. 53(2): p. 487-509.

[8] Iso, I., *ISO 10218-1: 2011: Robots and robotic devices–Safety requirements for industrial robots–Part 1: Robots.* Geneva, Switzerland: International Organization for Standardization, 2011.

[9] Oberer, S. and R.D. Schraft. *Robot-dummy crash tests for robot safety assessment*. in *Robotics and Automation, 2007 IEEE International Conference on*. 2007. IEEE.

[10] Kaplan, A. and M. Haenlein, *Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence.* Business Horizons, 2019. 62(1): p. 15-25.

[11] Arıkan, A., et al. *Control method simulation and application for autonomous vehicles*. in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 2018. IEEE.

[12] Bernhardt, R., et al., *Flexible assembly systems through workplace-sharing and time-sharing human-machine cooperation–pisa.* IFAC Proceedings Volumes, 2007. 40(3): p. 247-251.

[13] Zaeh, M.F. and M. Prasch, *Systematic workplace and assembly redesign for aging workforces.* Production Engineering, 2007. 1(1): p. 57-64.

[14] Durgun, I., et al., *Human-robot interaction in hybrid assembly systems using medium payload robots*. 2016.

[15] Flanagin, C., *A survey on robotics system and performance analysis*. 2016.

[16] Ogorodnikova, O. *Methodology of safety for a human robot interaction designing stage*. in *Human System Interactions, 2008 Conference on*. 2008. IEEE.

[17] Haddadin, S., A. Albu-Schäffer, and G. Hirzinger, *Requirements for safe robots: Measurements, analysis and new insights.* The International Journal of Robotics Research, 2009. 28(11-12): p. 1507-1527.

[18] Matthias, B., et al. *Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept*. in *Assembly and Manufacturing (ISAM), 2011 IEEE International Symposium on*. 2011. IEEE.

[19] Vasic, M. and A. Billard. *Safety issues in human-robot interactions*. in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013. IEEE.

[20] Michalos, G., et al., *ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future.* Procedia CIRP, 2014. 23: p. 71-76.

[21] LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. 86(11): p. 2278-2324.

[22] Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International journal of computer vision, 2015. 115(3): p. 211-252.

[23] He, K., et al. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*. in *Proceedings of the IEEE international conference on computer vision*. 2015.

[24] Chen, L.-C., et al., *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.* IEEE transactions on pattern analysis and machine intelligence, 2017. 40(4): p. 834-848.

[25] Dai, J., K. He, and J. Sun. *Instance-aware semantic segmentation via multi-task network cascades*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[26] Ullah, A., et al., *Comparative Analysis of AlexNet, ResNet18 and SqueezeNet with Diverse Modification and Arduous Implementation.* Arabian Journal for Science and Engineering, 2021: p. 1-21.

[27] Khan, A.S., et al., *A Spectrogram image-based Network Anomaly Detection System using Deep Convolutional Neural Network.* IEEE Access, 2021.

[28] Anmala, J. and V. Turuganti, *Comparison of the Performance of Decision Tree (DT) Algorithms and ELM model in the Prediction of Water Quality of the Upper Green River watershed.* Water Environment Research: a Research Publication of the Water Environment Federation, 2021.

[29] Losing, V., B. Hammer, and H. Wersing. *Interactive online learning for obstacle classification on a mobile robot*. in *2015 international joint conference on neural networks (ijcnn)*. 2015. IEEE.

[30] Hernández, A.C., et al. *Object classification in natural environments for mobile robot navigation*. in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2016. IEEE.

[31] Himmelsbach, M., T. Luettel, and H.-J. Wuensche. *Real-time object classification in 3D point clouds using point feature histograms*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009. IEEE.

[32] Spiers, A.J., et al., *Single-grasp object classification and feature extraction with simple robot hands and tactile sensors.* IEEE transactions on haptics, 2016. 9(2): p. 207-220.

[33] Ayub, A. and A.R. Wagner. *Tell me what this is: few-shot incremental object learning by a robot*. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020. IEEE.

[34] Jiang, Y., et al. *Robotic Grasp Detection Using Light-weight CNN Model*. in *2020 Chinese Control And Decision Conference (CCDC)*. 2020. IEEE.

[35] Abbas, M., et al. *AlexNet based Real-Time Detection and Segregation of Household Objects using Scorbot*. in *2020 4th International Conference on Computational Intelligence and Networks (CINE)*. 2020. IEEE.

[36] Lien, T. and F. Rasch, *Hybrid automatic-manual assembly systems.* CIRP Annals-Manufacturing Technology, 2001. 50(1): p. 21-24.

[37] Pichler, A., et al., *Towards shared autonomy for robotic tasks in manufacturing.* Procedia Manufacturing, 2017. 11: p. 72-82.

[38] ISO, T., *199: ISO 13855: 2010 Safety of machinery–Positioning of safeguards with respect to the approach speeds of parts of the human body.* International Organization for Standardization (ed.), 2010.

[39] Rajpoot, H.C., *HCR's Infinite-series (Analysis of oblique frustum of a right circular cone).* International Journal of Mathematics & Physical Sciences Research, 2014. 2(2): p. 17.

[40] Rao, S.S., *Engineering optimization: theory and practice*. 2009: John Wiley & Sons.

[41] Arora, J.S., *Introduction to optimum design*. 2004: Elsevier.

[42] Tan, J.T.C., et al. *Safety design and development of human-robot collaboration in cellular manufacturing*. in *2009 IEEE International Conference on Automation Science and Engineering*. 2009. IEEE.

[43] Himmelsbach, U.B., T.M. Wendt, and M. Lai. *Towards safe speed and separation monitoring in human-robot collaboration with 3d-time-of-flight cameras*. in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. 2018. IEEE.

[44] Michalos, G., et al., *Design considerations for safe human-robot collaborative workplaces.* Procedia CIrP, 2015. 37: p. 248-253.

[45] Schmidt, B. and L. Wang, *Depth camera based collision avoidance via active robot control.* Journal of manufacturing systems, 2014. 33(4): p. 711-718.

[46] ASSOCIATION, D.C.C.J.C.I., *CIPA DCG-001-2005, Guideline for Noting Digital Camera Specifications in Catalogs.* 2005.

[47] Camera and I.P. Association, *CIPA DC-003-2003, Resolution measurement methods for digital cameras.* Standard of the Camera & Imaging Products Association CIPA DC-003-Translation-2003, Japan, 2003.

[48] He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[49] Bang, S., et al., *Encoder–decoder network for pixel-level road crack detection in black-box images.* Computer-Aided Civil and Infrastructure Engineering, 2019. 34(8): p. 713-727.

[50] Wang, H. and Y. Xia, *Chestnet: A deep neural network for classification of thoracic diseases on chest radiography.* arXiv preprint arXiv:1807.03058, 2018.

[51] Son, H., et al., *Detection of construction workers under varying poses and changing background in image sequences via very deep residual networks.* Automation in Construction, 2019. 99: p. 27-38.

[52] Wu, Z., C. Shen, and A. Van Den Hengel, *Wider or deeper: Revisiting the resnet model for visual recognition.* Pattern Recognition, 2019. 90: p. 119-133.

[53] Ioffe, S. and C. Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. in *International conference on machine learning*. 2015. PMLR.

[54] Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *Icml*. 2010.

[55] Farooq, A., et al. *A deep CNN based multi-class classification of Alzheimer's disease using MRI*. in *2017 IEEE International Conference on Imaging systems and techniques (IST)*. 2017. IEEE.

[56] Korfiatis, P., et al., *Residual deep convolutional neural network predicts MGMT methylation status.* Journal of digital imaging, 2017. 30(5): p. 622-628.

[57] Yusuf, M., et al. *Webcam Based Lux Meter Using Grid-Based Image Processing*. in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*. 2018. IEEE.

[58] Ismail, A., et al. *Development of a webcam based lux meter*. in *2013 IEEE Symposium on Computers & Informatics (ISCI)*. 2013. IEEE.

[59] Schlyter, P., *Radiometry and photometry in astronomy.* Available: stjarnhimlen. se/comp/radfaq. html, 2009. 1.

[60] Pears, A., *Chapter 7: Appliance technologies and scope for emission reduction.* Strategic study of household energy and greenhouse issues, 1998: p. 61.

A
EKLER

Görüntü İşlemeye Dayalı Güvenlik Yazılımı

```
import os
import time
import serial
import cv2
import numpy as np
from PIL import Image, ImageDraw, ImageFilter

ustcam = cv2.VideoCapture(0, cv2.CAP_DSHOW)
altcam = cv2.VideoCapture(1, cv2.CAP_DSHOW)

ser=serial.Serial(port="COM6",baudrate=9600)

hassasiyet = 140

resize_value = (640, 480)
merkez_point = (int(resize_value[0]/2)+18, int(resize_value[1]/2)+5)
merkez_kucuk = (100, 100)
merkez_buyuk = (150, 130)
merkez_buyuk2 = (160, 140)
mavi = (0, 0, 255)
yesil = (0, 255, 0)
kirmizi = (255, 0, 0)
beyaz = (255, 255, 255)
siyah = (0, 0, 0)
onay = 0

yaricap1 = 130
yaricap2 = 150
bosluk1 = 30
bosluk2 = 40

elips_acisi1 = ((int(merkez_point[0]-yaricap1-bosluk1)+22),
        (int(merkez_point[1]-yaricap1-bosluk1)+30),
        (int(merkez_point[0]+yaricap1+bosluk1)-22),
        (int(merkez_point[1]+yaricap1+bosluk1)-22))

elips_acisi2 = ((int(merkez_point[0]-yaricap2-bosluk2)-5),
        (int(merkez_point[1]-yaricap2-bosluk2)+26),
        (int(merkez_point[0]+yaricap2+bosluk2)+0),
        (int(merkez_point[1]+yaricap2+bosluk2)+0))

elips_acisi3 = ((int(merkez_point[0]-yaricap2-bosluk2)-5),
```

```
            (int(merkez_point[1]-yaricap2-bosluk2)+26),
            (int(merkez_point[0]+yaricap2+bosluk2)+0),
            (int(merkez_point[1]+yaricap2+bosluk2)-25))

def frame_al():
    _, frame = ustcam.read()
    _, frame2 = altcam.read()
    frame = cv2.flip(frame, 1)
    frame2 = cv2.flip(frame2, -1)
    #frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #frame2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
    cam11 = cv2.resize(frame, resize_value, interpolation=cv2.INTER_AREA)
    cam2 = cv2.resize(frame2, resize_value, interpolation=cv2.INTER_AREA)
    cam12 = cam11.copy()
    cam21 = cam2.copy()
    return cam11, cam12, cam2, cam21

def ic_cember_ciz(img, cntr, ellipaci, aci, aci1, aci2, color, thickness):
    image = cv2.ellipse(img, cntr, ellipaci,
                        aci, aci1, aci2, color, thickness)
    return image

def dis_cember_ciz(img, acideger, colora):
    img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    background = Image.new(img.mode, img.size, colora)
    mask = Image.new("L", img.size, 0)
    draw = ImageDraw.Draw(mask)
    draw.ellipse(acideger, fill=255)
    img = Image.composite(img, background, mask)
    img = cv2.cvtColor(np.array(img), cv2.COLOR_RGB2BGR)
    return img

def time_convert(name, sec):
    mins = sec // 60
    sec = sec % 60
    hours = mins // 60
    mins = mins % 60
    print(name+"{0}:{1}:{2}".format(int(hours), int(mins), sec))

def diffImg(t0, t1, t2, start_date):
    d1 = cv2.absdiff(t2, t1)
    d2 = cv2.absdiff(t1, t0)
    btw = cv2.bitwise_and(d1, d2)
    if btw.max() > np.uint8(hassasiyet):
        ser.write(b'5')
        end_time1 = time.time()
        time_lapsed1 = end_time1 - start_date
        time_convert("ic deger ", time_lapsed1)
    else:
        start_date = time.time()
    return btw
```

```
def diffImg2(t0, t1, t2, start_date):
   d1 = cv2.absdiff(t2, t1)
   d2 = cv2.absdiff(t1, t0)
   btw = cv2.bitwise_and(d1, d2)
   if btw.max() > np.uint8(hassasiyet):
      ser.write(b'4')
      end_time1 = time.time()
      time_lapsed1 = end_time1 - start_date
      time_convert("dis deger ", time_lapsed1)
   else:
      start_date = time.time()
   return btw

start_date = time.time()
while(1):

   for sayac in range(0, 3, 1):
      if sayac == 0:
         cam1ic, cam1dis, cam2ic, cam2dis = frame_al()
         cam1ic = ic_cember_ciz(cam1ic, merkez_point,merkez_kucuk, 0, 0, 360, mavi, -1)
         cam2ic = ic_cember_ciz(cam2ic, merkez_point,merkez_kucuk, 0, 0, 360, yesil, -1)
         cam1ic = dis_cember_ciz(cam1ic, elips_acisi1, mavi)
         cam2ic = dis_cember_ciz(cam2ic, elips_acisi1, yesil)
         cam1dis = ic_cember_ciz(cam1dis, merkez_point, merkez_buyuk, 0, 0, 360, siyah, -1)
         cam2dis = ic_cember_ciz(cam2dis, merkez_point, merkez_buyuk2, 0, 0, 360, beyaz, -
1)
         cam1dis = dis_cember_ciz(cam1dis, elips_acisi2, siyah)
         cam2dis = dis_cember_ciz(cam2dis, elips_acisi3, beyaz)
         cam1ic[0:240] = 0
         cam2ic[240:480] = 0
         icalanlar = cam1ic+cam2ic
         cam1dis[0:240] = 0
         cam2dis[240:480] = 0
         disalanlar = cam1dis+cam2dis
         tic1 = icalanlar
         tdis1 = disalanlar
      elif sayac == 1:
         cam1ic, cam1dis, cam2ic, cam2dis = frame_al()
         cam1ic = ic_cember_ciz(cam1ic, merkez_point,merkez_kucuk, 0, 0, 360, mavi, -1)
         cam2ic = ic_cember_ciz(cam2ic, merkez_point,merkez_kucuk, 0, 0, 360, yesil, -1)
         cam1ic = dis_cember_ciz(cam1ic, elips_acisi1, mavi)
         cam2ic = dis_cember_ciz(cam2ic, elips_acisi1, yesil)
         cam1dis = ic_cember_ciz(cam1dis, merkez_point, merkez_buyuk, 0, 0, 360, siyah, -1)
         cam2dis = ic_cember_ciz(cam2dis, merkez_point, merkez_buyuk2, 0, 0, 360, beyaz, -
1)
         cam1dis = dis_cember_ciz(cam1dis, elips_acisi2, siyah)
         cam2dis = dis_cember_ciz(cam2dis, elips_acisi3, beyaz)
         cam1ic[0:240] = 0
         cam2ic[240:480] = 0
         icalanlar = cam1ic+cam2ic
         cam1dis[0:240] = 0
         cam2dis[240:480] = 0
```

```
            disalanlar = cam1dis+cam2dis
            tic2 = icalanlar
            tdis2 = disalanlar
        elif sayac == 2:
            cam1ic, cam1dis, cam2ic, cam2dis = frame_al()
            cam1ic = ic_cember_ciz(cam1ic, merkez_point,merkez_kucuk, 0, 0, 360, mavi, -1)
            cam2ic = ic_cember_ciz(cam2ic, merkez_point,merkez_kucuk, 0, 0, 360, yesil, -1)
            cam1ic = dis_cember_ciz(cam1ic, elips_acisi1, mavi)
            cam2ic = dis_cember_ciz(cam2ic, elips_acisi1, yesil)
            cam1dis = ic_cember_ciz(cam1dis, merkez_point, merkez_buyuk, 0, 0, 360, siyah, -1)
            cam2dis = ic_cember_ciz(cam2dis, merkez_point, merkez_buyuk2, 0, 0, 360, beyaz, -
1)
            cam1dis = dis_cember_ciz(cam1dis, elips_acisi2, siyah)
            cam2dis = dis_cember_ciz(cam2dis, elips_acisi3, beyaz)
            cam1ic[0:240] = 0
            cam2ic[240:480] = 0
            icalanlar = cam1ic+cam2ic
            cam1dis[240:480] = 0
            cam2dis[0:240] = 0
            disalanlar = cam1dis+cam2dis
            tic3 = icalanlar
            tdis3 = disalanlar

    cv2.imshow("ic Movement Indicator", diffImg(tic1, tic2, tic3, start_date))
    cv2.imshow("dis Movement Indicator", diffImg2(tdis1, tdis2, tdis3, start_date))
    cv2.imshow("Ic Alanlar", icalanlar)
    cv2.imshow("Dis Alanlar", disalanlar)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    elif cv2.waitKey(1) & 0xFF == ord('r'):
        ser.write(b'6')

ustcam.release()
altcam.release()
cv2.destroyAllWindows()
```

En Uygunu Bulma Kullanıcı Programı

```
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
import math

class calc:
   def first_kisit(self):
      self.kameralar_acisi_radian_result.delete(0,END)
      self.kameralar_arasi_mesafe_result.delete(0,END)
      self.max_kamera_h_result.delete(0,END)
      self.results2.delete(0,END)
      self.kamera_alan_result.delete(0,END)
      self.piksel_boyutu_result.delete(0,END)
      self.taranacak_piksel_result.delete(0,END)
      if self.var.get():
         x1=float( self.e1.get())/2.0
         a=int(self.e4.get())/2.0
         k=int(self.e5.get())
         k1=self.e7.get()
         ob=int(self.e6.get())
         if self.var2.get():
            ek=int(self.e3.get())
            if k1 == "1MP/720P-1280X720":
               k2=921600
               k3=720
            elif k1 == "QVGA/960P-1280X960":
               k2=1228800
               k3=960
            elif k1 == "1,3MP-1280X1024":
               k2=1310720
               k3=1024
            elif k1 == "2MP/1080P-1920X1080":
               k2=2073600
               k3=1080
            elif k1 == "2.3MP-1920X1200":
               k2=2304000
               k3=1200
            elif k1 == "3MP-2048X1536":
               k2=3145728
               k3=1536
            elif k1 == "5MP-2560X1920":
               k2=4915200
               k3=1920
            elif k1 == "6MP-3032X2008":
               k2=6088256
               k3=2008
            elif k1 == "8MP-3264X2448":
               k2=7990272
               k3=2448
            a1=a+ek
```

```
        y=[]
        b=[]
        x=math.radians(x1)
        min_a= a* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
        max_a=a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
        k4=max_a/k3
        k5=round(ob/k4)
        k6=math.pow(k5,2)

        y_altsinir=(a/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.ta
n(x/2)),3)-(1.0/(math.tan(x/2))))-
(a*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
        y_ustsinir=(a1/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a1/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.t
an(x/2)),3)-(1.0/(math.tan(x/2))))-
(a1*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))

alan_mm=(math.pi*(math.pow(min_a,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow((mat
h.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
        alan_m=alan_mm/1000000

alan_mm2=(math.pi*(math.pow(y_altsinir,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow(
(math.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
        alan_m2=alan_mm/1000000

        self.kameralar_acisi_radian_result.insert(0,str(round(x,4))+' radyan'+
str(round(x1))+' derece')
        self.kameralar_arasi_mesafe_result.insert(0,str(round(a1*2))+' mm')
        self.max_kamera_h_result.insert(0,str(round(max_a,2))+' mm')
        self.kamera_alan_result.insert(0,str(round(alan_m,2))+' m^2')
        self.piksel_boyutu_result.insert(0,str(round(k4,2))+' mm')
        self.taranacak_piksel_result.insert(0,str(round(k5,2)))
        if a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6) <k:
          self.results2.insert(0,' 2 kamera için:')
        else:
          if y_ustsinir<k:
            self.results2.insert(0,'120 derece açı farkı ile kameralar arası karşılıklı uzaklık
'+str(a)+' olacak şekilde en az 3 kamera gerekmektedir.')
            self.max_kamera_h_result.delete(0,END)
            self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
            self.kamera_alan_result.delete(0,END)
            self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
          else:
            self.results2.insert(0,'Verilen yükseklik sınırı için kamera yerleşimi
yapılamamaktadır.')
            self.max_kamera_h_result.delete(0,END)
```

```
                self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                self.kamera_alan_result.delete(0,END)
                self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
        else:
          if k1 == "1MP/720P-1280X720":
            k2=921600
            k3=720
          elif k1 == "QVGA/960P-1280X960":
            k2=1228800
            k3=960
          elif k1 == "1,3MP-1280X1024":
            k2=1310720
            k3=1024
          elif k1 == "2MP/1080P-1920X1080":
            k2=2073600
            k3=1080
          elif k1 == "2.3MP-1920X1200":
            k2=2304000
            k3=1200
          elif k1 == "3MP-2048X1536":
            k2=3145728
            k3=1536
          elif k1 == "5MP-2560X1920":
            k2=4915200
            k3=1920
          elif k1 == "6MP-3032X2008":
            k2=6088256
            k3=2008
          elif k1 == "8MP-3264X2448":
            k2=7990272
            k3=2448
          a1=a+3000
          y=[]
          b=[]
          x=math.radians(x1)
          min_a= a* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
          max_a=a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
          k4=max_a/k3
          k5=round(ob/k4)
          k6=math.pow(k5,2)
          y_altsinir=(a/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.ta
n(x/2)),3)-(1.0/(math.tan(x/2))))-
(a*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
          y_ustsinir=(a1/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a1/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.t
an(x/2)),3)-(1.0/(math.tan(x/2))))-
(a1*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
```

```
alan_mm=(math.pi*(math.pow(min_a,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow((mat
h.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m=alan_mm/1000000
alan_mm2=(math.pi*(math.pow(y_altsinir,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow(
(math.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m2=alan_mm/1000000
            self.kameralar_acisi_radian_result.insert(0,str(round(x,4))+' radyan'+
str(round(x1))+' derece')
            self.kameralar_arasi_mesafe_result.insert(0,str(round(a1*2))+' mm')
            self.max_kamera_h_result.insert(0,str(round(max_a,2))+' mm')
            self.kamera_alan_result.insert(0,str(round(alan_m,2))+' m^2')
            self.piksel_boyutu_result.insert(0,str(round(k4,2))+' mm')
            self.taranacak_piksel_result.insert(0,str(round(k5,2)))

            if a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6) <k:
              self.results2.insert(0,' 2 kamera için:')
            else:
              if y_ustsinir<k:
                self.results2.insert(0,'120 derece açı farkı ile kameralar arası karşılıklı uzaklık
'+str(a)+' olacak şekilde en az 3 kamera gerekmektedir.')
                self.max_kamera_h_result.delete(0,END)
                self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                self.kamera_alan_result.delete(0,END)
                self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
              else:
                self.results2.insert(0,'Verilen yükseklik sınırı için kamera yerleşimi
yapılamamaktadır.')
                self.max_kamera_h_result.delete(0,END)
                self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                self.kamera_alan_result.delete(0,END)
                self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
      else:
        x1=29
        a=int(self.e4.get())/2
        k=int(self.e5.get())
        k1=self.e7.get()
        ob=int(self.e6.get())
        if self.var2.get():
          ek=int(self.e3.get())
          if k1 == "1MP/720P-1280X720":
            k2=921600
            k3=720
          elif k1 == "QVGA/960P-1280X960":
            k2=1228800
            k3=960
          elif k1 == "1,3MP-1280X1024":
            k2=1310720
            k3=1024
          elif k1 == "2MP/1080P-1920X1080":
            k2=2073600
            k3=1080
```

```
            elif k1 == "2.3MP-1920X1200":
               k2=2304000
               k3=1200
            elif k1 == "3MP-2048X1536":
               k2=3145728
               k3=1536
            elif k1 == "5MP-2560X1920":
               k2=4915200
               k3=1920
            elif k1 == "6MP-3032X2008":
               k2=6088256
               k3=2008
            elif k1 == "8MP-3264X2448":
               k2=7990272
               k3=2448
            a1=a+ek
            y=[]
            b=[]
            x=math.radians(x1)
            min_a= a* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
            max_a=a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
            k4=max_a/k3
            k5=round(ob/k4)
            k6=math.pow(k5,2)
            y_altsinir=(a/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.ta
n(x/2)),3)-(1.0/(math.tan(x/2))))-
(a*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
            y_ustsinir=(a1/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a1/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.t
an(x/2)),3)-(1.0/(math.tan(x/2))))-
(a1*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
alan_mm=(math.pi*(math.pow(min_a,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow((mat
h.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m=alan_mm/1000000
alan_mm2=(math.pi*(math.pow(y_altsinir,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow(
(math.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m2=alan_mm/1000000
            self.kameralar_acisi_radian_result.insert(0,str(round(x,4))+' radyan    -    ' +
str(round(x1))+' derece')
            self.kameralar_arasi_mesafe_result.insert(0,str(round(a1*2))+' mm')
            self.max_kamera_h_result.insert(0,str(round(max_a,2))+' mm')
            self.kamera_alan_result.insert(0,str(round(alan_m,2))+' m^2')
            self.piksel_boyutu_result.insert(0,str(round(k4,2))+' mm')
            self.taranacak_piksel_result.insert(0,str(round(k5,2)))
            if a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6) <k:
               self.results2.insert(0,' 2 kamera için:')
```

```
            else:
                if y_ustsinir<k:
                    self.results2.insert(0,'120 derece açı farkı ile kameralar arası karşılıklı uzaklık
'+str(a)+' olacak şekilde en az 3 kamera gerekmektedir.')
                    self.max_kamera_h_result.delete(0,END)
                    self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                    self.kamera_alan_result.delete(0,END)
                    self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
                else:
                    self.results2.insert(0,'Verilen yükseklik sınırı için kamera yerleşimi
yapılamamaktadır.')
                    self.max_kamera_h_result.delete(0,END)
                    self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                    self.kamera_alan_result.delete(0,END)
                    self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
        else:
            ek=3000
            if k1 == "1MP/720P-1280X720":
                k2=921600
                k3=720
            elif k1 == "QVGA/960P-1280X960":
                k2=1228800
                k3=960
            elif k1 == "1,3MP-1280X1024":
                k2=1310720
                k3=1024
            elif k1 == "2MP/1080P-1920X1080":
                k2=2073600
                k3=1080
            elif k1 == "2.3MP-1920X1200":
                k2=2304000
                k3=1200
            elif k1 == "3MP-2048X1536":
                k2=3145728
                k3=1536
            elif k1 == "5MP-2560X1920":
                k2=4915200
                k3=1920
            elif k1 == "6MP-3032X2008":
                k2=6088256
                k3=2008
            elif k1 == "8MP-3264X2448":
                k2=7990272
                k3=2448
            a1=a+ek
            y=[]
            b=[]
            x=math.radians(x1)
            min_a= a* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
            max_a=a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6)
```

```
            k4=max_a/k3
            k5=round(ob/k4)
            k6=math.pow(k5,2)
            y_altsinir=(a/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.ta
n(x/2)),3)-(1.0/(math.tan(x/2))))-
(a*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
            y_ustsinir=(a1/2)/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))+(a1/2*math.pow(1.0/(math.tan(x/2)),4))/(math.pow(1.0/(math.t
an(x/2)),3)-(1.0/(math.tan(x/2))))-
(a1*3*math.pow(1.0/(math.tan(x/2)),2))/(math.pow(1.0/(math.tan(x/2)),3)-
(1.0/(math.tan(x/2))))
alan_mm=(math.pi*(math.pow(min_a,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow((mat
h.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m=alan_mm/1000000
alan_mm2=(math.pi*(math.pow(y_altsinir,2))*math.sin(x)*math.sin(2*x))/(2*(math.pow(
(math.pow(math.cos(x),2)-(math.pow(math.sin(x),2))),3/2)))
            alan_m2=alan_mm/1000000
            self.kameralar_acisi_radian_result.insert(0,str(round(x,4))+' radyan'+
str(round(x1))+' derece')
            self.kameralar_arasi_mesafe_result.insert(0,str(round(a1*2))+' mm')
            self.max_kamera_h_result.insert(0,str(round(max_a,2))+' mm')
            self.kamera_alan_result.insert(0,str(round(alan_m,2))+' m^2')
            self.piksel_boyutu_result.insert(0,str(round(k4,2))+' mm')
            self.taranacak_piksel_result.insert(0,str(round(k5,2)))
            if a1* math.sqrt(math.pow(math.tan(x/2),2)+ (1.0/(math.tan(x/2)))*
(1.0/(math.tan(x/2)))-6) <k:
               self.results2.insert(0,' 2 kamera için:')
            else:
               if y_ustsinir<k:
                  self.results2.insert(0,'120 derece açı farkı ile kameralar arası karşılıklı uzaklık
'+str(a)+' olacak şekilde en az 3 kamera gerekmektedir.')
                  self.max_kamera_h_result.delete(0,END)
                  self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                  self.kamera_alan_result.delete(0,END)
                  self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
               else:
                  self.results2.insert(0,'Verilen yükseklik sınırı için kamera yerleşimi
yapılamamaktadır.')
                  self.max_kamera_h_result.delete(0,END)
                  self.max_kamera_h_result.insert(0,str(round(y_ustsinir))+' mm')
                  self.kamera_alan_result.delete(0,END)
                  self.kamera_alan_result.insert(0,str(round(alan_m2,2))+' m^2')
   def __init__(self,master):
      """Constructor method"""
      master.title('Merdan Özkahraman Doktora Tez Programı')
      master.geometry()
      self.var = IntVar()
      self.var2 = IntVar()
      self.l0 = Label(master,text="Bir kamera açısı girmek istiyor musunuz?")
      self.l0.grid(row=0,column=0)
```

```
        self.cbp=Checkbutton(master,text="Evet",variable=self.var)
        self.cbn=Checkbutton(master,text="Hayır")
        self.cbn.grid(row=0,column=2)
        self.cbp.grid(row=0,column=1)

        self.l2 = Label(master,text="Emniyet mesafesi girmek istiyor musunuz?\n(Eğer
Hayır'ı seçerseniz otomatik olarak hesaplanacaktır.)")
        self.l2.grid(row=2,column=0)
        self.cbp2=Checkbutton(master,text="Evet",variable=self.var2)
        self.cbn2=Checkbutton(master,text="Hayır")
        self.cbn2.grid(row=2,column=2)
        self.cbp2.grid(row=2,column=1)

        self.l1 = Label(master,text="Kamera açısını giriniz.(derece):")
        self.l3 = Label(master,text="Emniyet mesafesi giriniz.(mm):")
        self.l4 = Label(master,text="Robot çalışma uzayı çapını giriniz.(mm):")
        self.l5 = Label(master,text="İş hücresi maksimum tavan yüksekliği ne
olmalıdır?(mm):")
        self.l6 = Label(master,text="Tespit edilecek objenin en büyük çap uzunluğu
nedir?(mm)")
        self.l7 = Label(master,text="Kamera çözünürlüğü seçiniz:")

        self.l1.grid(row=1,column=0)
        self.l3.grid(row=4,column=0)
        self.l4.grid(row=5,column=0)
        self.l5.grid(row=6,column=0)
        self.l6.grid(row=7,column=0)
        self.l7.grid(row=8,column=0)

        course=["1MP/720P-1280X720","QVGA/960P-1280X960","1,3MP-
1280X1024","2MP/1080P-1920X1080","2.3MP-1920X1200","3MP-2048X1536","5MP-
2560X1920","6MP-3032X2008","8MP-3264X2448"]

        self.e1 = Entry(master,width=50)
        self.e3 = Entry(master,width=50)
        self.e4 = Entry(master,width=50)
        self.e5 = Entry(master,width=50)
        self.e6 = Entry(master,width=50)
        self.e7 = ttk.Combobox(master,values=course,width=40)

        self.e1.grid(row=1,column=1)
        self.e3.grid(row=4,column=1)
        self.e4.grid(row=5,column=1)
        self.e5.grid(row=6,column=1)
        self.e6.grid(row=7,column=1)
        self.e7.grid(row=8,column=1)
        self.e7.current(0)

        self.results=Label(master,text="Sonuçlar")
        self.results.grid(row=9,column=0)
        self.results2=Entry(master,width=90)
        self.results2.grid(row=9,column=1)
```

```
        self.kameralar_acisi_radian_label = Label(master,text="Kameranın yatay düzlem ile
arasındaki açı")
        self.kameralar_acisi_radian_result = Entry(master,width=50)
        self.kameralar_acisi_radian_label.grid(row=10,column=0)
        self.kameralar_acisi_radian_result.grid(row=10,column=1)

        self.kameralar_arasi_mesafe_label = Label(master,text="Kameralar arası mesafe")
        self.kameralar_arasi_mesafe_result = Entry(master,width=50)
        self.kameralar_arasi_mesafe_label.grid(row=11,column=0)
        self.kameralar_arasi_mesafe_result.grid(row=11,column=1)

        self.max_kamera_h_label = Label(master,text="Kameranın Yerleştirilmesi Gereken
Yükseklik(h)")
        self.max_kamera_h_result = Entry(master,width=50)
        self.max_kamera_h_label.grid(row=12,column=0)
        self.max_kamera_h_result.grid(row=12,column=1)

        self.kamera_alan_label = Label(master,text="Kameranın Kapsadığı Çalışma Uzayı
Alanı(m^2)")
        self.kamera_alan_result = Entry(master,width=50)
        self.kamera_alan_label.grid(row=13,column=0)
        self.kamera_alan_result.grid(row=13,column=1)

        self.piksel_boyutu_label = Label(master,text="Kameradaki bir piksel boyutu(mm)")
        self.piksel_boyutu_result = Entry(master,width=50)
        self.piksel_boyutu_label.grid(row=14,column=0)
        self.piksel_boyutu_result.grid(row=14,column=1)

        self.taranacak_piksel_label = Label(master,text="Hareketin tespit edilmesi için
taranacak piksel sayısı")
        self.taranacak_piksel_result = Entry(master,width=50)
        self.taranacak_piksel_label.grid(row=15,column=0)
        self.taranacak_piksel_result.grid(row=15,column=1)

        self.e1.focus_set()
        Button(master,text="En Uygunu Bul",width=11,height=3,fg="blue",
           bg="orange",command=lambda:self.first_kisit()).grid(
                        row=9, column=9,columnspan=2)
root = Tk()
obj=calc(root)
root.mainloop()
```

Görüntü İşlemeye Dayalı Olarak Çalışan Güvenlik Yazılımıyla Eşzamanlı Çalışan Motor Kontrolü için Arduino Kodları

```
#include <Servo.h>

Servo sg90;
int pos = 0;
int pozisyon = 0;
int süre = 15;

void setup() {
  sg90.attach(5);
  Serial.begin(9600);}

void loop() {
  oku();
  for (pos = 0; pos <= 180; pos += 1) {
    sg90.write(pos);
    oku();
    delay(süre);}
  for (pos = 180; pos >= 0; pos -= 1) {
    sg90.write(pos);
    oku();
    delay(süre); }}

void oku() {
  int a = Serial.read();
  if (a == '4') {
    süre = 100;}
  if (a == '5') {
    delay(5000);}
  if (a == '6') {
    süre = 15; }}
```

TEZDEN ÜRETİLMİŞ YAYINLAR

Makaleler

1. Ozkahraman, M.; Yilmaz, C.; Livatyali, H. Design and Validation of a Camera-Based Safety System for Fenceless Robotic Work Cells. Appl. Sci. 2021, 11, 11679. https://doi.org/10.3390/app112411679.

2. Ozkahraman, M.; Livatyali, H. Artificial Intelligence in Foreign Object Classification in Fenceless Robotic Work Cells Using 2-D Safety Cameras. Technical Gazette. 2022, preprint https://doi.org/10.17559/TV-20211122150850.