



**RASPBERRY PI DEVRE KARTI KULLANARAK  
NESNE BULMA VE TANIMA ALGORİTMALARININ  
BİR ROBOT KOL ÜZERİNDE UYGULANMASI**

**Çağrı KAYMAK**

**Yüksek Lisans Tezi  
Mekatronik Mühendisliği Anabilim Dalı  
Danışman: Doç. Dr. Ayşegül UÇAR  
TEMMUZ-2016**

T.C  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

RASPBERRY PI DEVRE KARTI KULLANARAK NESNE BULMA VE TANIMA  
ALGORİTMALARININ BİR ROBOT KOL ÜZERİNDE UYGULANMASI

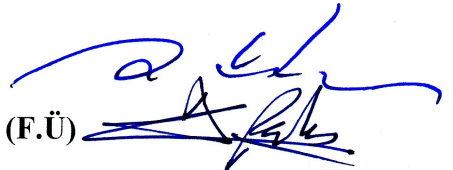

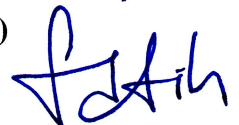
YÜKSEK LİSANS TEZİ

Çağrı KAYMAK

(131134103)

Tezin Enstitüye Verildiği Tarih : 11 Temmuz 2016

Tezin Savunulduğu Tarih : 29 Temmuz 2016

Tez Danışmanı : Doç. Dr. Ayşegül UÇAR (F.Ü)   
Diğer Jüri Üyeleri : Yrd. Doç. Dr. Mehmet POLAT (F.Ü)   
Doç. Dr. Muhammed Fatih TALU (İ.Ü) 

TEMMUZ-2016

## ÖNSÖZ

Öncelikle lisans eğitimim boyunca beni sürekli destekleyen ve çalışmama teşvik eden değerli hocam Sayın Doç. Dr. Ayşegül UÇAR'ın, yüksek lisans eğitimimde tez danışmanım olduğu için çok şanslı olduğumu belirtmekle başlamak istiyorum.

Yüksek lisans tez çalışma konumun seçiminde ve tez çalışmalarımın tamamlanmasında sağlamış olduğu katkıların yanı sıra göstermiş olduğu ilgi, hoşgörü ve sabır ile yüksek lisans eğitimim boyunca da desteğini esirgemeyen Sayın Doç. Dr. Ayşegül UÇAR hocama çok teşekkür ederim.

Ayrıca, tez çalışmalarına olan katkılarından dolayı Doç. Dr. Oğuz YAKUT hocama ve diğer bölüm hocalarıma teşekkürlerimi sunarım.

Yüksek lisans öğrencisi Rüya SARICI'ya da tez çalışmalarım süresince göstermiş olduğu manevi desteğinden dolayı çok teşekkür ederim.

Son olarak da, hayatım boyunca her daim yanımda olan, sevgi ve desteklerini hiçbir zaman esirgemeyen aileme teşekkürü borç bilir, saygılarımı sunarım.

**Çağrı KAYMAK**  
**ELAZIĞ – 2016**

# İÇİNDEKİLER

## Sayfa No

ÖNSÖZ .....	II
İÇİNDEKİLER.....	III
ÖZET .....	V
SUMMARY .....	VI
ŞEKİLLER LİSTESİ .....	VII
TABLolar LİSTESİ .....	X
KISALTMALAR.....	XI
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. Tezin Organizasyonu.....	6
<b>2. NESNE BULMA VE TANIMA.....</b>	<b>8</b>
2.1. Öznitelik Çıkarımı .....	12
2.1.1. Ölçekten Bağımsız Öznitelik Dönüşümü Algoritması .....	12
2.1.1.1. Ölçeksel Uzaydaki Tepe (Uç) Noktaların Tespiti .....	13
2.1.1.2. Anahtar Noktaların Yerinin Saptanması .....	15
2.1.1.3. Anahtar Noktalara Yönelim Atanması .....	18
2.1.1.4. Anahtar Nokta Tanımlayıcılarının Oluşturulması .....	18
2.1.2. Hızlandırılmış Gürbüz Öznitelikler Algoritması.....	19
2.1.2.1. Anahtar Noktaların Yerinin Saptanması .....	20
2.1.2.2. Anahtar Nokta Tanımlayıcılarının Oluşturulması .....	22
2.1.3. Hızlandırılmış Segment Testinden Öznitelikler Algoritması.....	24
2.1.4. İkili Gürbüz Bağımsız Temel Öznitelikler Algoritması.....	25
2.1.5. Yönlendirilmiş FAST ve Döndürülmüş BRIEF Algoritması.....	26
2.2. Öznitelik Eşleştirme ve Sınıflandırma .....	28
2.3. Homografi Tahmini .....	30
<b>3. ROBOTİK VE ROBOT MANİPÜLATÖRLER.....</b>	<b>32</b>
3.1. Robotların Tarihsel Gelişimi .....	32
3.2. Endüstriyel Robotik ve Robot Kollar .....	34
3.2.1. Robot Kol Eklem Tipleri .....	36
3.2.2. Serbestlik Derecesi .....	36
3.3. Robot Kolların Endüstrideki Kullanımları .....	37
3.4. Robot Manipülatörlerin Eklem Tiplerine Göre Sınıflandırılması .....	38
3.4.1. Kartezyen Robot.....	39
3.4.2. Silindirik Robot .....	39
3.4.3. Küresel Robot.....	40
3.4.4. SCARA Robot.....	41
3.4.5. Eklemlili (Mafsallı) Robot.....	41
<b>4. ROBOT KOL KİNEMATİK ANALİZİ .....</b>	<b>43</b>
4.1. Düz (İleri Yön) Kinematik Analiz.....	43
4.1.1. Denavit-Hartenberg (D-H) Yöntemi .....	44
4.1.2. Eksen Takımlarının Yerleştirilmesi.....	45
4.1.3. Homojen Dönüşüm Matrislerinin Bulunması .....	46
4.1.4. Temel Dönüşüm Matrislerinin Bulunması .....	48
4.2. Ters Kinematik Analiz .....	49
<b>5. DENEYSEL DÜZENEK VE ÇALIŞMA .....</b>	<b>54</b>

5.1.	Deneysel Düzenegi Oluşturan Kısımlar .....	54
5.1.1.	Raspberry Pi Devre Kartı ve Gereksinimleri .....	55
5.1.2.	OWI-535 Robot Kolu ve Kontrol Ünitesi .....	58
5.1.2.1.	OWI-535 Robot Kolu .....	58
5.1.2.2.	Robot Kol Kontrol Ünitesi .....	63
5.2.	OWI-535 Robot Kol Kinematik Analizi.....	68
5.2.1.	OWI-535 Robot Kol Düz Kinematik Analizi.....	68
5.2.2.	OWI-535 Robot Kol Ters Kinematik Analizi .....	73
5.3.	DeneySEL Çalıřmalar.....	76
5.3.1.	Nesne Tanıma ve Yer Tespiti Çalıřmaları .....	78
5.3.1.1.	Veri Kümesinin Oluşturulması.....	78
5.3.1.2.	Test İşlemleri.....	82
5.3.2.	Taninan Nesnenin Kavranıp Taşınmasına Yönelik Çalıřmalar .....	103
6.	<b>SONUÇLAR VE ÖNERİLER.....</b>	<b>107</b>
	<b>KAYNAKLAR.....</b>	<b>110</b>
	<b>ÖZGEÇMİŞ.....</b>	<b>116</b>

## ÖZET

Bu tezde, nesne bulma ve tanıma algoritmalarının bir robot kol üzerinde uygulanmasını gerçekleştiren bir deney düzeneği oluşturulmuştur.

Bu çalışmadaki deney düzeneği için, endüstride kullanılan robot kollara benzer tipte olan 4 serbestlik dereceli OWI-535 robot kolu seçilmiştir. Deneysel çalışmalar için ilk olarak, robot kolun çalışma alanını görecektir şekilde bir kamera yerleştirilmiştir. Kamera vasıtasıyla elde edilen görüntüde, kavranması istenen nesneyi tanımak ve yerini tespit etmek için yerel öznitelik tabanlı SIFT, SURF, FAST ve ORB gibi öznitelik çıkarım algoritmalarından faydalanılmıştır. Bu algoritmalar kullanılarak OpenCV kütüphanesi aracılığıyla C++ dilinde yazılan yazılım ile gerçekleştirilen nesne tanıma ve yer tespit işlemleri, Raspberry Pi devre kartı üzerinde gerçekleştirilmiştir. İkinci olarak, nesne tanınıp yeri saptandıktan sonra konum bilgisi robot kol kontrol ünitesine gönderilmiş ve robot kol tutucu ucunun nesneyi kavrayıp istenen bir hedefe bırakması işlemleri gerçekleştirilmiştir. Bu işlemler için robot kol eklemlerinin açıları, robot kolun ters kinematik analiziyle elde edilen eşitliklerin çözümü ile belirlenmiştir.

Deneysel çalışmalarda, SIFT, SURF, FAST ve ORB algoritmalarıyla çıkarılan özniteliklerin veri kümeleri üzerindeki performansları karşılaştırılmıştır. Ayrıca, robot kol tutucusunun nesneyi kavraması ve belirlenen hedefe taşınması için gerekli işlem adımları ayrıntılı bir şekilde incelenmiştir.

**Anahtar Kelimeler:** Nesne bulma ve tanıma algoritmaları, Öznitelik çıkarımı, Raspberry Pi devre kartı, OWI-535 robot kol, Ters kinematik analiz.

## SUMMARY

### IMPLEMENTATION OF OBJECT DETECTION AND RECOGNITION ALGORITHMS ON A ROBOTIC ARM USING RASPBERRY PI CIRCUIT BOARD

This thesis constructs an experimental setup to implement object detection and recognition algorithms on a robotic arm.

4 degrees of freedom robotic arm OWI-535 which is similar type to robotic arms used in industry is chosen for the experimental setup in this study. In the experimental studies, firstly a camera coverage area is arranged to cover all workspace of the robotic arm. Local feature based algorithms such as SIFT, SURF, FAST and ORB are used on the images which are captured via the camera to detect and recognize the target object to be grasped by the gripper of robotic arm.

These algorithms are implemented in the software for object recognition and localization, which is written in C++ programming language using OpenCV library and the software runs on the Raspberry Pi circuit board. Secondly, the location information of target object is sent to control unit of the robotic arm after recognition and localization of the object. Then, the gripper of robotic arm grasps the object and moves to desired location. In these processes, the angles of the robotic arm's joints are determined by the solutions of inverse kinematics equations of the robotic arm.

In the experimental studies, the performance of the features which are extracted with the algorithms such as SIFT, SURF, FAST, and ORB are compared on the data set. Moreover, all process steps of grasping the object and moving to the desired location relating to robotic arm's gripper are analyzed in detail.

**Keywords:** Object detection and recognition algorithms, Feature extraction, Raspberry Pi circuit board, OWI-535 robotic arm, Inverse kinematics analysis.

## ŞEKİLLER LİSTESİ

### Sayfa No

Şekil 2.1. Nesne tanıma sisteminin genel iş akış şeması.....	8
Şekil 2.2. Veri kümesi oluşturma aşamaları.....	10
Şekil 2.3. Yerel öznitelik tabanlı nesne tanıma ve yerini saptama işlemlerinin genel iş akış şeması .....	11
Şekil 2.4. Ölçek uzayının oluşturulması.....	14
Şekil 2.5. İki ölçek uzayı arasındaki Gauss farklarının bulunması .....	15
Şekil 2.6. Örnek nesne görüntüsü ve SIFT anahtar noktalarının gösterimi.....	17
Şekil 2.7. Görüntü gradyanları ve anahtar nokta tanımlayıcısının elde edilmesi.....	19
Şekil 2.8. İkinci dereceden Gauss türevlerine karşılık gelen kutu filtreleri .....	21
Şekil 2.9. İntegral görüntü .....	21
Şekil 2.10. Örnek nesne görüntüsü ve SURF anahtar noktalarının gösterimi.....	22
Şekil 2.11. Haar dalgacık filtreleri .....	23
Şekil 2.12. Yönelim atama .....	23
Şekil 2.13. Anahtar nokta tanımlayıcılarının oluşturulması .....	24
Şekil 2.14. Anahtar nokta tespit işlemi.....	25
Şekil 2.15. Örnek nesne görüntüsü ve FAST anahtar noktalarının gösterimi .....	25
Şekil 2.16. Örnek nesne görüntüsü ve ORB anahtar noktalarının gösterimi.....	27
Şekil 2.17. Yönelim açısı hesaplama gösterimi.....	28
Şekil 2.18. FAST anahtar noktaların Hamming uzaklığı birbirine en yakın olan BRIEF tanımlayıcılarının eşleştirilmesi.....	30
Şekil 2.19. RANSAC algoritması ile aykırı eşleşmelerin tespit edilerek elenmesi.....	31
Şekil 3.1. Otomatik çalışan su makinesi.....	33
Şekil 3.2. İlk endüstriyel robot olan Unimate Robot.....	34
Şekil 3.3. KUKA endüstriyel robot .....	35
Şekil 3.4. İnsan kolu ile robot kolu benzetimi .....	35
Şekil 3.5. 6 eksenli robot kolun serbestlik dereceleri .....	37
Şekil 3.6. Otomotiv endüstrisinde kullanılan robot kol örneği.....	38
Şekil 4.1. Kinematik bir çift üzerinde D-H parametrelerinin belirlenmesi .....	44
Şekil 4.2. Robot kol eklemlerine eksen takımlarının yerleştirilmesi.....	45



Şekil 4.3. Robot kolun örnek bir hareketi sonucu son üç eklemının dönme açılarının ( $\theta_2, \theta_3, \theta_4$ ) ve $\theta_{234}$ yönelim açısının gösterimi .....	50
Şekil 4.4. Robot kol tutucusunun bir noktaya farklı pozisyonlarda yaklaşımı.....	52
Şekil 5.1. Deneysel düzenek.....	54
Şekil 5.2. Raspberry Pi B+ modeli .....	55
Şekil 5.3. Raspberry Pi için gerekli donanımlar .....	56
Şekil 5.4. Raspberry Pi kamera modülü .....	57
Şekil 5.5. LXTerminal ekranı.....	58
Şekil 5.6. OWI-535 robot kolu .....	59
Şekil 5.7. OWI-535 robot kolunun eklemleri ve dönüş yönleri.....	59
Şekil 5.8. Robot kol eklem hareketlerinin açılal dönme sınırları.....	60
Şekil 5.9. DA motoru ve dişli kutusu .....	61
Şekil 5.10. Potansiyometrelerin yerleştirilmesi .....	62
Şekil 5.11. Kuvvet algılayıcı direncin yerleştirilmesi .....	63
Şekil 5.12. FSR'nin kuvvet-direnç karakteristiğı .....	63
Şekil 5.13. Robot kol kontrol ünitesi.....	64
Şekil 5.14. Arduino Mega 2560.....	65
Şekil 5.15. L293D motor sürücü kartı .....	66
Şekil 5.16. L298N motor sürücü kartı .....	67
Şekil 5.17. OWI-535 robot kol uzuvlarının uzunlukları .....	68
Şekil 5.18. OWI-535 robot kol eklemlerine eksen takımlarının yerleştirilmesi.....	69
Şekil 5.19. OWI-535 robot kol mekanik modelinin MATLAB/Simulink ile oluşturulan blok diyagramı.....	72
Şekil 5.20. Robot kolun 4 farklı hareketi .....	72
Şekil 5.21. OWI-535 robot kolun hedefe uygun yönelimi .....	74
Şekil 5.22. Robot kol ana koordinat sistemi.....	75
Şekil 5.23. 4 farklı hareket sonucu robot kolun görünümü .....	76
Şekil 5.24. Deneysel çalışmalar için işlem adımları.....	77
Şekil 5.25. Veri kümesi için kullanılan nesnelere RGB görüntüleri.....	80
Şekil 5.26. Veri kümesi için kullanılan nesnelere gri seviye görüntüleri .....	81
Şekil 5.27. Farklı algoritmalar ile çıkarılan öznitelik vektörlerinin toplam sayıları .....	82
Şekil 5.28. Örnek test görüntüsü-1 .....	83
Şekil 5.29. Örnek test görüntüsü-2 .....	83
Şekil 5.30. Örnek test görüntüsü-3 .....	84

Şekil 5.31. Test görüntülerinin doğru eşleşme yüzdeleri (SIFT+SIFT).....	86
Şekil 5.32. 2 no'lu test görüntüsü için test işlemi (SIFT+SIFT) .....	87
Şekil 5.33. 5 no'lu test görüntüsü için test işlemi (SIFT+SIFT) .....	87
Şekil 5.34. 10 no'lu test görüntüsü için test işlemi (SIFT+SIFT) .....	88
Şekil 5.35. 12 no'lu test görüntüsü için test işlemi (SIFT+SIFT) .....	88
Şekil 5.36. Test görüntülerinin doğru eşleşme yüzdeleri (SURF+SURF) .....	90
Şekil 5.37. 2 no'lu test görüntüsü için test işlemi (SURF+SURF).....	91
Şekil 5.38. 5 no'lu test görüntüsü için test işlemi (SURF+SURF).....	91
Şekil 5.39. 10 no'lu test görüntüsü için test işlemi (SURF+SURF).....	92
Şekil 5.40. 12 no'lu test görüntüsü için test işlemi (SURF+SURF).....	92
Şekil 5.41. Test görüntülerinin doğru eşleşme yüzdeleri (FAST+SURF).....	94
Şekil 5.42. 10 no'lu test görüntüsü için test işlemi (FAST+SURF).....	95
Şekil 5.43. 11 no'lu test görüntüsü için test işlemi (FAST+SURF).....	95
Şekil 5.44. 14 no'lu test görüntüsü için test işlemi (FAST+SURF).....	96
Şekil 5.45. 15 no'lu test görüntüsü için test işlemi (FAST+SURF).....	96
Şekil 5.46. Test görüntülerinin doğru eşleşme yüzdeleri (ORB+ORB).....	98
Şekil 5.47. 2 no'lu test görüntüsü için test işlemi (ORB+ORB) .....	99
Şekil 5.48. 5 no'lu test görüntüsü için test işlemi (ORB+ORB) .....	99
Şekil 5.49. 18 no'lu test görüntüsü için test işlemi (ORB+ORB) .....	100
Şekil 5.50. 19 no'lu test görüntüsü için test işlemi (ORB+ORB) .....	100
Şekil 5.51. Tespit edilen anahtar nokta sayılarının karşılaştırılması .....	101
Şekil 5.52. Öznitelik çıkarım zamanlarının karşılaştırılması .....	102
Şekil 5.53. Piksel-mm dönüşümü için gerekli bilgiler .....	103
Şekil 5.54. Kavranıp taşınması istenen nesnenin içinde bulunduğu örnek bir test görüntüsü .....	104
Şekil 5.55. Nesnenin tanınması ve yer tespiti işlemi (SURF+SURF).....	105
Şekil 5.56. Robot kolun hareket sırası: (a) nesneye ulaşma, (b) nesneyi kavrama, (c) nesneyi taşıma, (d) nesneyi bırakma ve (e) harekete başlamadan önceki pozisyonuna gelme .....	106

## TABLolar LİSTESİ

### Sayfa No

Tablo 3.1. Robot kol eklem tipleri.....	36
Tablo 3.2. Ana eksenlerin eklem tiplerine göre robot manipülatör sınıfları .....	38
Tablo 3.3. Kartezyen robot .....	39
Tablo 3.4. Silindirik robot .....	40
Tablo 3.5. Küresel robot .....	40
Tablo 3.6. SCARA robot .....	41
Tablo 3.7. Eklemli (Mafsallı) robot.....	42
Tablo 4.1. D-H tablosu .....	46
Tablo 5.1. OWI-535 robot kol D-H tablosu.....	69
Tablo 5.2. OWI-535 robot kolunun 4 farklı hareketi için eklem açıları ve tutucu ucun konumu .....	71
Tablo 5.3. OWI-535 robot kol tutucu ucunun bulunduğu konuma göre eklem açıları.....	75
Tablo 5.4. Test işlemleri (SIFT+SIFT).....	85
Tablo 5.5. Test işlemleri (SURF+SURF) .....	89
Tablo 5.6. Test işlemleri (FAST+SURF) .....	93
Tablo 5.7. Test işlemleri (ORB+ORB).....	97
Tablo 5.8. Algoritmaların çalışma sürelerinin karşılaştırılması .....	102
Tablo 5.9. Ağırlık merkezi ve tutucu ucun ulaşması gereken konum koordinatları .....	105

## KISALTMALAR

<b>3B</b>	: 3 Boyutlu
<b>ADC</b>	: Analog-Dijital Çevirici – Analog to Digital Converter
<b>BF</b>	: Kaba-Kuvvet – Brute-Force
<b>BRIEF</b>	: İkili Gürbüz Bağımsız Temel Öznitelikler – Binary Robust Independent Elementary Features
<b>DA</b>	: Doğru Akım
<b>D-H</b>	: Denavit-Hartenberg
<b>DoG</b>	: Gauss Farkı – Difference of Gaussian
<b>FAST</b>	: Hızlandırılmış Segment Testinden Öznitelikler – Features from Accelerated Segment Test
<b>FLANN</b>	: Yaklaşık En yakın Komşuluk için Hızlı Kütüphane – Fast Library for Approximate Nearest Neighbors
<b>FPGA</b>	: Alanda Programlanabilir Kapı Dizileri – Field Programmable Gate Array
<b>FSR</b>	: Kuvvet Algılayıcı Direnç – Force Sensitive Resistor
<b>KEK</b>	: k-En yakın Komşuluk – k-Nearest Neighbor
<b>LoG</b>	: Gauss Laplası – Laplacian of Gaussian
<b>OpenCV</b>	: Open Source Computer Vision
<b>ORB</b>	: Yönlendirilmiş FAST ve Döndürülmüş BRIEF – Oriented FAST and Rotated BRIEF
<b>PWM</b>	: Darbe Genişlik Modülasyonu – Pulse Width Modulation
<b>RANSAC</b>	: Rastgele Örnek Uzlaşımı – RANdom SAMple Consensus
<b>RGB</b>	: Kırmızı Yeşil Mavi – Red Green Blue
<b>SCARA</b>	: Seçimlere Uyan Montaj Robot Kolu – Selective Compliance Assembly Robot Arm
<b>SIFT</b>	: Ölçekten Bağımsız Öznitelik Dönüşümü – Scale Invariant Feature Transform
<b>SoC</b>	: Yongada Sistem – System on Chip
<b>SURF</b>	: Hızlandırılmış Gürbüz Öznitelikler – Speeded Up Robust Features
<b>TBA</b>	: Temel Bileşen Analizi
<b>YSA</b>	: Yapay Sinir Ağları

## 1. GİRİŞ

Günümüzde alışlagelmiş otomasyon teknolojileri, yerini hızlı bir şekilde bilgisayarla görme ve robotik uygulamalara bırakmaktadır. İnsanoğlunun işlevinin büyük bir kısmını yerine getiren bu sistematik yapı hayatımızın her alanında yerini almaktadır. Bilgisayarla görme alanı uygulamalarından olan nesne tanıma, ortak özelliğe sahip veya aralarında bir ilişki kurulabilen nesnelere çeşitli işlemler vasıtasıyla bulup sınıflandırma işlemidir.

Gelişen teknoloji, çeşitli sensörleri kullanımıyla robot kollarını daha kullanışlı ve etkin hale getirmiştir. Son yıllarda birçok endüstri alanında, insanların yerini alan robot kolların kullanımı oldukça artmıştır. Bu endüstri alanları, sıkı rekabet dolayısıyla daha etkin ve daha verimli üretim sistemlerine ihtiyaç duymaktadır. Bilgisayarla görme, birçok endüstriyel işlemlerde potansiyel uygulama sahasına sahip olması sebebiyle robotik alanında önemli bir algılama teknolojisi olmaktadır. Bilgisayarla görme alanındaki şu anki uygulamalarının çoğu henüz araştırma aşamalarında karşımıza çıkmaktadır ancak bilgisayarla görme teknolojisinin robotik alanının geleceğinde giderek artan önemde bir rolü olacağı öngörülmektedir. Nesne tanıma sistemi bunun için iyi bir örnektir. Bu sistem nesnelere ebatlarına ve özelliklerine göre birbirinden ayırmak, nesnelere tam olarak hangi koordinatlarda olduğunu belirlemek veya çalışma için en uygun alanları tespit etmek gibi birçok uygulama alanına sahiptir.

Bilimsel yazında, bilgisayarla görme alanlarında yapılan çalışmalar geniş bir yere sahiptir. Özellikle robotik ve bilgisayarla görme alanlarının etkileşimine yönelik birçok çalışma bulunmaktadır. Bilimsel yazında yapılan çalışmalardan bazılarının ilişkili ayrıntılara aşağıda yer verilmiştir:

Akec, Steiner ve Stenger [1], robot kol hareketinin yörünge takibine dayalı çalışmalar gerçekleştirmişlerdir. Robot kolun tutucu ucuna yerleştirilen kameradan alınan görüntüyle, 6 serbestlik dereceli robot kolunun konveyör bant sistemi üzerindeki kıvrımlı yolu takip etmesi sağlanmıştır.

Carusone ve D'Eleuterio [2], ortam aydınlığı ve çalışma alanı gibi dış etkenlerden etkilenmeyen Yapay Sinir Ağları (YSA'lar) tabanlı bilgisayar görme sistemi gerçekleştirilerek robot kol hareketinin kontrolünü sağlamışlardır.

Modi, Sahin ve Saber [3], CCD kameradan alınan gerçek zamanlı görüntülere uygulanan görüntü işleme algoritmaları ile pinpon topunun koordinatlarını tespit ederek robot kolunun hareketinin kontrolünü sağlamışlardır.

Kert [4], kameradan aldığı görüntülere görüntü işleme teknikleri uygulamış ve dairesel nesnelere konumlarını tespit etmiştir. Tespit edilen nesnelere 3 boyutlu koordinatları, genetik algoritma kullanılarak robot kolunun nesnelere erişim sırası optimize edilmiştir.

Çiçek [5], malzemelerin rengine göre taşıma yapan robot kol tasarımı ve uygulaması yapmıştır. Bu çalışma için, kontrolü mikrodenetleyici tabanlı gerçekleştirilen 5 serbestlik dereceli robot kol ile renk sensörü kullanmıştır.

Bustamente ve Gu [6], robot kol uç işlevcisine yerleştirilen kamera ile alınan görüntüye nesne bulma ve tanıma algoritmaları uygulayarak elektrik prizini algılama işlemini gerçekleştirip koordinatlarını tespit etmişlerdir. Lazer uzaklık sensöründen alınan bilgiler doğrultusunda, robotun şarjı bitmek üzere olduğu anda kendini şarj edebilmesi için robot kol eklemlerinde bulunan servo motorların kontrolü gerçekleştirilmiştir.

Dinh, Dunnigan ve Reay [7], görüntü işleme teknikleri ve radyal tabanlı fonksiyon ağı kullanarak 6 serbestlik dereceli bir robot kolun konum kontrolünü gerçekleştirmişlerdir. Bu kontrol için gerekli olan ters kinematik analiz çözümlerine yeni yaklaşımlar önermişlerdir.

Bayrak ve Sarıtaş [8], 5 serbestlik dereceli bir robot kolu ile hedef tespit etme işlemini gerçekleştirmişlerdir. Bu işlemi yaparken robot kolun karşılaşabileceği engellere çarpmasını önlemişlerdir. Önceden belirlenen hedefler bilgisayara kaydedilmiş ve ortamdaki görüntüde hedef tarama işlemini gerçekleştirmişlerdir. Hedef tespitinden sonra koordinat tespiti algoritmaları ile görüntü üzerindeki hedefin koordinatları belirlenmiştir. Böylelikle robot kolunu istenilen koordinata hareket ettirmişlerdir.

Luo ve arkadaşları [9], insansız robot kullanarak güvenlik uygulamaları gerçekleştirmişlerdir. Robotun gözlerindeki kameralar vasıtasıyla ortamın görüntüsü alınmış, güvenlik tehdidi oluşturan nesnelere tespit edilerek ters kinematik analiz sayesinde müdahale edilmiştir. Hedef nesnelere eğer robot kollarının çalışma aralığı dışında ise, üzerindeki mesafe sensörleri vasıtasıyla yakınlık ve uzaklık değiştirilerek nesnelere almayı başarmışlardır.

Erşan [10], 5 serbestlik dereceli bir robot kolun çalışma alanı içerisindeki nesneyi, kamera vasıtasıyla algıladıktan sonra bulunduğu konumdan başka bir konuma taşıma işlemini gerçekleştirmiştir.

Arserim [11], aktif bir kamera sistemi ile aldığı nesne görüntülerinin özelliklerini Zernike moment yöntemi ile çıkarmış ve nesnelere tanımak için YSA sınıflandırıcı algoritması önermiştir. FPGA ve kamera çifti kullanılarak önerilen stereo görme sayesinde robot kolunun gerçek zamanlı kontrolünü başarmıştır.

Anh ve Song [12], robot kol tutucusuna bađlı bir kamera ile yerde hareket halinde olan nesnelere yakalayabilen bir sistem geliřtirmişlerdir. Robot kol eklemlerini servo motorlar ile kontrol etmişlerdir. İlk olarak görüntü alınmış ve daha sonra alınan görüntüler önceki görüntüler ile karşılaştırılmıştır. Hareket algılandığı zaman, hareket halinde olan nesnenin konumu tespit edilmiştir.

Seelye ve arkadaşları [13], uzaktan kumandalı robot kol kullanarak, robot kolun tutucu ucuna yerleştirilen kamera ile bitkilerin büyümelerini gözlemlemişlerdir. Bitki büyüdükçe robot kol ile kamera açısını ayarlayarak istenilen bölgelerin gelişimini takip etmişlerdir.

Huang [14], bir düzlemde bulunan nesnelere görüntülerini kamera ile aldıktan sonra, bu görüntülerde bulunan nesnelere şekillerine göre ayırtmış ve yatay olarak hareket eden robot kol kullanarak dairesel olan nesnelere bir başka yere yerleřtirmişlerdir. Dairesel nesnelere tespiti için kenar belirleme algoritmaları kullanmışlardır.

Rapp [15], robot kol uç işlevcisine bađlı bir raket üzerinde pinpon topunu takip eden ve robot kola gelen topa vurabilen bir sistem tasarlamıştır. Bu sistemde 2 adet kamera kullanılmıştır. Kameralardan biri raketi üstten, diđeri yandan görecekte şekilde yerleştirilmiştir. Sistem, endüstriyel bir bilgisayar tarafından kontrol edilmiştir. Böylece, topun yukarı doğru yükselmesi ile raketin vurma hızı, sađa sola kayması ile de raketin konumu ayarlanmıştır. Bu sistem yaklaşık yarım saat boyunca yanlış vuruř yapmadan topu sektirebilmiştir.

Ali ve arkadaşları [16], kameradan alınan görüntüdeki nesnelere şekilsel olarak incelemişlerdir. Kare, dikdörtgen, yuvarlak vb. şekilde olan nesnelere tanımışlar ve bilgisayara bađlı bulunan robot kol ile istedikleri nesnelere alıp istenilen herhangi bir yere yerleřtirmişlerdir.

Erdoğan [17], kamera, robot kol ve kontrol ünitesinden oluşan sistemde, nesne görüntülerinin özelliklerini çıkarmak için bir görüntü işleme algoritması ve çıkarılan bu özellikleri kullanarak nesnelere tanınması için bir YSA algoritması kullanmıştır. Böylece, nesnelere görüntü içerisindeki yerine ve açısına bađlı kalmadan sınıflandırılması ve robot kol ile istenen yere yerleřtirilmesi işlemini gerçekleřtirmiştir.

Basile ve arkadaşları [18], istenen hedef nesnelere koordinatlarını belirleyerek birden fazla robot kolunu eş zamanlı olarak çalıştırmışlardır. Koordinatlar belirlendikten sonra ters kinematik analiz ile robot kollardan hedef için en uygun olanı tespit edilmiş ve bu robot kolu nesnenin kavranıp alınması için harekete geçirilmiştir.

Şenel [19], gömülü sistem ve görüntü işleme tekniklerini bir arada kullanarak robot kollar vasıtasıyla üretim bandı üzerinde hareket eden nesnelere üretim hatası olanların ayrıştırılması işlemini gerçekleştirmiştir.

Prabhu ve Soman [20], ses komutları ve nesne tanıma algoritmaları vasıtasıyla söylenen cismin bir noktadan bir noktaya hareketini gerçekleştirmişlerdir.

İnan [21], kameradan alınan görüntüye gerekli görüntü işleme teknikleri uygulayarak hareket halinde olan hedef cismin tespitini yapmış ve robot kolun hedef olarak tespit edilen cisme doğru yönelip tutucu ucu ile robot kol çalışma alanının merkezine almasını sağlamıştır.

Son zamanlarda 3 boyutlu sensör teknolojisinin gelişimiyle, 3 boyutlu bilgisayarla görme teknikleri üzerine yapılan araştırmalar birçok yönde ilerlemiştir. Bunlardan bir tanesi de 3 boyutlu nesne tanımadır ve bu konuda çeşitli araştırmalar yapılmıştır [22-25].

Nesne bulma ve tanıma uygulamaları, genel olarak kullanım amacına göre, görünüm tabanlı veya yerel öznitelik tabanlı yaklaşımlar kullanılarak yapılmaktadır.

Görünüm tabanlı yaklaşımda, nesnelere şablon veya model görüntüleri kullanılır. Şablon veya model, arama ortamında gezdirilerek uygun düşen eşleşmeler, mesafe ölçümlerinden de faydalanılarak nesne tanıma işlemi gerçekleştirilir [26].

Yerel öznitelik tabanlı yaklaşım, görünüm tabanlı yaklaşımların yetmediği, aydınlanma değişiklikleri veya nesnenin bir başka nesne tarafından kısmen kapanmış olduğu durumlarda kullanılırlar. Yerel öznitelikler, nesnelere hakkında bilgiler içeren belli bölgeler olarak ifade edilebilir [27]. Bu yaklaşımdan elde edilen öznitelik vektörleri; merkeze olan uzaklıklar, çevreyi oluşturan eğrilerin eğrilik dereceleri ve köşeler gibi tanımlayıcılardır [28]. Bu öznitelikler sayesinde nesnelere, bütününden bağımsız olarak tanımlanabilir olacaktır.

Nesne tanıma uygulamalarında ortaya çıkan zorluklardan dolayı, nesne tanıma algoritmaları geniş bilimsel yazına sahiptir. Bilimsel yazındaki bazı çalışmalara aşağıda yer verilmiştir:

Yerel öznitelik tabanlı yaklaşım ile görüntü eşleştirme çalışmalarına Morevec'in köşe bulucusu [29] ile başlanmıştır. Morevec'in geliştirdiği bu bulucuyu, görüntüdeki küçük ayrıntılar ve birbirine yakın konumdaki kenarlar üzerinde daha iyi sonuçlar verecek şekilde Harris ve Stephans [30] geliştirilmiştir. [31]'de Harris, geliştirilen bulucuyu hareket takibi ve 3 boyutlu yapıların yeniden oluşturulması için kullanmıştır.

Schmid ve Mohr [32], Harris bulucusu kullanarak seçtikleri noktalar ile yerel öznitelikler oluşturmuşlar ve yönelimden bağımsız vektörler tanımlayarak, başarılı sonuçlar elde



etmişlerdir. Yerel öznitelik eşleştirmenin geliştirilmesi ile geniş görüntü veri kümeleri içinde öznitelik aramasını da başarıyla yapmışlardır.

Nelson ve Selinger [33], görüntü dış hatlarının gruplanması temeline dayanan yerel özniteliklerle iyi sonuçlar elde etmişlerdir.

Görünüm tabanlı nesne tanıma yaklaşımları kullanan; Murase ve Nayar [34], özuzay eşleştirmelerini, Swain ve Ballard [35], renk histogramlarını, Schiele, Bernt ve Crowley [36], alıcı alan histogramlarını kullanmışlardır. Tüm bu yaklaşımlar yalıtılmış veya önceden parçalanmış nesnelere üzerinde başarılı bir şekilde çalışmalarına rağmen karışık ve kısmen kapanmış görüntülerde sorun yaşamışlardır.

Köşe noktaları ile yerel öznitelik vektörlerini oluşturan nesne tanıma algoritmalarının karşılaştığı ortak sorun; tek bir ölçekte çalıştıklarında başarılı olmaları, farklı ölçeklerde çalışıldığında ise, her ölçekte belirlenen noktaların farklı konumlarda olması sebebiyle ölçekten bağımsızlığın sağlanamaması olmuştur. Lowe tarafından geliştirilen ölçekten bağımsız öznitelik dönüşümü, kısaca SIFT (Scale Invariant Feature Transform) olarak isimlendirilen algoritma ile köşe noktaları kullanılarak nesnenin yerel öznitelikleri, ölçekten bağımsız olarak çıkarılmıştır [37].

Ledwich ve Williams [38], iç mekânlarda çalışarak SIFT özniteliklerinin karmaşıklığını ve ortamı tanımlayan öznitelik sayısını azaltmışlardır. Özniteliklerin boyutlarının ve karmaşıklığının azaltılmasıyla eşleştirme süresi kısaltılmıştır.

Guan ve arkadaşları [39], yerel öznitelikleri çıkarmak için hızlandırılmış gürbüz öznitelikler, kısaca SURF (Speeded Up Robust Features) olarak isimlendirilen algoritmayı geliştirerek hızlı ve etkili bir biçimde nesne tanıma işlemini gerçekleştirmişlerdir. Bu algoritmanın sonucunda, nesne görüntüsünün ölçekten bağımsız olduğu, dönmeye karşı dayanıklı olduğu ve görüntünün bir kısmının kapanması durumunda bile, sadece SIFT ve hem Temel Bileşen Analizi (TBA) hem de SIFT algoritmalarının birlikte kullanıldığı duruma göre daha dayanıklı olduğu sonucuna varmışlardır.

Heo ve arkadaşları [40], nesne tanıma için hızlandırılmış segment testinden öznitelikler, kısaca FAST (Features from Accelerated Segment Test) isimli algoritma ile ikili gürbüz bağımsız temel öznitelikler, kısaca BRIEF (Binary Robust Independent Elementary Features) isimli algoritmayı birlikte kullanarak öznitelikleri çıkarmışlardır. Birlikte kullanılan bu algoritmaların SIFT veya SURF algoritmalarından herhangi birine karşı hız ve bellek kapasitesi yönünden en az yüzde elli daha iyi olduğu sonucunu ortaya koymuşlardır.

Rublee ve arkadaşları [41], SIFT ve SURF algoritmalarının nesne bulma ve eşleştirme için fazla sayıda öznelik kullanmasından dolayı, bu algoritmalara bir seçenek olarak BRIEF ve FAST algoritmalarının iyi özelliklerini birleştirmeyi amaçlayan yönlendirilmiş FAST ve döndürülmüş BRIEF, kısaca ORB (Oriented FAST and Rotated BRIEF) olarak isimlendirilen algoritmayı kullanmışlardır. ORB'nin, nesne görüntüsünün dönmesine karşı dayanıklı ve gürültüye karşı dirençli olduğu gösterilmiştir.

Bu tez çalışmasında, endüstriyel bir robot kol benzeri olan, 4 serbestlik derecesi ve bir tutucu ucu bulunan OWI-535 robot kolu kullanarak tutucu ucun, çalışma alanındaki tanınan bir nesneyi kavradıktan sonra belirli bir hedefe bırakması amaçlanmıştır. Bu amaca uygun deneysel bir düzenek kurulmuştur. Çalışmada, Linux işletim sistemi tabanlı tek kart bilgisayar olan Raspberry Pi devre kartı kullanılarak OpenCV (Open Source Computer Vision) kütüphanesiyle C++ dilinde yazılan yazılım ile nesne tanıma ve yer tespiti işlemleri gerçekleştirilmiştir. Bu yazılım ile birinci olarak, robot kolun çalışma alanına yerleştirilen tanınması istenen nesnelerin görüntüleri alınıp veri kümesi oluşturulmuştur. İkinci olarak, alınan test görüntülerine bazı yerel öznelik tabanlı nesne bulma ve tanıma algoritmaları uygulayarak test görüntüsünde yer alan nesnelerin veri kümesindeki hangi nesne ile aynı olduğu belirlenmiştir. Üçüncü olarak, robot kol tutucu ucunun kavrayacağı nesnenin ağırlık merkezi piksel cinsinden tespit edilmiş ve robot kol tutucusunun yönelimi için gerekli olan konum koordinatlarına dönüştürülmüştür. Son olarak, belirlenen koordinat değerleri Raspberry Pi devre kartı üzerinden, robot kolun hareketini gerçekleştirecek olan mikrodenetleyici tabanlı Arduino Mega kartına seri haberleşme ile gönderilmiştir. Robot kol tutucusunun bu koordinat değerlerine hareketi için robot kol eklemlerinin bulunması gereken eklem açıları ters kinematik analizi ile belirlenerek robot kolun hareketi gerçekleştirilmiştir.

Sonuç olarak, bu tez çalışması için oluşturulan deney düzeneği ile nesne bulma ve tanıma algoritmalarının başarısının eğitim ve endüstri alanlarındaki uygulamalar için elverişli olduğu görülmüştür.

## **1.1. Tezin Organizasyonu**

Tez, 6 bölümden oluşmaktadır. Birinci bölümde, bilgisayarla görme ve robotik alanlarının etkileşimi ile ilgili olan tezin hedefi ve gerekli işlem adımları sunularak bu alanlardaki bilimsel yazın taramasına yer verilmiştir.

İkinci bölümde, tez çalışmasında kullanılan robot kolun tutucu ucunun kavraması istenen nesnenin tanınması ve yerinin tespiti için gerekli olan işlem adımları anlatılmış ve en önemli adım olan öznitelik çıkarımında kullanılan nesne bulma ve tanıma algoritmalarına ayrıntılı olarak yer verilmiştir.

Üçüncü bölümde, robotik ve robot manipülatörler hakkında bilgiler verilmiş olup genel olarak robotların tarihsel gelişimi, robot manipülatörlerin eklem tiplerine göre sınıflandırılmasına yer verilmiştir.

Dördüncü bölümde, robot kol kinematik analizi hakkında bilgilere yer verilmiş olup tez çalışmasında 4 serbestlik dereceli robot kol kullanıldığı için düz (ileri yön) ve ters kinematik analizin işlem adımları 4 serbestlik dereceli robot kol üzerinde gerçekleştirilmiştir.

Beşinci bölümde, öncelikle tez çalışması için kurulan deneysel düzenek ve düzeneği oluşturan kısımlardan bahsedilmiş ve OWI-535 robot kol için kinematik eşitlikler elde edilmiş ve eşitliklerin çözümleri gerçekleştirilmiştir. Daha sonra deney düzeneğinde yer alan kamera aracılığıyla elde edilen test görüntülerine nesne bulma ve tanıma algoritmaları uygulanıp çalışma alanı içerisinde bulunan nesnelere hangisinin aranan nesne olduğu tespit edilerek robot kol tutucusunun kavraması gereken nesnenin koordinatları belirlenmiştir. Uygulanan nesne bulma ve tanıma algoritmalarının çeşitli test görüntüleri üzerindeki karşılaştırmalı sonuçlarına da yer verildikten sonra, belirlenen koordinatlarla OWI-535 robot kolunun hesaplanan kinematik denklem sonuçları değerlendirilerek robot kolun hedef noktaya hareketi gerçekleştirilmiş ve nesnenin kavranıp taşınması işlemi gerçekleştirilmiş olup yapılan uygulamadan örnekler sunulmuştur.

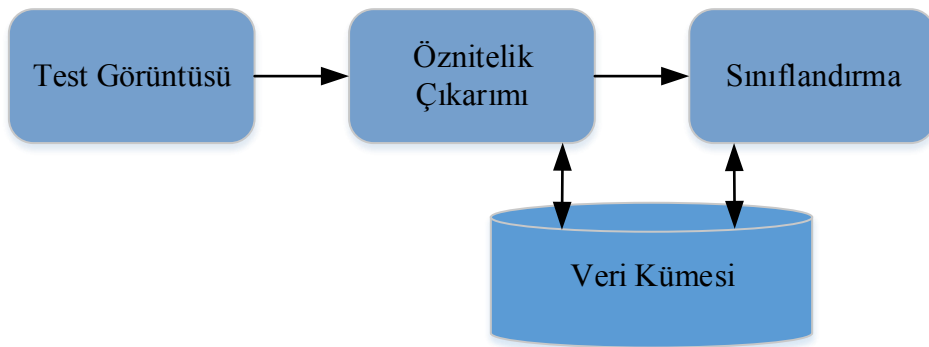
Altıncı bölümde ise elde edilen sonuçlar ve öneriler sunulmuştur.

## 2. NESNE BULMA VE TANIMA

Bilgisayarla görme alanlarından olan nesne bulma ve tanıma, son yıllarda bilim adamlarının çoğu için en cezbedici araştırma konularından biri haline gelmiştir. Nesne bulma; bir görüntüde bulunması istenen nesnenin, görüntünün neresinde olduğunu belirleme veya nesnenin yerini sınırlayıcı kutu ile gösterme işlemidir. Bir örnek ile açıklamak gerekirse, silgi nesnesinin bulunduğu bir görüntüde silginin görüntünün sol üstünde yer aldığını belirleme işlemi diyebiliriz. Nesne tanıma ise; bilinmeyen nesnelere içeren bir görüntüde nesnelere sınıf etiketlerini (isimlerini) ve de istenirse konumlarını belirleme işlemidir. Nesne etiketleri, belli bir kategori içerisinde yer alan yani bilinen sınıflar arasından elde edilir. Tanıma işlemine örnek olarak, nesnenin bulunduğu bir görüntüde silginin yerini bulup “silgi” olarak isimlendirilmesi verilebilir.

Nesne tanıma alanındaki çalışmalar, L. Roberts tarafından 1960’lı yıllarda başlatılmış ve artarak günümüze kadar devam etmiştir [42]. Nesne bulma ve tanıma, tıp alanında tanı koyma, savunma sanayiinde güvenlik veya hedef tanıma, fabrikalarda otomasyon sistemleri gibi çeşitli alanlarda kapsamlı uygulamalara sahiptir. Bu yüzden nesne bulma ve tanıma algoritmaları bu uygulamalarda çok önemli bir yere sahiptir.

Şekil 2.1’de gösterildiği gibi genel olarak nesne tanıma sistemleri; nesne görüntülerinin özelliklerinin çıkarılıp saklandığı bir veri kümesi ve test görüntüsünden çıkarılan özelliklerin veri kümesindeki özelliklerle eşleştirilip nesnenin sınıflandırıldığı belli aşamalardan oluşmaktadır [43].

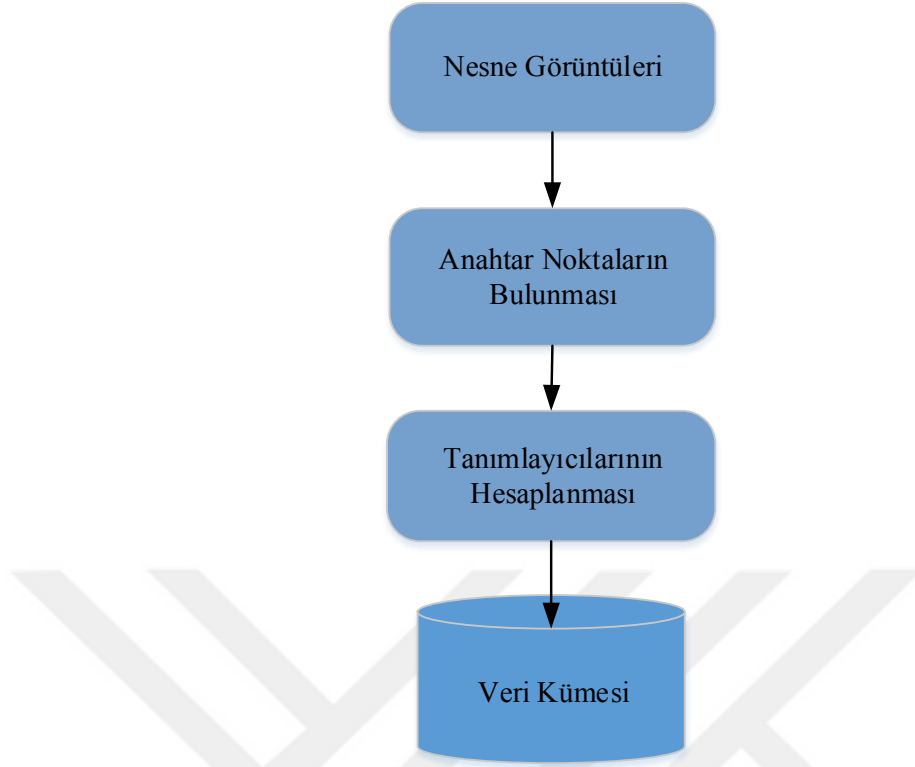


Şekil 2.1. Nesne tanıma sisteminin genel iş akış şeması

Nesne tanıma işleminin en önemli aşaması olan öznitelik çıkarımı için genel olarak görünüm tabanlı ve yerel öznitelik tabanlı algoritmalar kullanılmaktadır. Son yıllarda endüstri ve tıp gibi hatanın kabul edilemediği alanlarda nesne görüntülerinin farklı ölçek, farklı dönme, kapanma veya aydınlanma gibi dış etmenlere maruz kaldığı durumlarda nesne takibi, görüntü eşleştirme, nesne yeri saptama ve nesne tanıma gibi uygulamaları gerçekleştirebilmek için nesnelerin öznitelik çıkarımında yaygın olarak yerel öznitelik tabanlı algoritmalar kullanılmaktadır.

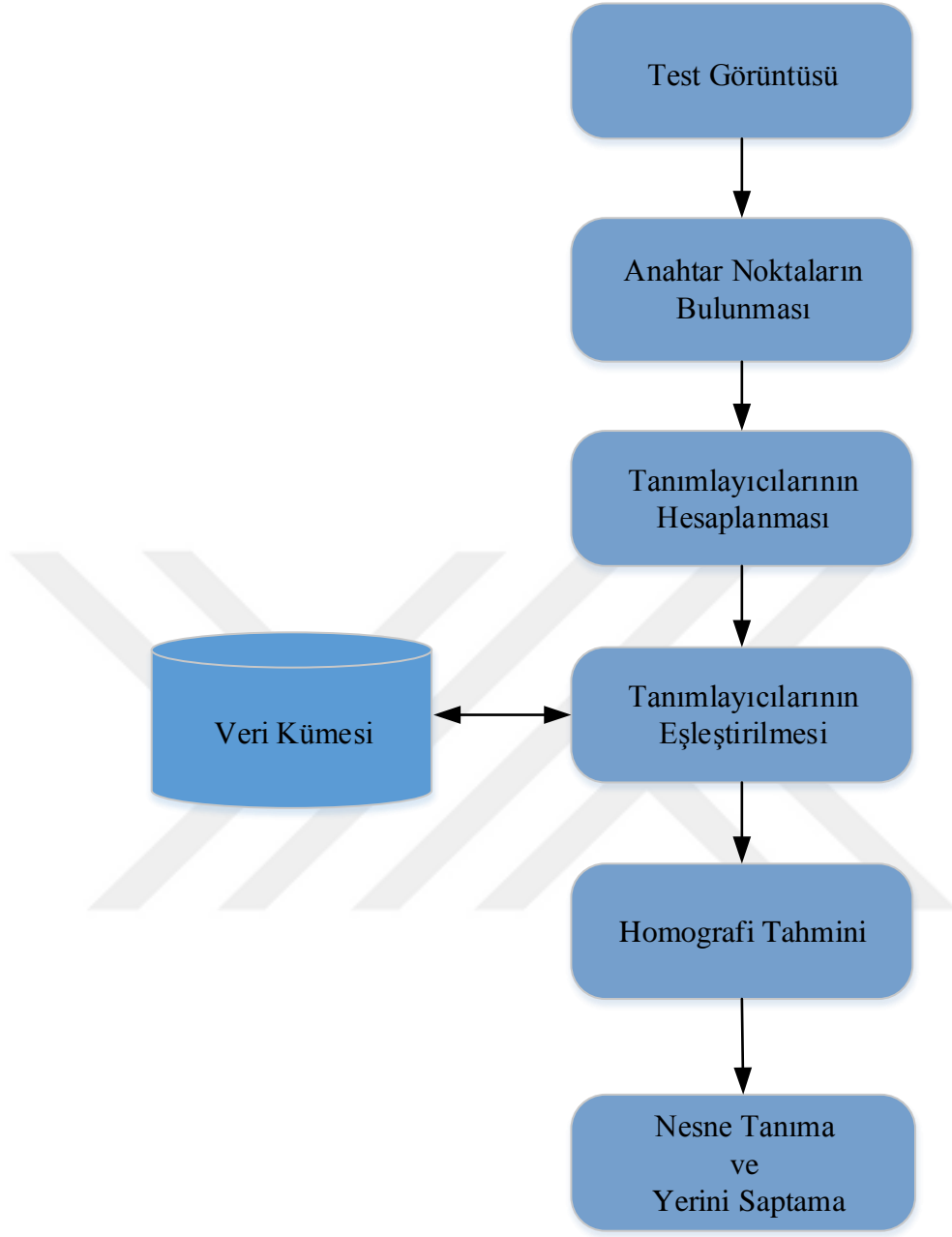
Yerel öznitelik tabanlı algoritmalar kullanılarak çıkarılan öznitelikler, görüntüde bulunan anahtar noktaların tanımlayıcıları olarak ifade edilmektedir. Görüntüde bulunan bu anahtar noktalar parça, kenar, köşe veya küçük yuvarlak olabilmekte ve görüntüyü gereksiz bilgilerden arındırarak nesne tanıma işlemini kolaylaştırmaktadır.

Bu tez çalışmasında, nesne tanıma işlemi için öznitelik çıkarımında yerel öznitelik tabanlı algoritmalar kullanılmıştır. Bu işlemler için oluşturulan sistemde, ilk olarak sistemin eğitimi için veri kümesinde yer alması istenen nesnelerin her bir görüntüsünün öznitelikleri çıkarılır. Öznitelikler, nesne görüntülerinde yer alan anahtar noktaların bulunup, bu anahtar noktaların tanımlayıcılarının hesaplanmasıyla çıkarılır. Bu tanımlayıcılar, ait oldukları nesnelerin sınıf etiketleri ile birlikte veri kümesinde saklanır. Şekil 2.2’de veri kümesinin oluşturma aşamaları gösterilmiştir.



Şekil 2.2. Veri kümesi oluşturma aşamaları

Öznitelik vektörlerinin yer aldığı veri kümesi oluşturulup eğitim aşaması tamamlandıktan sonra verilen bir test görüntüsünde yer alan nesneyi uygun bir sınıfa atamak için test görüntüsünün anahtar noktaları bulunup tanımlayıcıları hesaplandıktan sonra bu tanımlayıcılar, veri kümesinde saklanan tanımlayıcıların her biri ile ayrı ayrı eşleştirilir. Eşleştirilen tanımlayıcılar arasındaki en küçük uzaklık değeri referans alınarak, belirlenen eşik değerinin altında kalan eşleşmeler belirlendikten sonra en fazla eşleşmenin olduğu nesne görüntüsüne ait olan nesne sınıfı belirlenerek test görüntüsündeki nesne sınıflandırılır. Nesne uygun olan sınıfa atandıktan sonra en fazla eşleşmenin olduğu nesne ile test görüntüsü arasındaki en doğru eşleşmeleri belirlemek için hatalı eşleşen noktalar homografi tahmini ile filtrelenir ve tanınan nesne atandığı sınıf etiketiyle etiketlenilerek sınırlayıcı bir kutu içine alınır. Sınırlayıcı kutunun da ağırlık merkezi belirlenerek nesnenin yerini gösteren koordinatlar tespit edilir. Gerçekleştirilen yerel öznitelik tabanlı nesne tanıma ve yerini saptama sisteminin genel aşamaları Şekil 2.3'te verilmiştir.



**Şekil 2.3.** Yerel öznitelik tabanlı nesne tanıma ve yerini saptama işlemlerinin genel iş akış şeması

## 2.1. Öznitelik Çıkarımı

Nesne tanıma işleminde sınıflandırılacak bir nesne görüntüsü genellikle çok fazla miktarda gereksiz bilgi içermektedir. Bu durum, sınıflandırma hassasiyetini düşürürken işlem süresini de yükseltmektedir. Bu olumsuzluğu gidermek için, nesne bilgisi daha düşük boyuttaki başka bir veri kümesine dönüştürülür. En basit ifadeyle boyut indirgeme işlemi olarak tanımlanabilen öznitelik çıkarımı, nesneye ait fazla ve gereksiz verinin elenip, sadece nesneyi temsil eden ve toplam veriden çok düşük boyuttaki karakteristik bilginin elde edildiği dönüşüm işlemidir.

Bu tez çalışmasında, robot kol tutucu ucunun kavrayacağı nesnenin tanınması ve yerinin tespit edilmesi ana hedeftir. Görüntüdeki dönme, aydınlanma veya kamera görüş açısından kaynaklanabilecek hatalar belirtilen hedefe ulaşamamaya neden olur. Uygun bir şekilde çıkarılan öznitelikler nesne bulma ve tanıma gibi uygulamaların başarısını olumlu yönde etkiler. Bu nedenle, tezde nesne tanıma için gerekli olan öznitelik çıkarımında bilimsel yazında başarısı kanıtlanmış SIFT, SURF, FAST ve ORB gibi yerel öznitelik tabanlı algoritmaların kullanılması tercih edilmiştir.

Çalışmada kullanılan yerel öznitelik tabanlı bu algoritmaların işlem adımları, kameradan alınan RGB (Red Green Blue–Kırmızı Yeşil Mavi) görüntülerin gri seviyeye dönüştürülmesinden sonra gerçekleştirilmiştir.

### 2.1.1. Ölçekten Bağımsız Öznitelik Dönüşümü Algoritması

Lowe tarafından görüntüdeki öznitelikleri çıkarmak için 2004 yılında kısaca SIFT olarak isimlendirilen ölçekten bağımsız öznitelik dönüşümü algoritması önerilmiştir. SIFT'in bilimsel yazındaki diğer algoritmalarından üstün yanları; görüntünün ölçeğinden, görüntünün alındığı kameranın bakış açısından kaynaklanan küçük açı değişimlerinden ve ortamın ışık koşullarından etkilenmeksizin öznitelikleri çıkarabilmesidir [44]. Diğer taraftan, yüksek çözünürlüklü görüntülere sahip nesnelere algılama için uygun bir algoritma olmasına rağmen, algoritmanın gerçekleşme süresi bakımından yavaştır [45].

SIFT algoritması, aşağıdaki gibi temel olarak 4 adımdan oluşmaktadır:

- 1) Ölçeksel uzaydaki tepe (uç) noktaların tespiti,
- 2) Anahtar noktaların yerinin saptanması,
- 3) Anahtar noktalara yönelim atanması,



4) Anahtar nokta tanımlayıcılarının oluşturulması.

### 2.1.1.1. Ölçeksel Uzaydaki Tepe (Uç) Noktaların Tespiti

İlk olarak görüntü bulanıklaştırılır ve daha sonra görüntünün boyutu yarıya düşürülerek ikinci seviye görüntü oluşturulur ve tekrar bulanıklaştırılır. Bu işlem her defasında tekrarlanarak Şekil 2.4'te görüldüğü gibi 4 defa gerçekleştirilir. Aynı boyuta sahip görüntülerin Oktav değeri 4, bulanıklaştırma değeri 5 alınır.

Farklı ölçek değerlerinde değişmeyen öznelikleri tespit edebilmek için (2.1)'de hesaplanan Gauss ölçek değeri, görüntü ile konvolüsyon yapılarak (2.2)'deki gibi ölçek uzay fonksiyonu elde edilir [44].

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.2)$$

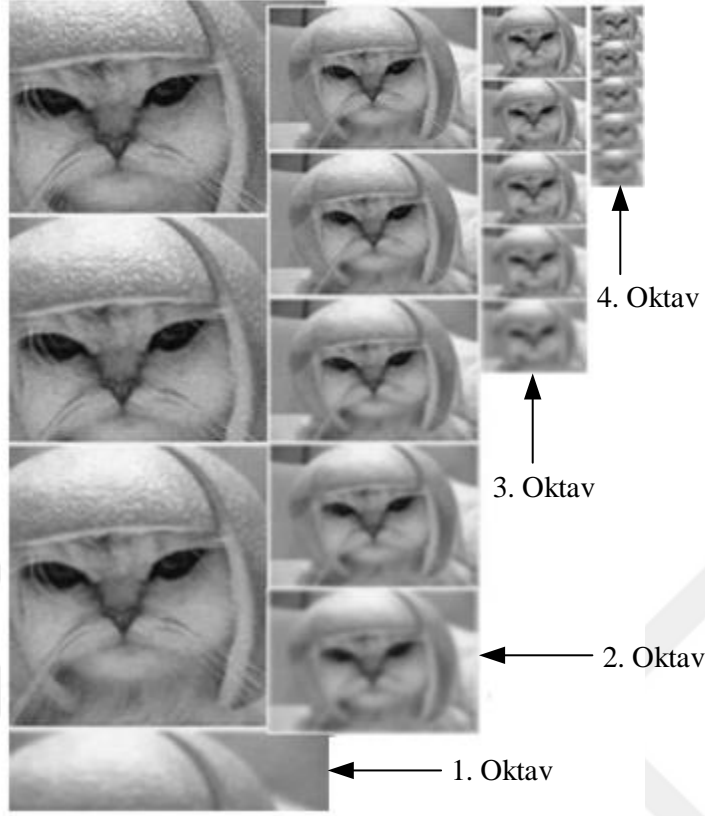
\* : Konvolüsyon operatörü,

L : Ölçek uzay fonksiyonu (Bulanıklaştırılan görüntü),

I : Giriş görüntüsü,

x, y : Piksel konumu,

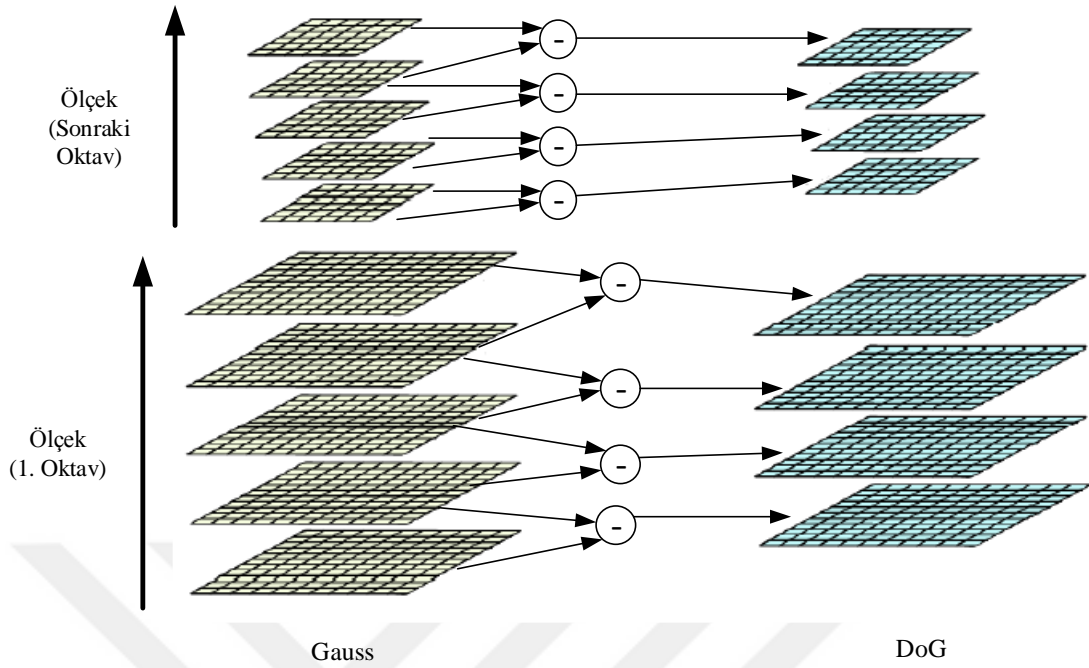
$\sigma$  : Gauss ölçeğinde kullanılan standart sapma değeri.



Şekil 2.4. Ölçek uzayının oluşturulması

$\sigma$ , bulanıklaştırma miktarını ifade etmektedir. Öznitelik çıkarımında bulanıklaştırma miktarı önemli olup bulanıklaştırma miktarını  $\sigma$  alırsak, bir sonraki adımda  $k \times \sigma$  alınır.  $k$ 'nın optimum değeri  $k = \sqrt{2}$  olarak belirlenmiştir [44].

Görüntünün belirli bir oranda bulanıklaştırılıp ikinci türevinin alınması işlemi olan LoG (Laplacian of Gaussian–Gauss Laplası) ile görüntüdeki anahtar noktaları bulmaya yardımcı olan görüntünün kenar ve köşeleri bulunur. Ancak bu işlem, hesapsal açıdan maliyetli olduğundan bunun yerine benzer sonuçları basit bir çıkarma işlemi ile veren DoG (Difference of Gaussian–Gauss Farkı) işlemi kullanılmaktadır. Şekil 2.5'te DoG işlemi örnek olarak gösterilmiştir.



Şekil 2.5. İki ölçek uzayı arasındaki Gauss farklarının bulunması

LoG işlemi Gauss işleminde yer alan bulanıklaştırma miktarına ( $\sigma$ ) bağlı olduğundan ölçekten bağımsız değildir. (2.1)'de görüldüğü gibi varyans ( $\sigma^2$ ) parametresinin paydada olması bu ifadenin ölçeğe bağımlı olduğu anlamına gelmektedir. DoG işleminde  $\sigma^2$  ile çarpma işlemi gerçekleştiği için ölçekten bağımsızdır. Bu işlem sayesinde SIFT algoritması ölçekten bağımsız bir şekilde gerçekleşmektedir. Görüntüde birbirine komşu olan aynı oktavdaki iki ölçek uzayının aralarındaki fark (2.3) ile DoG bulunarak ölçek uzaydaki tepe noktaların yerleri yaklaşık olarak bulunarak görüntüdeki anahtar noktaların kontrolü yapılmış olur.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.3)$$

### 2.1.1.2. Anahtar Noktaların Yerinin Saptanması

Tepe noktaların konumlarının iyi bir şekilde tespit edilmesi gerekmektedir. Bulunan tepe noktaları ile bulunması gereken tepe noktaları arasında bir kayma olabilir. Tepe noktaların yerinin tam olarak bulunması için Brown'un geliştirdiği 3B quadratik fonksiyon kullanılmaktadır [46]. Öncelikle,  $D(x, y, \sigma)$  üzerinde (2.4)'teki gibi Taylor açılımı kullanılır [44].

$$D(x) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.4)$$

$x = (x, y, \sigma)^T$ : Tepe noktalarından olan uzaklık

(2.4)'teki DoG fonksiyonu olan  $D(x)$  ve türevleri tepe noktaları üzerinde hesaplanmıştır. Tepe noktaların tam yeri olan  $\hat{x}$ ,  $D(x)$  fonksiyonunun türevinin alınıp 0'a eşitlenmesi ile (2.5)'teki gibi bulunur.

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{dD}{dx} \quad (2.5)$$

Daha sonra düşük kontrasta sahip pikseller belli bir eşik değere tabi tutulur ve eşik değerin altında kalan pikseller (2.6) kullanılarak elenir. Örneğin; [0 1] aralığındaki piksel değerlerinden oluşan görüntüde 0.03 değerinden küçük olan tüm  $D(\hat{x})$  tepe değerleri elenebilir [44].

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (2.6)$$

$\hat{x}$ , 0.5 değerinden büyükse başka bir tepe noktaya daha yakındır. Bu durumda tepe nokta değiştirilir. Değişen  $\hat{x}$  değeri, o tepe noktanın konumu olarak eklenir. Bu sayede tepe noktaların tam yeri bulunmuş olur.

Gauss fonksiyonu kenar noktalarda yüksek tepkiler verdiği için kenar noktaların ortadan kaldırılması gerekir. Bunun için Harris ve Stephens'ın "kenar çıkarma" yöntemi kullanılır. Bir anahtar nokta etrafındaki birbirine dik olan iki gradyan incelenir. Bu gradyan değerlerine göre anahtar nokta etrafında kenar, köşe ya da düz bir alanın olup olmadığı anlaşılır.

Anahtar nokta olarak köşeler en verimli olanlardır ve bu nedenle köşeler bulunarak diğerleri elenir. Bir noktanın köşe olup olmadığının tespiti için (2.7)'deki "Hessian matrisi" kullanılır.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (2.7)$$

Hessian matrisindeki özdeğerleri ( $D_{xx}, D_{xy}, D_{yy}$ ) hesaplamak yerine (2.10)'da verilen oran şartını kullanmak verimliliği arttırmaktadır.

$$\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (2.8)$$

Hessian matrisi simetriktir. Yani  $D_{xy} = D_{yx}$ 'tir. Böylece Hessian matrisinin determinanı (2.9)'daki gibi yazılabilir.

$$\text{Det}(H) = D_{xx} \times D_{yy} - D_{xy}^2 = \alpha \times \beta \quad (2.9)$$

$\text{Det}(H) < 0$  ise nokta tepe noktası değildir ve elenir.

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r + 1)^2}{r} \quad (2.10)$$

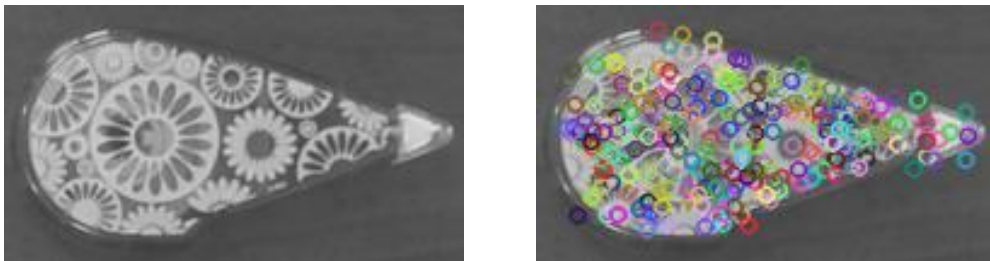
$\alpha$  : En büyük gradyan büyüklüğüne sahip özdeğer,

$\beta$  : Daha küçük gradyan büyüklüğüne sahip özdeğer,

$r$  :  $\alpha$  ve  $\beta$  arasındaki oran (genel olarak 10 kabul edilir.).

(2.10)'daki eşitliği sağlayan noktalar köşe noktaları ifade eder ve diğer gereksiz noktalar elenerek anahtar noktalar belirlenmiş olur. Bu sayede verim artırılmış olur.

Şekil 2.6'da tez çalışmasında kullanılan bir nesne görüntüsünün SIFT algoritması ile bulunan anahtar noktalarının gösterimine yer verilmiştir.



Şekil 2.6. Örnek nesne görüntüsü ve SIFT anahtar noktalarının gösterimi

### 2.1.1.3. Anahtar Noktalara Yönelim Atanması

Belirlenen her anahtar noktaya gradyan yönelimi atanarak, anahtar noktaların dönmeye karşı bağımsız olması sağlanmaktadır. Bu işlem, bir bölge seçerek bu bölgenin ağırlıklandırılarak yöneliminin tanımlanması olarak gerçekleştirilir.

Her bir anahtar nokta, gradyan büyüklüğüne ve gradyan yönelimine sahiptir. Gradyan büyüklüğü, normalize edilmiş dikey ve yatay türevlerinin karelerinin toplamının karekökü alınarak (2.11)'deki gibi bulunur.

$$m(x, y) = \sqrt{(L(x, y + 1) - L(x, y - 1))^2 + (L(x + 1, y) - L(x - 1, y))^2} \quad (2.11)$$

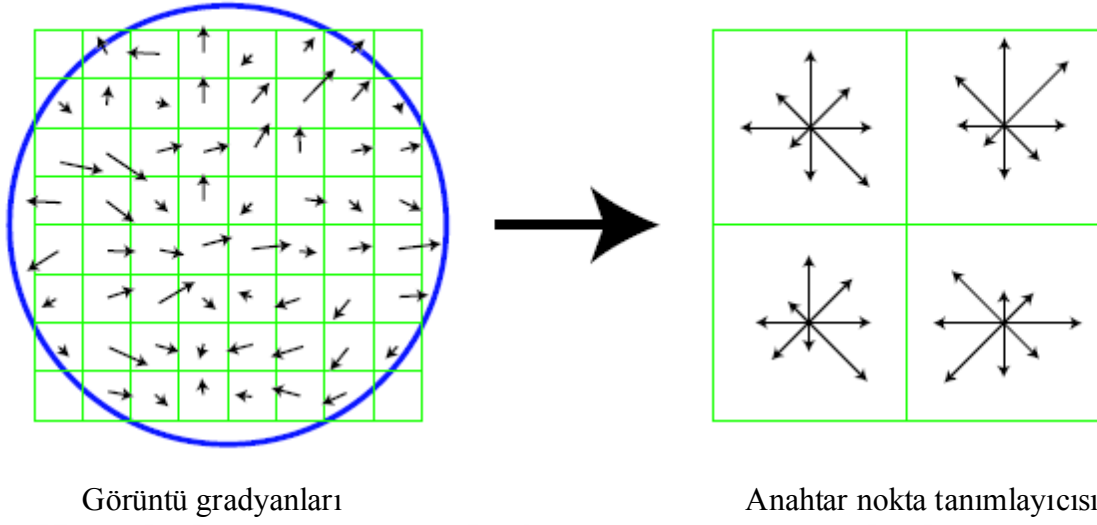
Gradyan yönelimi ise (2.12)'de verildiği gibi yatayda alınan türevin dikeyde alınan türeve oranı ile bulunur ve bütün noktaların açısı belirlenmiş olur.

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (2.12)$$

Anahtar noktanın komşu bölgesinde yer alan bütün pikseller için  $m$  ve  $\theta$  hesaplanır. Daha sonra bu değerler için histogram oluşturularak bütün ölçek bağımsız hale getirilir.

### 2.1.1.4. Anahtar Nokta Tanımlayıcılarının Oluşturulması

Anahtar noktalar bulunup gradyan yönelimleri belirlendikten sonra ölçek, dönme ve konumdan bağımsız özniteliklerin elde edilmesi sağlanmıştır. Bu işlemlerden sonra, Şekil 2.7'de gösterildiği gibi öznitelik vektörlerini temsil eden anahtar nokta tanımlayıcıları oluşturulur. Her bir anahtar nokta için bir tanımlayıcı elde edilerek ışık ve bakış açısı gibi etkenlere karşı bağımsız öznitelikler elde edilir.



**Şekil 2.7.** Görüntü gradyanları ve anahtar nokta tanımlayıcısının elde edilmesi

İlk olarak anahtar nokta etrafında  $16 \times 16$  boyutunda bölgeler oluşturulur ve bu bölgeler  $4 \times 4$ 'lük parçalara ayrılır. Her bir parça içerisine 8 farklı açı aralığına sahip histogramlar oluşturularak  $4 \times 4 \times 8 = 128$  elemanlı tanımlayıcı elde edilir. Ayrıca, elde edilen bu anahtar nokta tanımlayıcıları L-2 normuna göre normalize işlemine tabi tutularak aydınlanma değişimlerinin görüntü üzerindeki etkisi azaltılır.

### 2.1.2. Hızlandırılmış Gürbüz Öznitelikler Algoritması

Bay tarafından görüntüdeki öznitelikleri çıkarmak için 2008 yılında kısaca SURF olarak isimlendirilen ve Hessian matrisinin hesaplanmasına dayalı hızlandırılmış gürbüz öznitelikler algoritması önerilmiştir. Hessian'ın hesaplanmasında integral görüntüler kullanıldığı için hesaplama süresi düşmekte ve bu da algoritmanın SIFT algoritmasına göre daha hızlı çalışmasını sağlamaktadır [47].

SURF algoritması, aşağıdaki gibi temel olarak 2 adımdan oluşmaktadır:

- 1) Anahtar noktaların yerinin saptanması,
- 2) Anahtar nokta tanımlayıcılarının oluşturulması.

### 2.1.2.1. Anahtar Noktaların Yerinin Saptanması

SURF, küçük yuvarlak algılayıcısı Hessian matrisinin determinantını temel almaktadır. Hessian matrisinin determinantının maksimum olduğu konum, küçük yuvarlak türü yapıları algılamak için kullanılarak anahtar noktalar bulunur. Verilen bir I görüntüsü için Hessian matrisi (2.13)'teki gibi hesaplanır.

$$H(I(x,y)) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (2.13)$$

Bu Hessian matrisinin determinantı ise (2.14) ile hesaplanır.

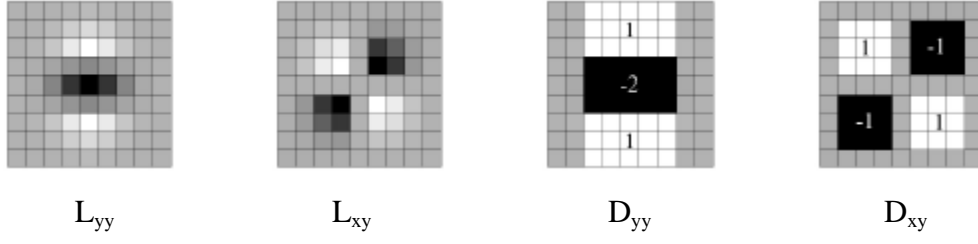
$$\text{Det}(H(I(x,y))) = \frac{\partial^2 I}{\partial x^2} \frac{\partial^2 I}{\partial y^2} - \left( \frac{\partial^2 I}{\partial x \partial y} \right)^2 \quad (2.14)$$

I görüntüsünün  $\sigma$  ölçeğinde olan bir  $x=(x,y)$  noktası için Hessian matrisi (2.15)'te gösterildiği gibi elde edilir [47].

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.15)$$

$L_{xx}(x, \sigma)$ , ikinci dereceden kısmi Gauss türevinin I görüntüsünde yer alan  $x=(x,y)$  noktasındaki evriminin sonucudur.  $L_{yy}(x, \sigma)$  ve  $L_{xy}(x, \sigma)$  de aynı şekilde hesaplanarak sonuçlandırılır. LoG olarak bilinen bu türevler SIFT algoritmasında DoG olarak yaklaşık olarak hesaplanırken SURF algoritmasında Şekil 2.8'de gösterilen  $\sigma = 1.2$  ölçeğinde ikinci dereceden Gauss türevlerine karşılık gelen  $9 \times 9$  boyutundaki kutu filtreleri kullanılarak çözüme yaklaşılr [47].





**Şekil 2.8.** İkinci dereceden Gauss türevlerine karşılık gelen kutu filtreleri

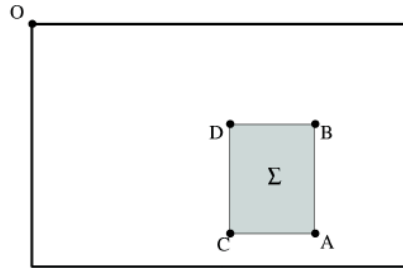
(2.16)'da Hessian matrisinin yaklaşık determinantı verilmiştir. Determinantı dengelemek için kullanılan  $w$  bağıl ağırlık yaklaşık 0.9 kabul edilmektedir [47].

$$\text{Det}(H) \cong D_{xx} \times D_{yy} - (wD_{xy})^2 \quad (2.16)$$

$$\text{Det}(H) \cong D_{xx} \times D_{yy} - (0.9D_{xy})^2 \quad (2.17)$$

Şekil 2.8'deki kutu filtreleri integral görüntü kullanılarak elde edilir. İntegral görüntüler, Şekil 2.9'daki gibi verilen dikdörtgen bir alanın hesaplanmasını hızlandırır. İntegral görüntü,  $I$  giriş görüntüsü için  $x = (x, y)^T$  koordinatlarının içinde kalan alan (2.18) ile hesaplanır [47].

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (2.18)$$

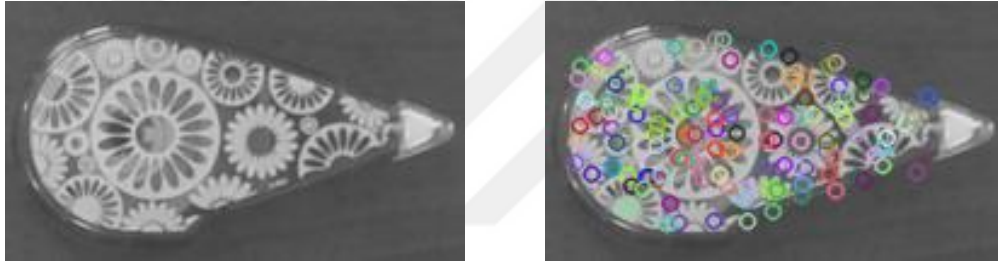


**Şekil 2.9.** İntegral görüntü

İntegral görüntü (2.18)'deki gibi bir kez hesaplandıktan sonra Şekil 2.9'daki dikdörtgenin alanı içindeki herhangi bir piksel için parlaklıkların (yoğunlukların) toplamı basit işlemler ile (2.19)'daki gibi yapılır [47].

$$\Sigma = A - B - C + D \quad (2.19)$$

Anahtar noktalar, kutu filtrelerden elde edilen ölçeklerden yararlanarak bulunmaktadır. Farklı ölçek değerlerinde farklı sayıda anahtar nokta elde edilir. Ölçek değeri arttıkça bulunan anahtar nokta sayısı düşmektedir. Ardışık olarak 3 ölçekten  $3 \times 3$ 'lük alanlar seçilerek toplam 27 piksel arasında en yüksek gradyan değerine sahip olan piksel, anahtar noktadır [47]. Şekil 2.10'da tez çalışmasında kullanılan bir nesne görüntüsünün SURF algoritması ile bulunan anahtar noktalarının gösterimine yer verilmiştir.



Şekil 2.10. Örnek nesne görüntüsü ve SURF anahtar noktalarının gösterimi

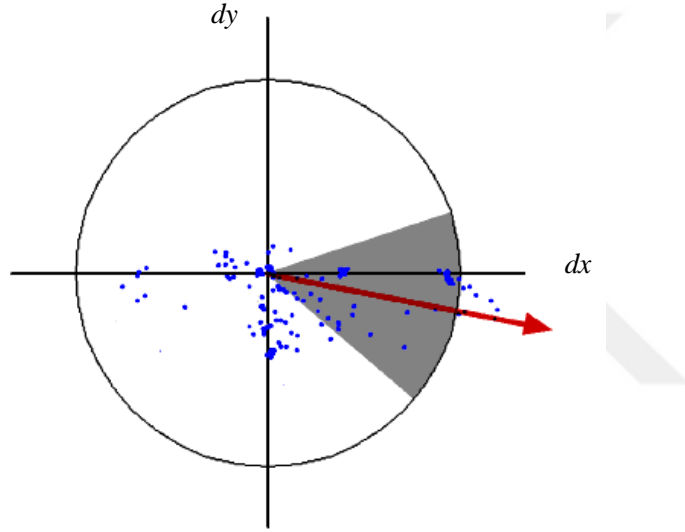
### 2.1.2.2. Anahtar Nokta Tanımlayıcılarının Oluşturulması

Anahtar noktalar elde edildikten sonra anahtar noktalar etrafında çember şeklinde bir alan seçilir ve bu alana Haar dalgacık filtreleri uygulanarak her bir anahtar nokta için bir tanımlayıcı elde edilerek öznitelikler çıkarılmış olur [48]. Bu işlem için öncelikle  $(x, y)$  koordinatında bulunan anahtar nokta etrafında daha önce kullanılan ölçek büyüklüğü cinsinden  $20\sigma$  büyüklüğünde bir alan belirlenir. Daha sonra bu alan büyüklüğü  $5\sigma$  olan  $4 \times 4$ 'lük alt kare alanlara bölünür.  $4 \times 4$ 'lük bu alanlara Şekil 2.11'de gösterilen Haar dalgacık filtreleri yatay ve dikey olarak uygulanıp x yönündeki türevi  $dx$  ve y yönündeki türevi  $dy$  hesaplanarak Haar dalgacık yanıtları elde edilir. Dalgacık yanıtları  $2\sigma$  ile ağırlıklandırılır ve yanıtların baskın yönelimi, Şekil 2.12'de gösterildiği gibi  $60^\circ$ 'lik kayan yönelim pencereleri ile hesaplanarak yönelim ataması gerçekleştirilir. Böylece görüntülerdeki anahtar nokta tanımlayıcıları eşleştirilirken dönmeye ve aydınlanmaya karşı

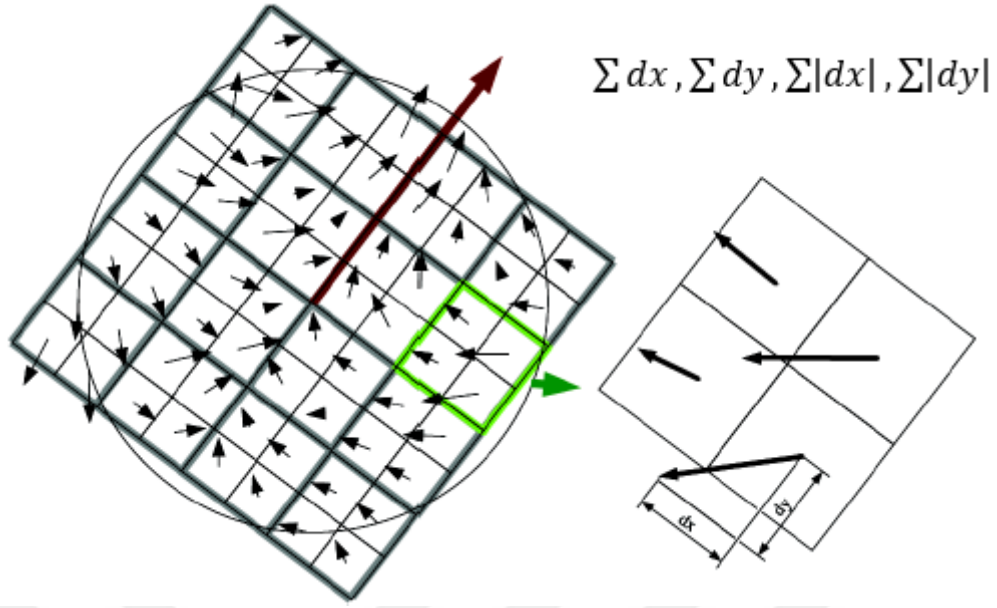
duyarsız hale gelir. Şekil 2.13'te verildiği gibi  $4 \times 4$ 'lük alandaki her bir alana Haar dalgacık filtresinden elde edilen  $\sum dx, \sum dy, \sum |dx|, \sum |dy|$  bileşen değerleri bulunup toplamda  $4 \times 4 \times 4 = 64$  elemanlı tanımlayıcı elde edilir [47].



Şekil 2.11. Haar dalgacık filtreleri



Şekil 2.12. Yönelim atama



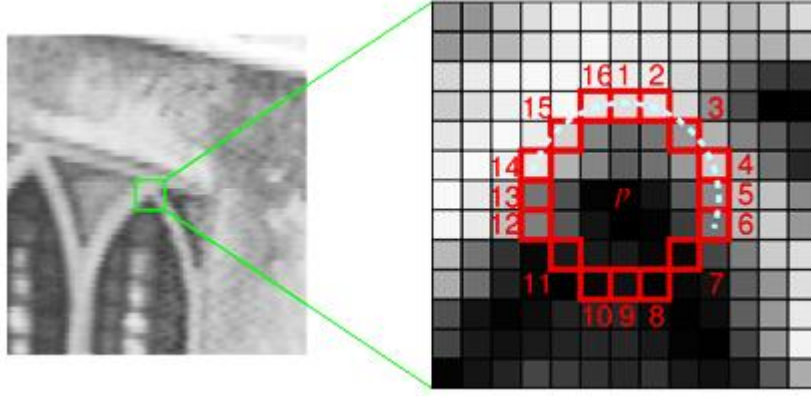
Şekil 2.13. Anahtar nokta tanımlayıcılarının oluşturulması

### 2.1.3. Hızlandırılmış Segment Testinden Öznitelikler Algoritması

Rosten ve Drummond tarafından görüntüdeki anahtar noktaları tespit etmek için 2006 yılında kısaca FAST olarak isimlendirilen hızlandırılmış segment testinden öznitelikler algoritması önerilmiştir [49]. FAST, öznitelikleri elde etmek için gerekli olan anahtar noktaların tespitini köşe noktalarını bularak gerçekleştirmekte olup yüksek hız ve güvenilirliğe sahiptir [45]. Ayrıca, bu algoritma sadece görüntüdeki anahtar noktaların tespitini yapabildiği için SIFT, SURF ve BRIEF gibi anahtar nokta tanımlayıcıları ile birlikte kullanılarak öznitelik çıkarımı da gerçekleştirilebilmektedir.

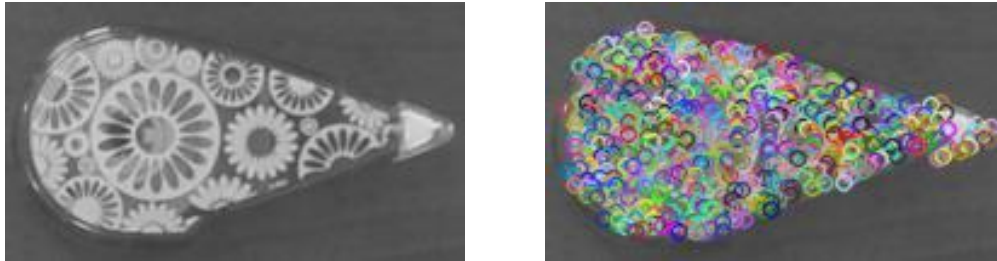
FAST algoritması, bir aday noktanın köşeyi belirtip belirtmediğini bulmak için yarıçapı 3 olan 16 piksellik Bresenham isimli bir çember kullanır. Şekil 2.14'te örnek olarak verildiği gibi görüntüde bir  $p$  aday noktası seçilip 16 piksellik çemberin her bir pikseli saat yönünde olacak şekilde 1'den başlayarak 16'ya kadar numaralandırılır.  $p$  noktasının piksel parlaklığı  $I_p$  ve eşik parlaklık değeri  $T$  (genellikle piksel parlaklık değerinin %20'si olacak şekilde) belirlendikten sonra algoritmanın hızlı çalışabilmesi için  $I_p$ , ilk olarak çember üzerinde yer alan 1, 5, 9 ve 13 numaralı piksellerin parlaklık değerleri ile karşılaştırılır. Bu dört pikselden en az üç tanesinin  $I_i > I_p + T$  veya  $I_i < I_p - T$  eşik değer kriterini sağlaması durumu,  $p$  noktasının köşe belirten anahtar nokta olabileceği ipucunu verir. Daha sonra çember üzerindeki 16 piksel arasından en az 12 bitişik (komşu) pikselin eşik değer kriterini

sağlaması durumunda da  $p$  noktasının anahtar nokta olduğu tespit edilir. Eğer  $I_1, I_5, I_9$  ve  $I_{13}$ 'ten en az üçü bu eşik değer kriterini sağlamazsa  $p$  noktası köşe belirtmeyen bir noktadır ve anahtar nokta sayılmaz [50]. Görüntüdeki tüm pikseller için aynı işlemler yapılarak anahtar noktaların tespiti yapılır.



Şekil 2.14. Anahtar nokta tespit işlemi

Şekil 2.15'te bu tez çalışmasında kullanılan bir nesne görüntüsünün FAST algoritması ile bulunan anahtar noktalarının gösterimine yer verilmiştir.



Şekil 2.15. Örnek nesne görüntüsü ve FAST anahtar noktalarının gösterimi

#### 2.1.4. İkili Gürbüz Bağımsız Temel Öznitelikler Algoritması

Calonder ve arkadaşları tarafından görüntüdeki anahtar noktaların tanımlayıcılarını hızlı bir şekilde hesaplayarak çıkarmak için 2010 yılında kısaca BRIEF olarak isimlendirilen ikili gürbüz bağımsız temel öznitelikler algoritması önerilmiştir [51]. Bu algoritma ile çıkarılan tanımlayıcılar, aydınlanma değişimleri ve görüntü bulanıklıklarına karşı dirençli olup son derece ayırt edici özelliklere sahiptir.

BRIEF algoritması, sadece anahtar noktaların tanımlayıcılarını çıkaran bir algoritma olduğu için anahtar noktaları bulma işleminde SIFT, SURF ve FAST gibi anahtar noktaları tespit edebilen algoritmalar ile birlikte kullanılmak şartıyla anahtar nokta tanımlayıcılarını elde edebilmektedir.

BRIEF algoritmasında ilk olarak  $S \times S$  boyutunda bir parça seçilir ve bu parçaya bir önişlem uygulanarak parça içerisinde yer alan pikseller yumuşatılır. Yumuşatma işleminden sonra pikseller birbirleriyle karşılaştırılır. Örneğin; bir  $p$  parçası seçip üzerinde  $\tau$  test tanımlayalım. Bunun için parçadan  $x$  ve  $y$  olarak temsil edilen iki piksel aldığımızı düşünelim.  $\tau$ ,  $x$  pikselinin parlaklık değeri  $p(x)$  ile  $y$  pikselinin parlaklık değeri  $p(y)$ , (2.20)'deki şartlar ile karşılaştırılıp 0 veya 1 ikili (binary) dizi çıktıları elde edilerek gerçekleştirilir [51].

$$\tau(p; x, y) := \begin{cases} 1, & \text{eğer } p(x) < p(y) \\ 0, & \text{değilse} \end{cases} \quad (2.20)$$

BRIEF algoritmasında elde edilen tanımlayıcılar 32 elemanlı olup, (2.21)'de gösterildiği gibi  $n_d$  boyutlu bit dizisi olarak ifade edilmektedir [51].

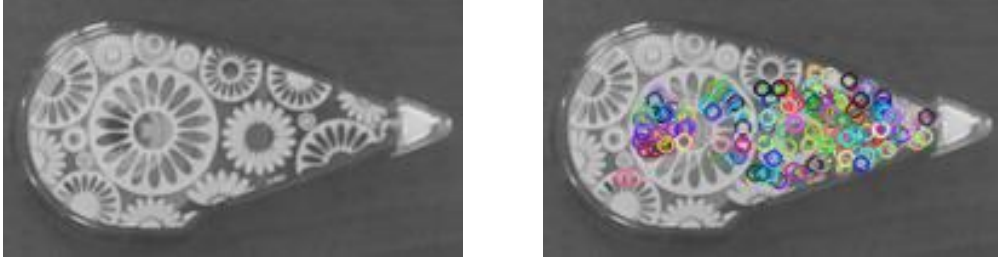
$$f_{n_d}(p) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p; x_i, y_i) \quad (2.21)$$

### 2.1.5. Yönlendirilmiş FAST ve Döndürülmüş BRIEF Algoritması

Rublee ve arkadaşları tarafından görüntüdeki özniteliklerin çıkarımında SIFT veya SURF algoritmalarına karşı hızlı ve etkili bir seçenek olabilmek için 2011 yılında kısaca ORB olarak isimlendirilen yönlendirilmiş FAST ve döndürülmüş BRIEF algoritması önerilmiştir [41]. Anahtar noktaların tespiti için FAST, anahtar noktaların tanımlayıcılarını çıkarmak için ise son yıllarda geliştirilen BRIEF algoritması kullanılmıştır. Bu algoritma ile çıkarılan öznitelikler, hem dönmeye ve gürültüye karşı duyarsız hem de aydınlanma değişikliklerine karşı dirençlidir.

ORB algoritmasında öncelikle FAST algoritması ile farklı ölçeklerde anahtar noktalar elde edilir. Harris köşe bulucu yöntemi kullanılarak anahtar noktalar sıralanır. Belli bir eşik değere bağlı olarak en belirgin köşeleri ifade eden anahtar noktalar seçilir.

Şekil 2.16’da bu tez çalışmasında kullanılan bir nesne görüntüsünün ORB algoritması ile bulunan anahtar noktalarının gösterimine yer verilmiştir.



**Şekil 2.16.** Örnek nesne görüntüsü ve ORB anahtar noktalarının gösterimi

Daha sonra seçilen bu anahtar noktaları dönmeye karşı bağımsız hale getirmek için basit bir köşe yönelim ölçüsü olan parlaklık kütle merkezi yöntemi [52] kullanılır. İlk olarak görüntüdeki bir parçada bulunan köşeleri belirten anahtar noktaların momentleri (2.22)’deki gibi hesaplanır.

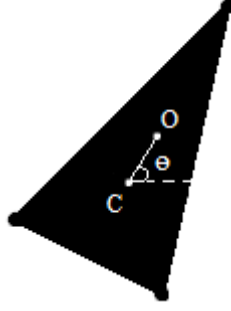
$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (2.22)$$

(2.22) kullanılarak bulunan momentler ile parçanın kütle merkezi (2.23)’te verildiği gibi bulunur.

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.23)$$

Köşe merkezi O noktasından, kütle merkezi C noktasına bir vektör oluşturularak parçanın yönelim açısı  $\theta$ , (2.24)’teki gibi hesaplanıp, örnek gösterimi Şekil 2.17’de verilmiştir.

$$\theta = \tan^{-1}(m_{01}, m_{10}) \quad (2.24)$$



Şekil 2.17. Yönelim açısı hesaplama gösterimi

$\theta$  hesaplandıktan sonra kanonik olarak döndürülen parça üzerinde BRIEF tanımlayıcıları hesaplanır ve bit dizisi şeklinde ORB tanımlayıcıları oluşturulur.

## 2.2. Öznitelik Eşleştirme ve Sınıflandırma

Eğitim görüntülerinin öznitelikleri çıkarılıp veri kümesinde saklandıktan sonra verilen bir test görüntüsündeki bilinmeyen bir nesneyi sınıflandırabilmek için ilk olarak nesne veri kümesinde yer alan öznitelikler, test görüntüsünden çıkarılan özniteliklerin her biri ile ayrı ayrı eşleştirilir. Bu eşleştirmeler sonucunda nesne sınıfı hakkında bilgiler elde edilir.

Veri kümesindeki öznitelikler ile test görüntüsünden çıkarılan özniteliklerin eşleştirilerek benzerliğini belirlemek için genellikle k-En yakın Komşuluk (k-EK) algoritması kullanılmaktadır. Bu algorithmada test verileri en yakın k komşularının sınıf etiketlerine bakılarak sınıflandırılır. Bu işlem, verilen bir noktaya en yakın k komşu sayılarının belirlenip bu nokta ile diğer tüm noktalar arasındaki uzaklıkların tek tek hesaplanması ile gerçekleştirilir.

Tanımlayıcılar arasındaki benzerlik ölçümü veri tipine bağlıdır. Gerçek değerli veriler için Öklid uzaklığı kullanılabilirken ikili veri tipleri için Hamming uzaklığı tercih edilmektedir. Uzaklıkların hesaplanmasında genellikle Öklid metriği kullanılmaktadır. p ve q tanımlayıcılar, n ise tanımlayıcıyı ifade eden vektörün eleman sayısı olmak üzere Öklid metriği (2.25) ile hesaplanmaktadır.

$$\text{Uzaklık}(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.25)$$



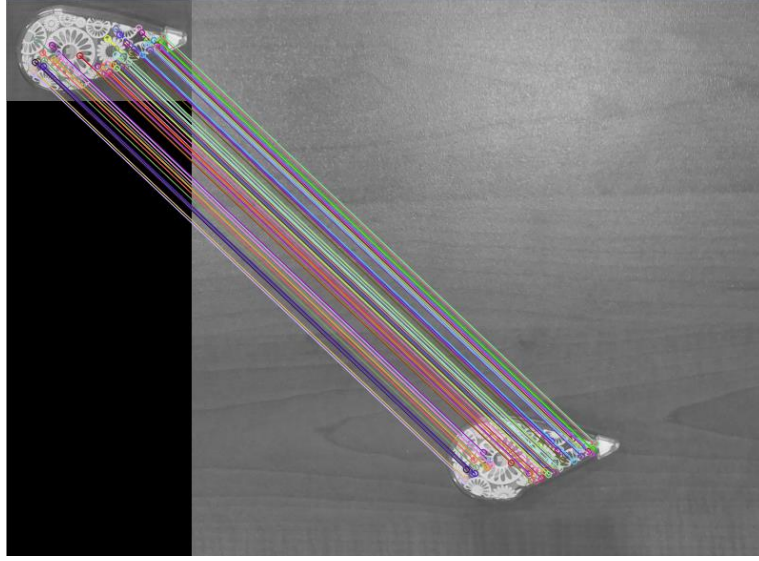
(2.26)'daki Hamming metriği ile de ikili veri tipindeki tanımlayıcılar arasındaki uzaklıklar hesaplanabilmektedir.

$$\text{Uzaklık}(p, q) = \sqrt{\sum_{i=1}^n \delta_i} , \delta_i = \begin{cases} 1, & \text{eğer } p_i \neq q_i \\ 0, & \text{değilse} \end{cases} \quad (2.26)$$

En yakın komşuluk için gerçekleştirilen doğrusal arama yani noktalar arasındaki uzaklıkların tek tek hesaplanması işleminin hesaplama maliyeti yüksektir. Ayrıca, en iyi sonucun alınabilmesi için hangi metriğin kullanılması gerektiği de açık değildir. Bu yüzden, çoğu algoritma yaklaşık en yakın komşu (approximate nearest neighbor) arama üzerine yoğunlaşmaktadır. Muja ve Lowe tarafından önerilip C++ dilinde yazılan kısaca FLANN [53] (Fast Library for Approximate Nearest Neighbors) olarak isimlendirilen yaklaşık en yakın komşuluk için hızlı kütüphane algoritması en yakın komşu aramaya dayanmakta olup “hierarchical k-means tree” [54] ve “multiple randomized kd-trees” [55] gibi derleme algoritmalarından oluşmaktadır. Bu kütüphane tabanlı eşleştirici, tanımlayıcıların eşleşmesi için uygun olan en iyi algoritmayı seçerek hızlı bir şekilde uygulamaktadır.

Bir başka tanımlayıcı eşleştirici algoritma olan BF (Brute-Force–Kaba Kuvvet) ise, Öklid veya Hamming metriklerinden herhangi biriyle birlikte kullanılarak veri kümesi ile test görüntüsünden elde edilen tanımlayıcılar arasındaki tüm eşleşme kombinasyonlarını gerçekleştirerek birbirine en yakın uzaklıkta olan tanımlayıcıları bulmaktadır.

Bu tez çalışmasında kullanılan örnek bir nesne ve test görüntüsünün FAST algoritması ile elde edilen anahtar noktaların Hamming uzaklığı birbirine en yakın olan BRIEF tanımlayıcılarının eşleştirilme örneği Şekil 2.18’de verilmiştir.



**Şekil 2.18.** FAST anahtar noktaların Hamming uzaklığı birbirine en yakın olan BRIEF tanımlayıcılarının eşleştirilmesi

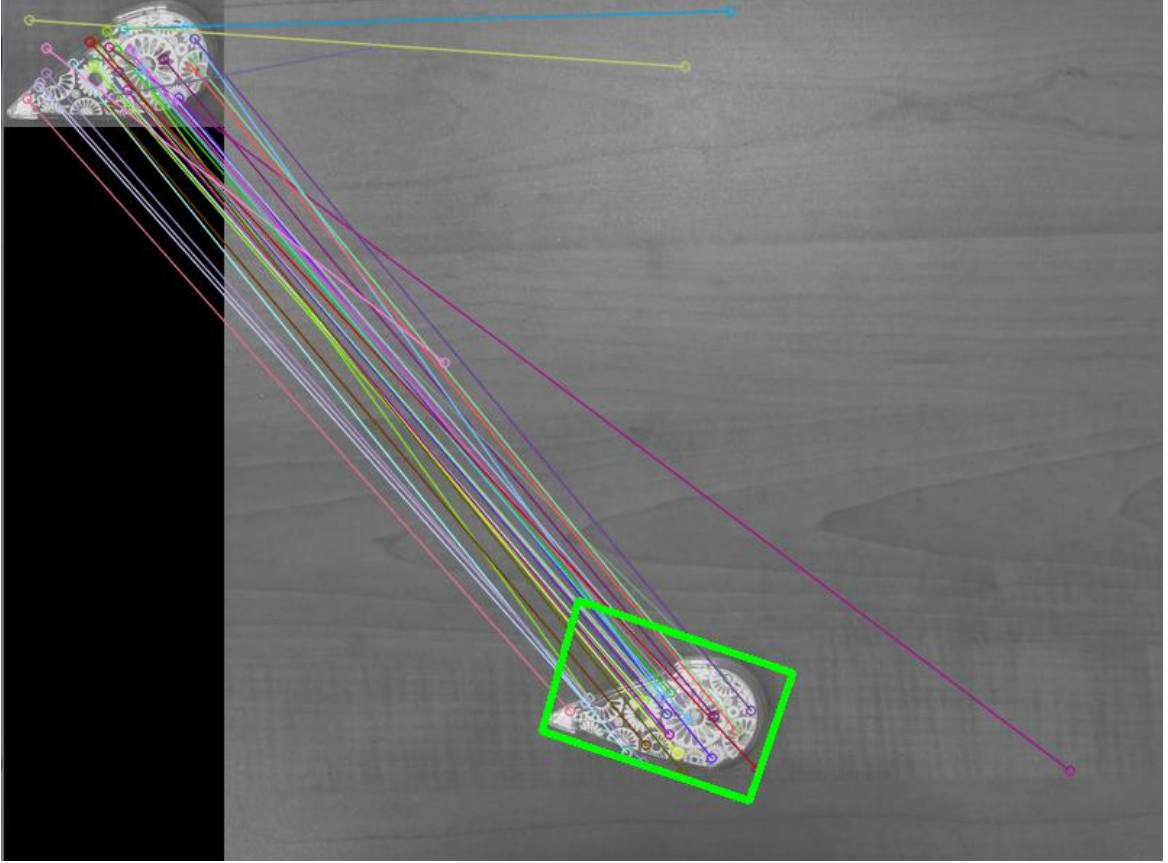
Öznitelikleri temsil eden anahtar nokta tanımlayıcıları eşleştirilip aralarındaki uzaklıklar hesaplandıktan sonra birbirine en yakın olan eşleşmeler belirlenir. En yakın eşleşmelerin en fazla olduğu nesne görüntüsünün bulunduğu sınıf test görüntüsündeki nesnenin ait olduğu nesne sınıfını belirler.

### 2.3. Homografi Tahmini

Homografi, bir düzlemdeki noktaları diğeri ile eşleştiren iki boyutlu bir izdüşümsel dönüşümdür [56]. Görüntülerden çıkarılan anahtar nokta tanımlayıcıları benzer noktaları elde etmek için eşleştirildiklerinde bazı aykırı eşleşmeler gerçekleşebilmektedir. Elde edilen bu benzer nokta seti arasındaki aykırı eşleşmeleri eleyip en doğru eşleşenleri ortaya çıkarabildiği için tez çalışmasında kısaca RANSAC (RANdom SAmple Consensus) olarak isimlendirilen rastgele örnek uzlaşımı algoritması kullanılmıştır.

Fischler ve Bolles tarafından önerilen RANSAC algoritması [57], giriş verisindeki aykırı noktaların büyük bir kısmının üstesinden gelmek için tasarlanmış genel bir parametre tahmin yaklaşımıdır [58]. Bu algoritma, eşleşen nokta çiftlerinden elde ettiği dönüşüm matrisini kullanarak noktaların üst üste gelip gelmediğinin kontrolünü gerçekleştirir. İşlemlere bu şekilde tekrar ederek en fazla doğruluk sonucunu veren dönüşüm matrisini hesapladıktan sonra aykırı eşleşmeleri tespit edip eleyerek en uygun nokta çiftlerini bulur.

RANSAC algoritması sayesinde homografi tahmini gerçekleştirilip en doğru eşleşen tanımlayıcılar elde edilerek nesnenin yeri doğru bir şekilde tespit edilmiş olur. Şekil 2.19'da örnek bir nesne görüntüsü ile test görüntüsü arasındaki eşleşmeler sonucunda uygun nokta çiftleri sınırlayıcı bir kutu içine alınarak aykırı eşleşmelerin elenmesi gösterilmiştir.



Şekil 2.19. RANSAC algoritması ile aykırı eşleşmelerin tespit edilerek elenmesi

### 3. ROBOTİK VE ROBOT MANİPÜLATÖRLER

Robotik; mekatronik mühendisliğinin çalışma alanlarından biridir. Bu alan, günlük yaşantımızda ya da endüstriyel alanda insanların zorlanabilecekleri, yaparken sıkılabilecekleri veya güvenliklerini tehlikeye düşürebilecek işlerin yapılmasında tercih edilir.

Gittikçe gelişen robotik teknoloji, robot adını verdiğimiz teknolojik ürünler üzerinde bütünleşmeyi ve robot uygulamalarını içerir. Robotlar; otonom olarak ya da önceden programlanarak taşıma veya yer değiştirme gibi işlemleri gerçekleştirebilen elektro-mekanik sistemlerdir.

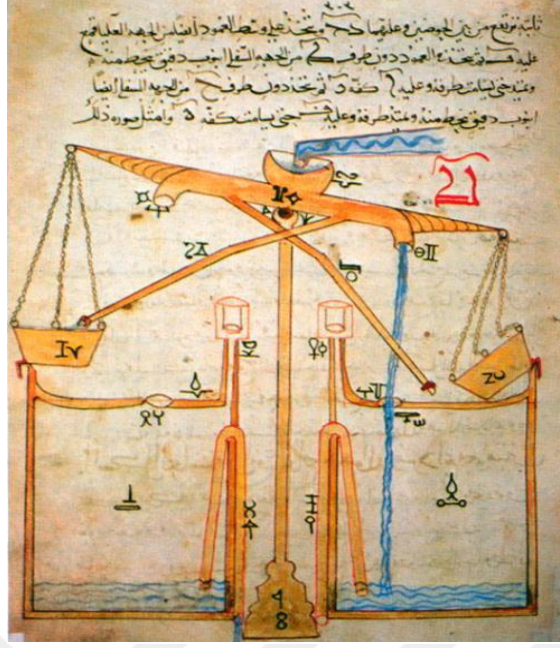
Robot manipülasyonu; robot gövdesinin pozisyonunda meydana gelen değişikliktir. Robot manipülatör ise; bir küme eklem birbiriyle bağlandığı bir küme rijit kol içerir [59]. Yani robot manipülatör, robotun kinematik zincirini oluşturan ve sahip olduğu eksenleri vasıtasıyla hareket etmeyi sağlayan birbirine bağlı çok sayıda hareketli eleman ile beraber mekanik ve elektronik yapının da içinde bulunduğu robot kolunu ifade etmektedir. Robot kol eklemlerine motorlar takılarak robotun verilen bir görevi gerçekleştirmesinin kontrolü sağlanır.

#### 3.1. Robotların Tarihsel Gelişimi

Robot kavramı ilk olarak 1920'li yıllardaki Çek yazar Karel Capek tarafından yazılan "Rossum's Universal Robots" adlı tiyatro oyununda kullanılmıştır. Bu tiyatro oyunundaki bir tiplenenin adı olan "robot", insani duygulardan yoksun mekanik ve otonom varlıklar olarak tanımlanmıştır.

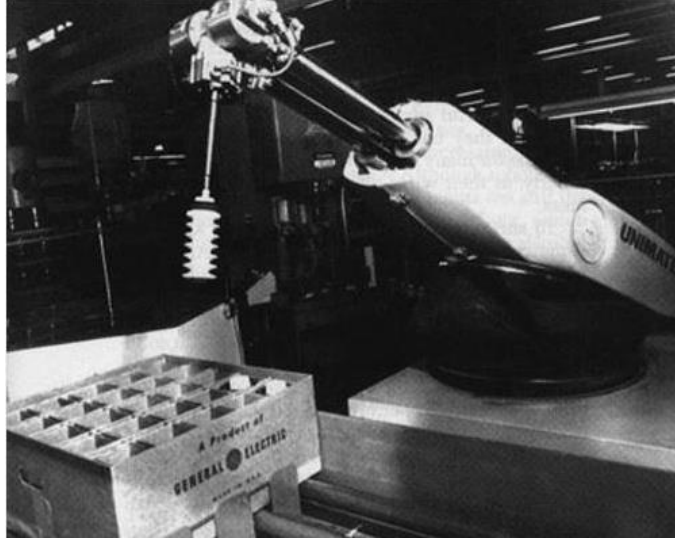
Aslında robot olarak tanımlanabilecek makineler için bilgiler eski Yunan eserlerine göre M.Ö. 3000 yıllarına kadar uzanmaktadır. M.Ö. 100 yıllarında yaşayan İskenderiyeli bir mühendisin de otomatik açılan kapı gibi düzenekleri su ve buhar gücü ile çalıştırdığı bilinmektedir. İlerleyen zamanda, robot bilimi hakkında bilimsel çalışmalar yoğunlaşmıştır.

12. yy'de yaşamış sibernetiğin temellerini attığı ve ilk robotu yapıp çalıştırdığı kabul edilen bilim adamı ve mühendis El Cezerî otomatik makineler geliştirme konusunda birçok çalışma yapmış ve robotik biliminin gelişimine büyük katkı sağlamıştır. El Cezerî'nin yapmış olduğu otomatik su makinesi Şekil 3.1'de verilmiştir.



Şekil 3.1. Otomatik çalışan su makinesi

Teknolojik olarak günümüz teknolojisine en yakın robot 1960'lı yıllarda Şekil 3.2'de verilen George Charles Devol ve Joseph Engelberger tarafından geliştirilen ilk endüstriyel robotlardan olan "Unimate Robot" 'tur. "Unimate Robot", bir manipülatörle bir bilgisayarın uyumlu çalışabilmesi sonucu birçok görevi otomatik olarak yerine getirebilmektedir. Bu ilk bilgisayar kontrollü mekanik kol, 1940'lı yıllarda bir insan tarafından kontrol edilerek radyoaktif malzemeleri taşımak için kullanılmıştır. Daha sonra, kontrolü insandan alınıp bilgisayara devredilmiştir. 1967 yılında Japonya ilk endüstriyel robotunu ithal etmiştir. 1976 yılında NASA tarafından Mars'a gidecek olan uzay araçlarına robot kollar yerleştirilmiştir. Robot teknolojisi, ilerleyen zamanlarda elektronik biliminin desteğiyle daha da ilerlemiştir.



Şekil 3.2. İlk endüstriyel robot olan Unimate Robot

### 3.2. Endüstriyel Robotik ve Robot Kollar

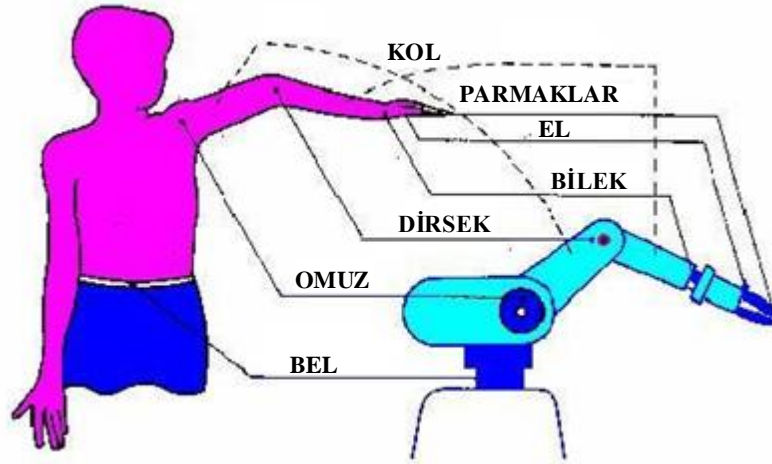
Endüstride en çok kullanılan makineler robotlardır. Günümüzde endüstriyel robotlar çeşitli boyutlarda üretilebilmesi ve bir insanın yapabileceğinden daha büyük işler yapabilmesinden dolayı sanayide vazgeçilmez hale gelmiştir. Bu robotları, fabrikada çalışan işçilerden ayıran, programlanabilmesi, kusursuz olarak ve yorulmaksızın çalışabilmesidir.

Endüstriyel robotlar genellikle zemine monte edilmiş şekilde olup sabit konumludur. Robot kollar olarak da bilinen endüstriyel robotlar, ana gövdeden ve gövdeye bağlı kollardan meydana gelmektedir. Dünya çapındaki endüstriyel robot üreticisi olan KUKA firmasının Şekil 3.3'teki robot kolu endüstriyel robotlara örnek olarak gösterilebilir.



Şekil 3.3. KUKA endüstriyel robot

Robot kollar, insan koluna benzer şekilde tasarlanmış mekanik yapılardır. İnsan koluna benzerliği; kavrama ve yerleştirme işlemlerini yapabilmesidir. Robot kolu ile insan kolunun benzerliği Şekil 3.4’te gösterilmiştir.



Şekil 3.4. İnsan kolu ile robot kolu benzetimi

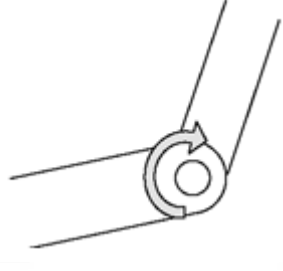

Robot kol, eklemlerden ve her bir eklemi hareket ettiren motorlardan meydana gelmektedir. Robot kolların kabiliyetleri, eklem tipleri ve serbestlik derecelerinin sayısına bağlı olarak değişmektedir.

### 3.2.1. Robot Kol Eklem Tipleri

Eklem, robot kolun çalışma alanının belirlenmesinde büyük rol oynamaktadır. Eklem sayesinde robot kola çok yönlü hareket yeteneği kazandırılmaktadır. Bu yüzden hareket yeteneğinin robotun çalışma alanının belirlenmesiyle doğrudan ilişkisi bulunmaktadır.

Robot kolların tasarımında genellikle döner ve prizmatik olmak üzere 2 temel eklem tipi kullanılmaktadır. Bu eklem tipleri, Tablo 3.1’de gösterilmiştir.

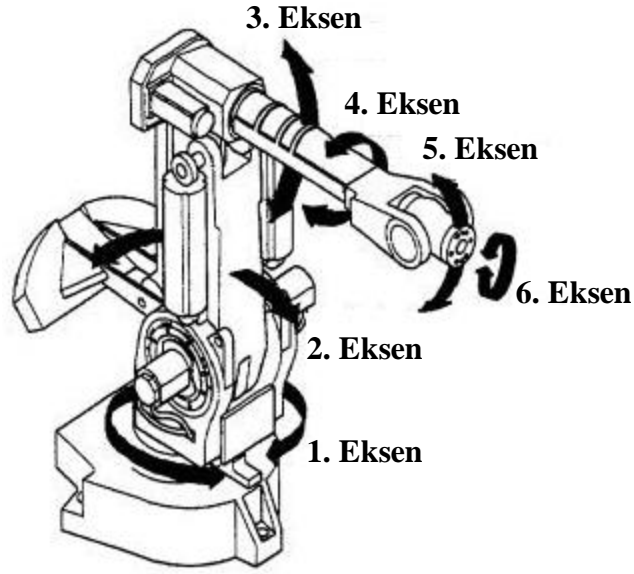
**Tablo 3.1.** Robot kol eklem tipleri

Eklem Tipi	Sembolik Gösterim		Eklem Tanımı
	Harfsel	Şekilsel	
Döner (Revolute)	R		Eksen etrafında dönme hareketi
Prizmatik (Prismatic)	P		Eksen boyunca doğrusal hareket

### 3.2.2. Serbestlik Derecesi

Robot kollara, bağımsız hareket özelliği kazandıran her eklem, bir serbestlik derecesi meydana getirir ve buna bağlı olarak bir robot ekseninin oluşmasını sağlar. Bu yüzden, robotik alanında “eksen” kavramı serbestlik derecesi olarak düşünülmektedir. Robot kolun serbestlik derecesi, yani eksen sayısı arttırıldığında daha fazla çalışma alanına sahip olunur. Şekil 3.5’te 6 eksenli bir robot kolun serbestlik dereceleri görülmektedir.





Şekil 3.5. 6 eksenli robot kolun serbestlik dereceleri

### 3.3. Robot Kolların Endüstrideki Kullanımları

Robot kollar, son yıllardaki tıp ve askeri uygulamaların dışında otomotiv endüstrisi başta olmak üzere endüstride de sıkça kullanılmaktadır. Robot kolların kullanıldığı bazı işlemler:

- Kesme
- Şekil verme
- Yüzey taşlama
- Paketleme
- Toplama
- Boyama
- Kaynak

Yukarıda yer alan imalat işlemlerinin yanı sıra, endüstride taşıma ve montaj uygulamalarında da kullanılmaktadır. Robot kolların otomotiv endüstrisindeki örnek kullanımı Şekil 3.6'da gösterilmiştir.



**Şekil 3.6.** Otomotiv endüstrisinde kullanılan robot kol örneği

### 3.4. Robot Manipülatorlerin Eklem Tiplerine Göre Sınıflandırılması

Bir robot manipülatorün her eklemi sınırlı bir hareket aralığına sahiptir. Robot manipülatorlerin ilk üç eklemlerinin oluşturduğu ana eksenlerin eklem tiplerine göre robot manipülatorler, Tablo 3.2'deki gibi sınıflandırılmaktadır.

**Tablo 3.2.** Ana eksenlerin eklem tiplerine göre robot manipülator sınıfları

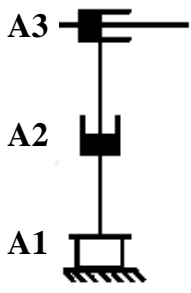
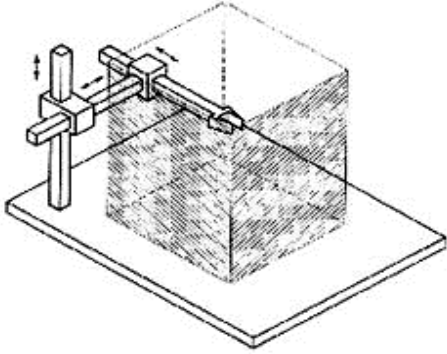

Robot Manipülator Sınıfı	Ana Eksenler			Ana Eksen Tipi
	Eksen 1	Eksen 2	Eksen 3	
Kartezyen	P	P	P	PPP
Silindirik	R	P	P	RPP
Küresel	R	R	P	RRP
SCARA	R	R	R	RRR-RRP (Yatay hareket)
Eklemlili (Mafsallı)	R	R	R	RRR (Dikey hareket)

### 3.4.1. Kartezyen Robot

İlk üç eklem tipi PPP olan kartezyen robotlar, en kısıtlı hareket yeteneğine sahip tasarımlarına rağmen mekanik yönden çok dayanıklıdırlar. Tasarımlarının basitliğinden dolayı da kol hareketleri kolaydır. Birbirine dik açılı kayan eklemleri sayesinde x, y ve z koordinatlarındaki hareketleri birbirine dik açıdır. Genellikle büyük boyutlu nesnelere taşımada kullanılmaktadır.

Tablo 3.3'te kartezyen robotun kinematik yapısı, çalışma alanı ve örneğine yer verilmiştir.

Tablo 3.3. Kartezyen robot

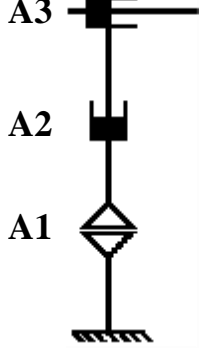
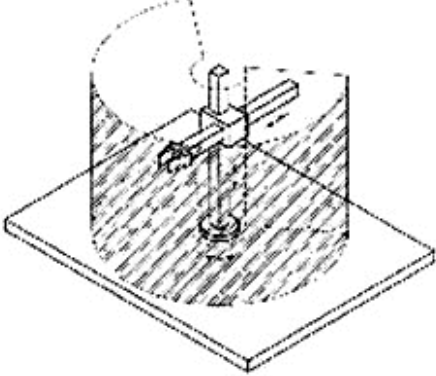

Kinematik Yapı	Çalışma Alanı	Robot Örneği
		

### 3.4.2. Silindirik Robot

İlk üç eklem tipi RPP olan silindirik robotlar, silindirik koordinatlarda hareket etmektedir. Kartezyen robotlara benzer olarak mekanik yönden dayanıklıdır ve büyük boyutlu nesnelere taşımada tercih edilmektedir.

Tablo 3.4'te silindirik robotun kinematik yapısı, çalışma alanı ve örneğine yer verilmiştir.

Tablo 3.4. Silindirik robot

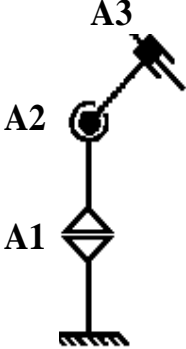
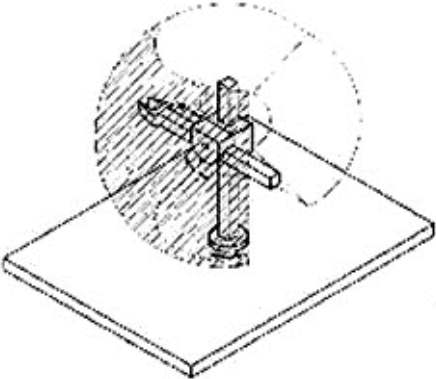
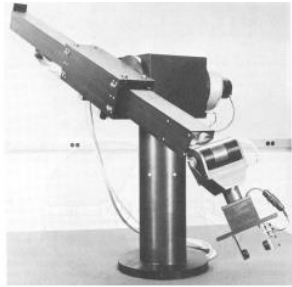
Kinematik Yapı	Çalışma Alanı	Robot Örneği
		

### 3.4.3. Küresel Robot

İlk üç eklem tipi RRP olan küresel robotlar, çalışma alanı içerisindeki hedef noktalara ulaşabilme bakımından yeteneklidir. Robot kolunun dikey hareketi, açısız hareketiyle tutucunun aşağıya ve yukarıya çıkabileceği kadardır. Mekanik açıdan silindirik ve kartezyen robotlara göre daha zayıftır. Genellikle makine montaj işlemlerinde kullanılmaktadırlar.

Tablo 3.5'te küresel robotun kinematik yapısı, çalışma alanı ve örneğine yer verilmiştir.

Tablo 3.5. Küresel robot

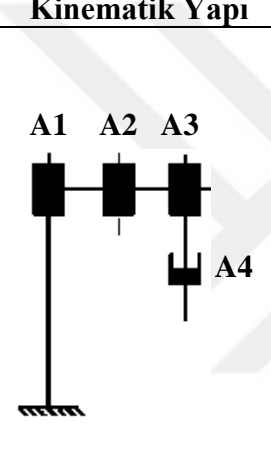
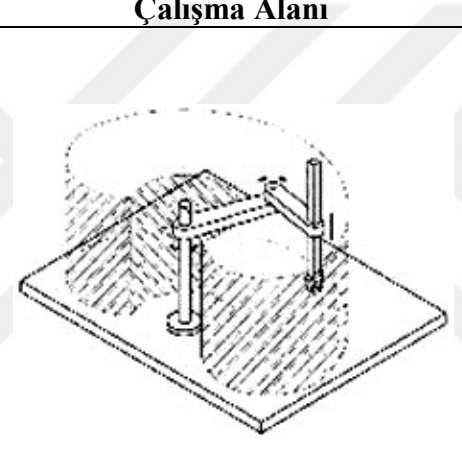

Kinematik Yapı	Çalışma Alanı	Robot Örneği
		

### 3.4.4. SCARA Robot

İlk üç eklem tipi RRR veya RRP olan kısaca SCARA (Selective Compliance Assembly Robot Arm) olarak isimlendirilen seçimlere uyan montaj robot kolunda, ilk iki eklem tipi olan döner eklemler yatay düzlemdeki hareketi, üçüncü eklem ise düşey yöndeki hareketi sağlar. Taşıma kapasiteleri düşük olmalarına rağmen yüksek hızları bakımından montaj işlemlerinde çok fazla tercih edilirler.

Tablo 3.6’da SCARA robotun kinematik yapısı, çalışma alanı ve örneğine yer verilmiştir.

Tablo 3.6. SCARA robot

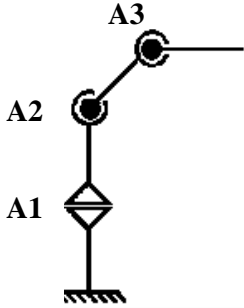
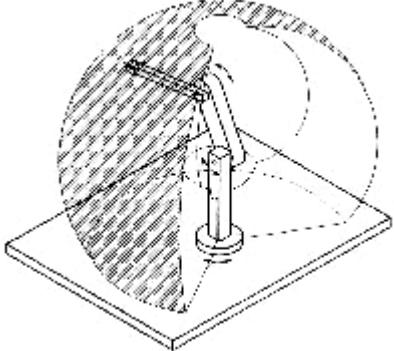

Kinematik Yapı	Çalışma Alanı	Robot Örneği
		

### 3.4.5. Eklemlili (Mafsallı) Robot

İlk üç eklem tipi RRR olan eklemlili robotlar, insan kolunun yapısına benzetilmektedir. Tüm eklemleri döner tipte olduğu için robot kol maksimum derecede hareket edebilmekte, bu da onu hareket yeteneği en fazla olan robot tiplerinden yapmaktadır. Genellikle, boyama ve kaynak işlemleri için kullanılmaktadırlar.

Tablo 3.7’de eklemlili robotun kinematik yapısı, çalışma alanı ve örneğine yer verilmiştir.

Tablo 3.7. Eklemlı (mafsallı) robot

Kinematik Yapı	Çalışma Alanı	Robot Örneđi
		



## 4. ROBOT KOL KİNEMATİK ANALİZİ

Robot kol kinematiği, bir robot kolun sabit bir referans koordinat sistemine göre hareketinden kaynaklanan kuvvet veya moment etkisini göze almaz. Bu yüzden kinematik analiz ile robotun uzaysal yer değiştirmesinin zamana bağlı bir fonksiyonunun analitik tanımı elde edilmektedir. Kinematik analiz ile elde edilen bu bilgiler, robot kolun bir cismi bir yerden alıp bir yere koyma işlemi bakımından hareketlerini belirtebilir. Robot sistemlerinde bu kinematik analizler büyük önem taşımaktadır. Robot kinematiği ile robotun hız, kuvvet ve ivme analizleri yapılmaktadır.

Bir robot, öteleme ve dönmeyi gerçekleştiren eklemlerden ve bu eklemleri birbirine birleştiren bağlardan oluşmaktadır. Her bir robot eklemine konumu, bir önceki veya bir sonraki eklem göre ifade edilmektedir. Arka arkaya oluşturulan bu ilişki, açık kinematik zincir olarak isimlendirilir [60]. Zincirin bir ucu zemine bağlı iken diğer ucu tutucu uca bağlıdır. Maxwell [61], 4x4 homojen dönüşüm matrislerini tanımlamıştır. Denavit-Hartenberg [62] ise bu dönüşüm matrislerini kullanarak herhangi bir koordinat sisteminin konum ve yönelim bilgilerini başka bir koordinat sisteminin durumuna göre belirtmiştir.

Kinematik analiz, tüm robotik sistemlerde olduğu gibi robot kol sistemlerinde de iki farklı şekilde ele alınmaktadır. Bunlardan birincisi, robot kol eklemlerinin istenilen açı değerlerini alması sonucunda tutucu ucun gideceği konumu belirleyen “düz kinematik”, diğeri ise, tutucu ucun istenilen konuma gitmesi için eklemlerin alması gereken açı değerlerini belirleyen “ters kinematik” analizdir.

Aşağıdaki bölümlerde yapılan düz ve ters kinematik analiz işlem adımları, tez çalışmasında eklemleri döner tipteki 4 serbestlik dereceli robot kol kullanıldığı için 4 serbestlik dereceli robot kol üzerinde gerçekleştirilmiştir.

### 4.1. Düz (İleri Yön) Kinematik Analiz

Düz kinematik analiz, robot kolun her bir eklemine belirlenen açılarda hareketi sonucunda robot uç noktası olan tutucu ucun ulaştığı konumun koordinatlarını elde etmek için yapılmaktadır.

Bilindiği üzere eklemlerle bir robot kol, birbirine prizmatik veya döner eklemlerle bağlanmış seri uzuvlardan oluşur. Robot kolun her bir uzvuna bir koordinat sistemi yerleştirilerek bu koordinat sistemleri arasındaki bağıl konum ve yönelim, homojen dönüşüm matrislerini

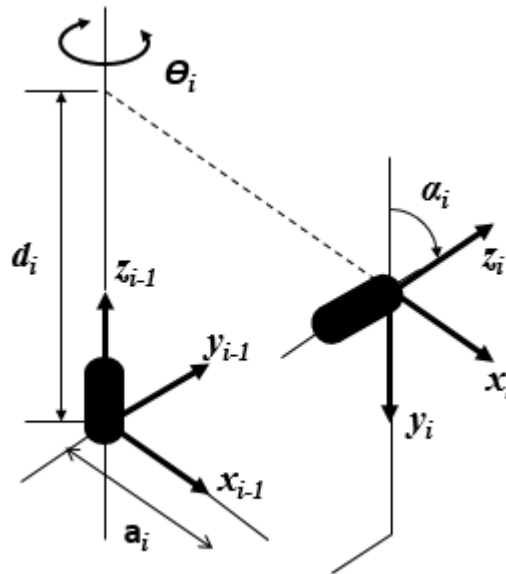
kullanarak belirlenir. Bir uzuv ile bir sonraki uzuv arasındaki bağıl öteleme ve dönmeyi gösteren homojen dönüşüm matrisi “A”, robot kol eklemlerinin konumunu belirleyen temel dönüşüm matrisi ise “T” ile gösterilir. Konum koordinatları elde edilirken öncelikle “Denavit-Hartenberg” yöntemi kullanılır.

#### 4.1.1. Denavit-Hartenberg (D-H) Yöntemi

Denavit-Hartenberg yönteminde, dört temel parametre kullanılarak robot kolun kinematik analizi yapılır. Bu parametreler, döner veya prizmatik eklemleri belirler.

- $\theta_i$  :  $x_{i-1}$  ekseninden  $x_i$  eksenine geçişte  $z_{i-1}$  eksenini etrafındaki dönme açısı
- $d_i$  : (i-1). koordinat sisteminin orijininden  $x_i$  ile  $z_{i-1}$  ekseninin kesim noktası arasındaki  $z_{i-1}$  eksenini doğrultusundaki öteleme
- $a_i$  : (i-1). koordinat sisteminin orijininden  $z_{i-1}$  ile  $x_i$  ekseninin kesim noktası arasındaki  $x_i$  eksenini doğrultusundaki öteleme
- $\alpha_i$  :  $z_{i-1}$  eksenini ile  $z_i$  eksenini arasındaki açı

Şekil 4.1’de kinematik bir çift üzerinde D-H parametrelerinin nasıl belirleneceği gösterilmiştir.



Şekil 4.1. Kinematik bir çift üzerinde D-H parametrelerinin belirlenmesi

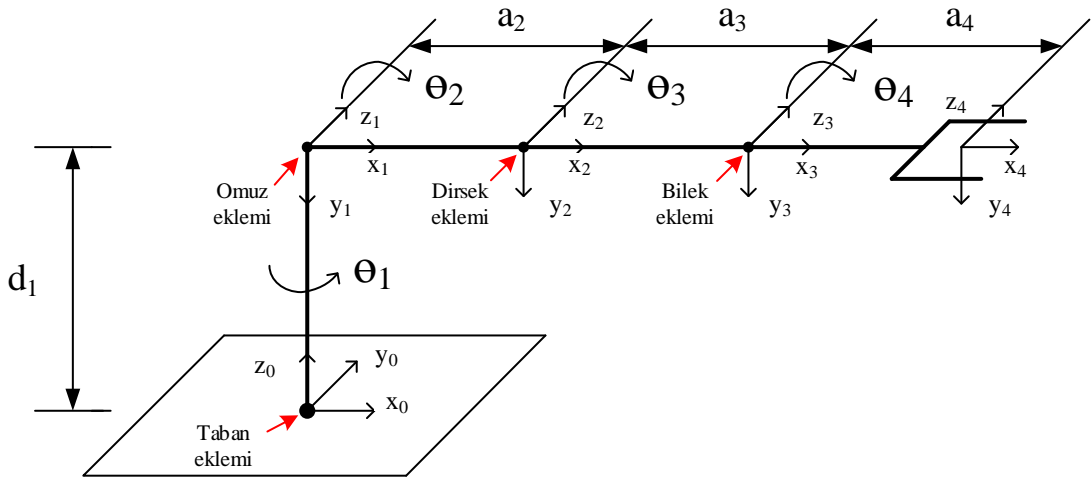


#### 4.1.2. Eksen Takımlarının Yerleştirilmesi

Eksen takımları, robot kol eklemlerine yerleştirilirken sırasıyla aşağıdaki işlemler gerçekleştirilir.

- Eklem eksenlerinin dönme ve kayma yönleri belirlenerek bu eksene paralel bir doğru çizilir.
- Eklem eksenleri döner eklemler için dönme yönü z, prizmatik eklemler için kayma yönü z-ekseni olarak belirlenir.
- Robot kol tutucu uç noktasına son eksene paralel bir eksen takımı yerleştirilir.
- z-eksenine dik olarak x-ekseni yerleştirilir.
- Temel dönüşüm matrisini hesaplamada kullanılmamasına rağmen y-ekseni, z ve x eksenleri yerleştirildikten sonra sağ el kuralı ile belirlenebilir.
- Art arda gelen iki eklem z-eksenindeki hareketleri aynı ise aynı x-eksenleri üzerinde oldukları belirlenir.

4 serbestlik dereceli robot kolun eklemlerine eksen takımlarının yerleştirilmesi Şekil 4.2'de gösterilmiştir.



Şekil 4.2. Robot kol eklemlerine eksen takımlarının yerleştirilmesi

Eksen takımları yerleştirildikten sonra, D-H tablosu Tablo 4.1'deki gibi oluşturulur.  $\theta_1, \theta_2, \theta_3, \theta_4$  sırasıyla taban, omuz, dirsek, bilek dönme açılarını ifade ederken  $d_1, a_2, a_3, a_4$  ise robot kol uzuvlarının uzunluklarını ifade etmektedir.

**Tablo 4.1.** D-H tablosu

<b>Eksen</b>	<b><math>\theta</math> (rad)</b>	<b>d (mm)</b>	<b>a (mm)</b>	<b><math>\alpha</math> (rad)</b>
<b>1</b>	$\theta_1$	$d_1$	0	$-\pi/2$
<b>2</b>	$\theta_2$	0	$a_2$	0
<b>3</b>	$\theta_3$	0	$a_3$	0
<b>4</b>	$\theta_4$	0	$a_4$	0

### 4.1.3. Homojen Dönüşüm Matrislerinin Bulunması

Her bir uzvun uzaydaki konumunu ve yönelimini belirten dönme (rotation) matrisi ve öteleme (translation) vektörü vardır. Homojen dönüşüm matrisi, dönme matrisi ve öteleme vektörünü birleştirmek için (4.1)'de gösterildiği gibi kullanılmaktadır. Ekleme sayısı kadar homojen dönüşüm matrisi bulunmaktadır.

$$A_i = \text{Rot}(z, \theta_i) \times \text{Trans}(0,0, d_i) \times \text{Trans}(a_i, 0,0) \times \text{Rot}(x, \alpha_i) \quad (4.1)$$

$\text{Rot}(z, \theta_i)$ : eklemin z eksenini etrafındaki dönme açısını ifade eden matris,

$\text{Trans}(0,0, d_i)$ : eklemin z eksenini doğrultusundaki ötelemesini ifade eden matris,

$\text{Trans}(a_i, 0,0)$ : eklemin x eksenini doğrultusundaki ötelemesini ifade eden matris,

$\text{Rot}(x, \alpha_i)$ : eklemler arasındaki z eksenlerinin birbirine olan açı izdüşümlerini ifade eden matristir.

(4.1)'de belirtilen matrisler (4.2)'deki gibi yerleştirilerek homojen dönüşüm matrisi oluşturulur. Bu matris, matris işlemlerinde çarpma kuralı gereği en sağdan başlayıp sola doğru çarpılarak sırasıyla (4.3), (4.4) ve son olarak da (4.5)'te gösterildiği gibi i. eklemin (i-1). ekleme göre konumunu ve yönelimini belirten tek bir matris haline getirilir.

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cdot \cos\alpha_i & \sin\theta_i \cdot \sin\alpha_i & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cdot \cos\alpha_i & -\cos\theta_i \cdot \sin\alpha_i & a_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

Şekil 4.2'deki 4 serbestlik dereceli bir robot kol için oluşturulan Tablo 4.1'deki D-H parametreleri kullanılarak 1, 2, 3 ve 4 no'lu eklemlerin homojen dönüşüm matrisleri sırasıyla (4.6), (4.7), (4.8) ve (4.9)'da gösterildiği gibi elde edilir.

$$A_1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

$$A_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \cdot \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2 \cdot \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$$A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3 \cdot \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3 \cdot \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

$$A_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_4 \cdot \cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & a_4 \cdot \sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

#### 4.1.4. Temel Dönüşüm Matrislerinin Bulunması

Her bir eklem için homojen dönüşüm matrisleri elde edildikten sonra bu matrisler çarpılarak (4.10)'daki temel dönüşüm matrisi olan "T" matrisi elde edilir.

$$T_0^N = A_1 A_2 \dots A_N \quad N: \text{eklem sayısı} \quad (4.10)$$

(4.11)'de belirtilen  $T_0^N$  matrisinde,  $(n, o, a)_{x,y,z}$ : robot kol tutucu ucunun yönelimini yani bir koordinat sisteminin başka bir koordinat sistemine göre dönme miktarını nitelendirirken,  $p_x, p_y, p_z$ : tutucu ucun x, y ve z konum koordinatlarını nitelendirir.

$$T_0^N = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

4 eklemli robot kolun dördüncü uzvu olan tutucu ucun ana koordinat sistemine göre temel dönüşüm matrisi (4.12)'de verilmiştir.

$$T_0^4 = A_1 A_2 A_3 A_4 \quad (4.12)$$

4 eklem için de oluşturulan homojen dönüşüm matrislerini kullanarak (4.12)'deki işlemler gerçekleştirildiğinde (4.13) elde edilir. (4.13)'teki matriste yer alan trigonometrik ifadeler için aşağıdaki kısaltmalar kullanılmıştır.

$$s_1 = \sin \theta_1, s_2 = \sin \theta_2, s_3 = \sin \theta_3, s_4 = \sin \theta_4$$

$$c_1 = \cos \theta_1, c_2 = \cos \theta_2, c_3 = \cos \theta_3, c_4 = \cos \theta_4$$

$$T_0^4 = \begin{bmatrix} c_1(c_2(c_3c_4 - s_3s_4) - s_2(c_3s_4 + c_4s_3)) & -c_1(c_2(c_3s_4 + c_4s_3) + s_2(c_3c_4 - s_3s_4)) \\ s_1(c_2(c_3c_4 - s_3s_4) - s_2(c_3s_4 + c_4s_3)) & -s_1(c_2(c_3s_4 + c_4s_3) + s_2(c_3c_4 - s_3s_4)) \\ -c_2(c_3s_4 + c_4s_3) - s_2(c_3c_4 - s_3s_4) & s_2(c_3s_4 + c_4s_3) - c_2(c_3c_4 - s_3s_4) \\ 0 & 0 \\ -s_1 & c_1(a_2c_2 + c_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - s_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3)) \\ c_1 & s_1(a_2c_2 + c_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - s_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3)) \\ 0 & d_1 - c_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3) - s_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - a_2s_2 \\ 0 & 1 \end{bmatrix} \quad (4.13)$$

Temel dönüşüm matrisinin elemanlarını ifade eden (4.11) göz önünde bulundurularak tutucu uç konum koordinatlarını ifade eden  $p_x$ ,  $p_y$  ve  $p_z$  sırasıyla (4.14), (4.15) ve (4.16)'daki gibi elde edilir.

$$p_x = c_1(a_2c_2 + c_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - s_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3)) \quad (4.14)$$

$$p_y = s_1(a_2c_2 + c_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - s_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3)) \quad (4.15)$$

$$p_z = d_1 - c_2(a_3s_3 + a_4c_3s_4 + a_4c_4s_3) - s_2(a_3c_3 + a_4c_3c_4 - a_4s_3s_4) - a_2s_2 \quad (4.16)$$

Elde edilen bu eşitlikler sonucunda düz kinematik analiz tamamlanır.

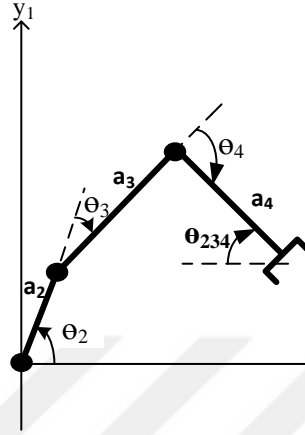
#### 4.2. Ters Kinematik Analiz

Ters kinematik analiz, düz kinematik analizin tersi olarak, verilen robot tutucu ucunun konum ve yönelimi için gerekli olan eklem değişken değerlerini elde etmek için yapılmaktadır. Başka bir deyişle, robot uç işlevcisinin ana koordinat sistemine göre elde edilen temel dönüşüm matrisindeki pozisyon ve yönelim parametrelerini kullanarak robot kol hareketi için gerekli olan her bir eklem dönmeye açıları ( $\theta_i$ ) elde etmek için yapılmaktadır.

Ters kinematik analiz, homojen dönüşüm matrisleriyle oluşturulan doğrusal olmayan denklemlerin çözümünü içerdiğinden düz kinematik analize göre daha kapsamlı hesaplamalar gerektirmektedir. Ters kinematik analizdeki doğrusal olmayan denklemlerin çözüm zorluğu; robot serbestlik derecesine, eklem tipine (prizmatik/döner eklem sayısı ve dizilişi) göre değişmektedir. Robotun sahip olduğu serbestlik derecesi ve döner eklem sayısı arttıkça ters kinematik problemlerinin çözümü zorlaşmaktadır. Serbestlik derecesi üçten daha fazla olan robotların ters kinematik problemlerinin çözümünde birçok çözüm kümesi ortaya çıkmakta ve çözümü bir hayli zorlaştırmaktadır. Bu nedenle problemlerin çözümünde bazı kısıtlamalar yaparak çözüme ulaşılmak istenmesi kaçınılmazdır.

Bu tez çalışmasında kullanılan 4 serbestlik dereceli robot kolun ters kinematik analizi için Şekil 4.3'te örnek olarak gösterildiği gibi robot kol tutucu ucunun başlangıç noktasına göre yönelim açısı yani omuz, dirsek ve bilek eklem açılarının toplamı olan  $\theta_{234}$  açı değerini önceden belirleyerek denklemlerin çözümü için kısıtlama yapılmıştır. Kısıtlama yaparak ters

kinematik denklemlerinin çözümlerine, Laurent GAY'ın 2007 yılında 3 serbestlik derecesine sahip Lynx robot kollar için uyguladığı yöntemi temel alarak çözümler ortaya koyan [63]'teki tez çalışması referans olmuştur.



**Şekil 4.3.** Robot kolun örnek bir hareketi sonucu son üç ekleminin dönme açılarının ( $\theta_2, \theta_3, \theta_4$ ) ve  $\theta_{234}$  yönelim açısının gösterimi

Ters kinematik analiz ile her bir eklemin açılarını belirlemek için öncelikle (4.11)'deki temel dönüşüm matrisi 4 eklemli robot kol için (4.17)'deki gibi elde edilir.

$$T_0^4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$

Daha sonra, (4.12) ve (4.17)'deki eşitlikleri kullanarak her iki taraf da  $A_1^{-1}$  ile çarpılırsa sırasıyla (4.18), (4.19), (4.20) elde edilir.

$$A_1^{-1}T_0^4 = A_1^{-1}A_1A_2A_3A_4 \quad (4.18)$$

$$A_1^{-1} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}A_1A_2A_3A_4 \quad (4.19)$$

$$A_1^{-1} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_2 A_3 A_4 \quad (4.20)$$

Trigonometrik ifadeler için aşağıdaki kısaltmaları da kullanarak (4.20)'deki işlemler yapıldığında (4.21) elde edilir.

$$s_{234} = \sin(\theta_2 + \theta_3 + \theta_4), c_{234} = \cos(\theta_2 + \theta_3 + \theta_4), s_{23} = \sin(\theta_2 + \theta_3), c_{23} = \cos(\theta_2 + \theta_3)$$

$$\begin{bmatrix} n_x c_1 + n_y s_1 & o_x c_1 + o_y s_1 & a_x c_1 + a_y s_1 & p_x c_1 + p_y s_1 \\ -n_z & -o_z & -a_z & d_1 - p_z \\ n_y c_1 - n_x s_1 & o_y c_1 - o_x s_1 & a_y c_1 - a_x s_1 & p_y c_1 - p_x s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{234} & -s_{234} & 0 & a_4 c_{234} + a_3 c_{23} + a_2 c_2 \\ s_{234} & c_{234} & 0 & a_4 s_{234} + a_3 s_{23} + a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

(4.21)'de yer alan iki matrisin eleman eşitliklerini kullanarak robot kol eklem açılarını ifade eden denklemler elde edilebilmektedir.

Bunun için iki matrisin 1. satır 4. sütun, 2. satır 4. sütun ve 3. satır 4. sütündeki elemanlar birbirine eşitlenirse (4.22), (4.23) ve (4.24) elde edilir.

$$p_x c_1 + p_y s_1 = a_4 c_{234} + a_3 c_{23} + a_2 c_2 \quad (4.22)$$

$$d_1 - p_z = a_4 s_{234} + a_3 s_{23} + a_2 s_2 \quad (4.23)$$

$$p_y c_1 - p_x s_1 = 0 \quad (4.24)$$

(4.24) kullanılarak taban dönme açısı olan  $\theta_1$ , (4.25)'teki gibi bulunur.

$$\theta_1 = \tan^{-1}(p_y/p_x) \quad \text{veya} \quad \theta_1 = \theta_1 + 180^\circ \quad (4.25)$$

$\theta_3$ 'ü bulmak için öncelikle (4.22)'deki  $a_4c_{234}$  ve (4.23)'teki  $a_4s_{234}$  eşitliklerin soluna atılır ve eşitliklerin her iki taraflarının kareleri alınır. Daha sonra  $c_{23} = c_2c_3 - s_2s_3$  ve  $s_{23} = s_2c_3 + c_2s_3$  trigonometrik fark formülleri kullanılarak eşitlikler taraf tarafa toplandığında (4.26), (4.27) ve (4.28) elde edilir.

$$(p_xc_1 + p_ys_1 - a_4c_{234})^2 + (d_1 - p_z - a_4s_{234})^2 = a_2^2 + a_3^2 + 2a_2a_3c_3 \quad (4.26)$$

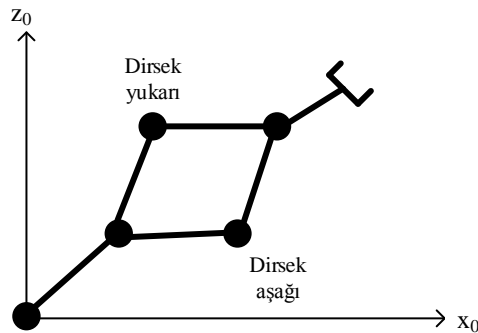
$$c_3 = ((p_xc_1 + p_ys_1 - a_4c_{234})^2 + (d_1 - p_z - a_4s_{234})^2 - a_2^2 - a_3^2)/2a_2a_3 \quad (4.27)$$

$$s_3 = \pm\sqrt{1 - c_3^2} \quad (4.28)$$

(4.27) ve (4.28) kullanılarak dirsek eklemının dönme açısı olan  $\theta_3$ , (4.29)'daki gibi bulunur.

$$\theta_3 = \tan^{-1}(s_3/c_3) \quad (4.29)$$

(4.28)'deki birden fazla çözüm kümesinden dolayı  $\theta_3$ 'ü ifade eden çözüm de birden fazla olabilmektedir. Bunun sebebi, Şekil 4.4'te görüldüğü gibi robot kol tutucusunun bir noktaya bazen birden fazla yoldan ulaşabilmesidir. Robot kolun istenen hareketi göz önünde bulundurularak çözüm kümesi teke düşürülür.



Şekil 4.4. Robot kol tutucusunun bir noktaya farklı pozisyonlarda yaklaşımı

(4.22) ve (4.23)'teki eşitlikler, yukarıda verilen trigonometrik fark formüllerini de kullanarak (4.30) ve (4.31)'deki gibi düzenlendikten sonra omuz eklemının dönme açısını elde etmek için  $s_2$  ve  $c_2$  (4.32) ve (4.33)'teki gibi yalnız bırakılır.



$$p_x c_1 + p_y s_1 - a_4 c_{234} = a_3 (c_2 c_3 - s_2 s_3) + a_2 c_2 \quad (4.30)$$

$$d_1 - p_z - a_4 s_{234} = a_3 (s_2 c_3 - c_2 s_3) + a_2 s_2 \quad (4.31)$$

$$s_2 = ((a_2 + a_3 c_3)(d_1 - p_z - a_4 s_{234}) - a_3 s_3 (p_x c_1 + p_y s_1 - a_4 c_{234})) / ((a_2 + a_3 c_3)^2 + a_3^2 s_3^2) \quad (4.32)$$

$$c_2 = ((a_2 + a_3 c_3)(p_x c_1 + p_y s_1 - a_4 c_{234}) + a_3 s_3 (d_1 - p_z - a_4 s_{234})) / ((a_2 + a_3 c_3)^2 + a_3^2 s_3^2) \quad (4.33)$$

(4.32) ve (4.33) kullanılarak omuz ekleminin dönme açısı olan  $\theta_2$ , (4.34)'teki gibi bulunur.

$$\theta_2 = \tan^{-1}(s_2/c_2) \quad (4.34)$$

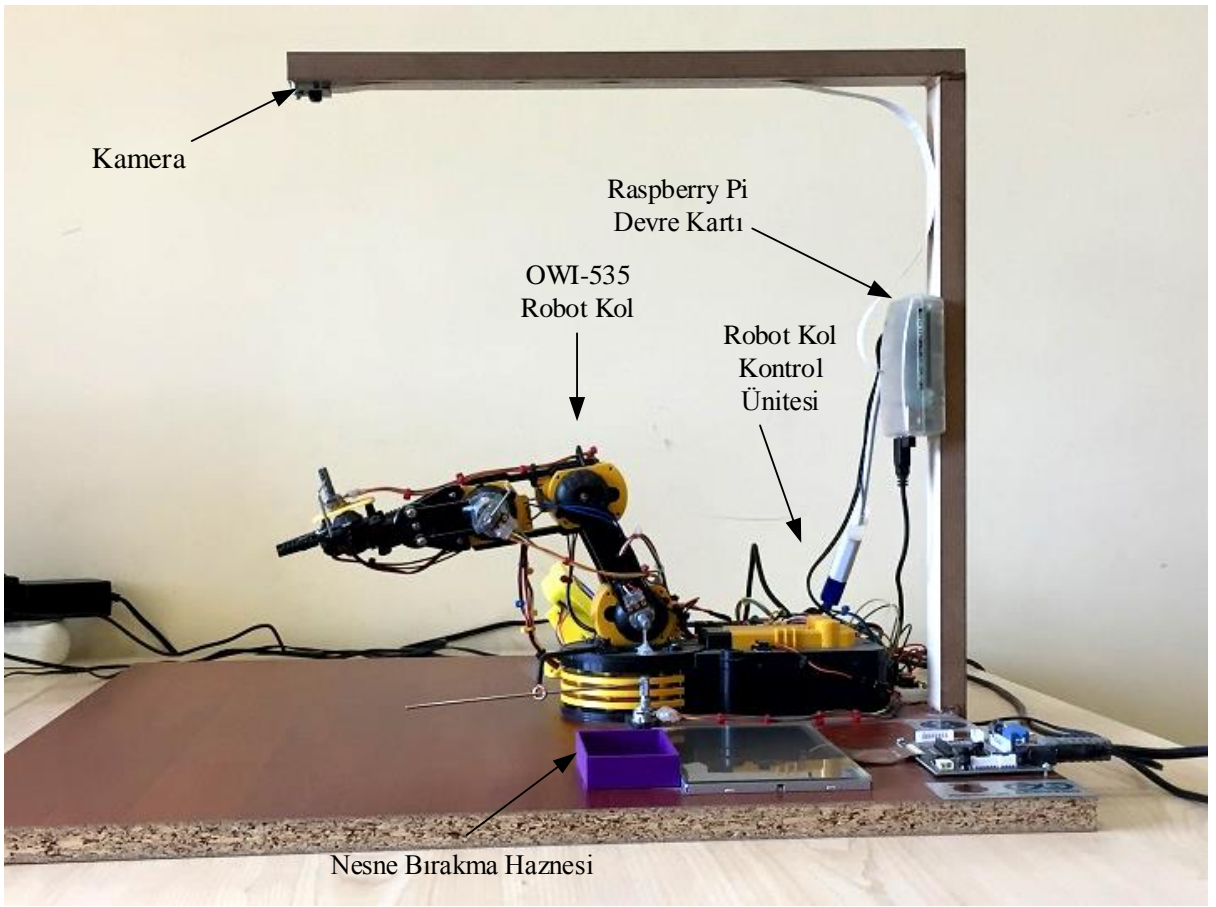
$\theta_{234}$  açı değeri önceden belirlendiği için bilek ekleminin dönme açısı olan  $\theta_4$ , (4.35)'teki gibi bulunur.

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (4.35)$$

Yukarıdaki eşitlikler sonucunda, robot kol tutucu ucunun gitmesi istenen  $p_x$ ,  $p_y$ ,  $p_z$  konum koordinatları ve tutucu uç için uygun bir  $\theta_{234}$  yönelim açısı girdileri sonucunda robot kol eklemlerinin bulunması gereken açı değerleri ( $\theta_1, \theta_2, \theta_3, \theta_4$ ) elde edilerek ters kinematik analiz tamamlanır.

## 5. DENEYSEL DÜZENEK VE ÇALIŞMA

Bu tez çalışmasında, 4 serbestlik dereceli OWI-535 robot kolunun çalışma alanında bulunan nesne veya nesneler yerel öznitelik tabanlı nesne bulma ve tanıma algoritmalar kullanılarak tanınıp yeri saptandıktan sonra sırasıyla nesneye yönelmesi, nesneyi kavraması ve nesneyi istenen bir hedefe bırakması amaçlanmıştır. Bu amaca uygun olarak oluşturulan deneysel düzenek Şekil 5.1’de gösterilmiştir.



Şekil 5.1. Deneysel düzenek

### 5.1. Deneysel Düzenegi Oluşturan Kısımlar

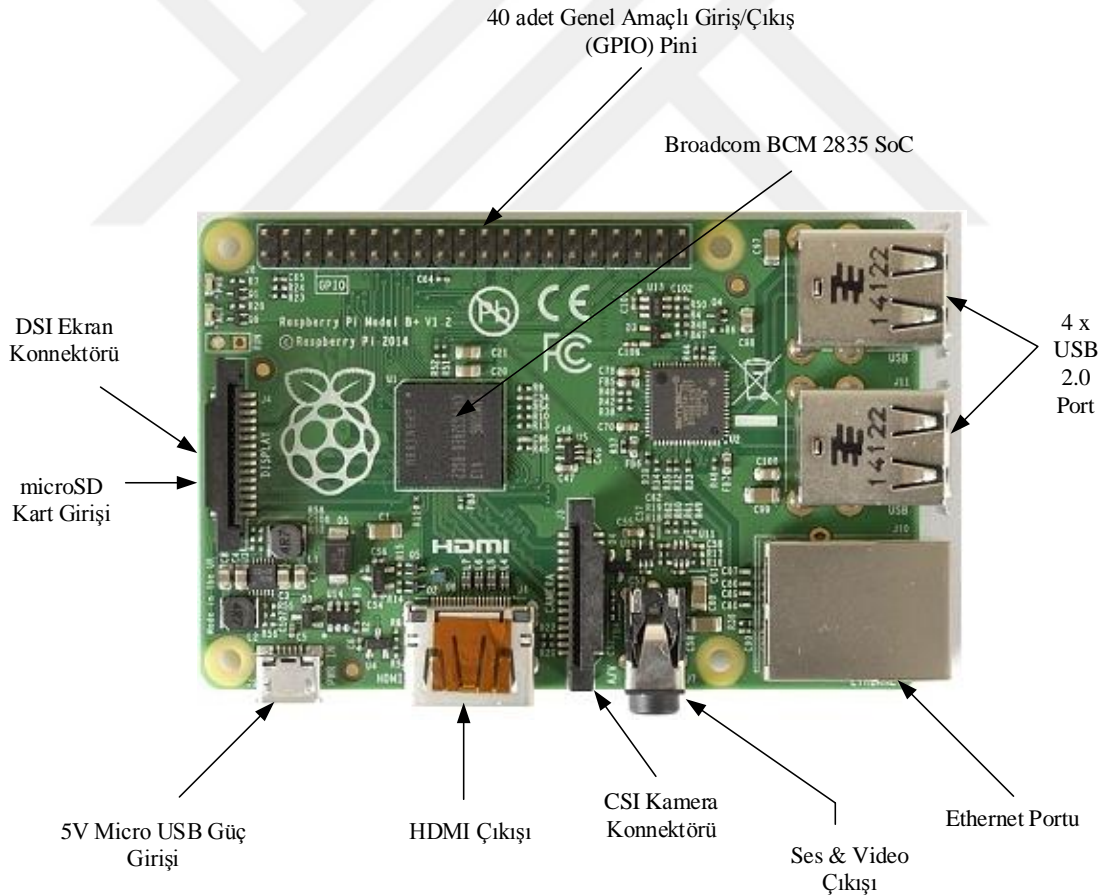
Tez çalışması için oluşturulan deneysel düzenek iki ana kısımdan oluşmaktadır. Bunlardan birincisi, robot kol çalışma alanında bulunan nesneyi tanıma ve yer tespiti işlemlerinin üzerinde gerçekleştirildiği Raspberry Pi devre kartı ve bu kart kullanılarak yapılacak işlemler için gerekli olan donanım ve yazılım gereksinimleridir. İkincisi ise,

tanınan ve yeri tespit edilen nesnenin kavranıp nesne bırakma haznesine bırakılmasını sağlayan OWI-535 robot kol ve kontrol ünitesidir.

### 5.1.1. Raspberry Pi Devre Kartı ve Gereksinimleri

Tez çalışmasında kullanılan Raspberry Pi devre kartı, Birleşik Krallık'ta Raspberry Pi Vakfı tarafından okullarda bilgisayar bilimini öğretmek amacıyla geliştirilmiş kredi kartı büyüklüğündeki tek kart bilgisayardır. Daha çok gömülü sistem uygulamalarında ve işletim sistemi gerektiren uygulamalarda kullanılan Raspberry Pi, bilgisayardan bağımsız ve kolaylıkla taşınabilir sistemler geliştirilebildiği için sıklıkla tercih edilmektedir.

Raspberry Pi, bazı donanımsal değişiklikler ile çeşitli modellerde piyasaya sürülmektedir. Tez çalışmasında Şekil 5.2'deki B+ modeli kullanılmıştır.

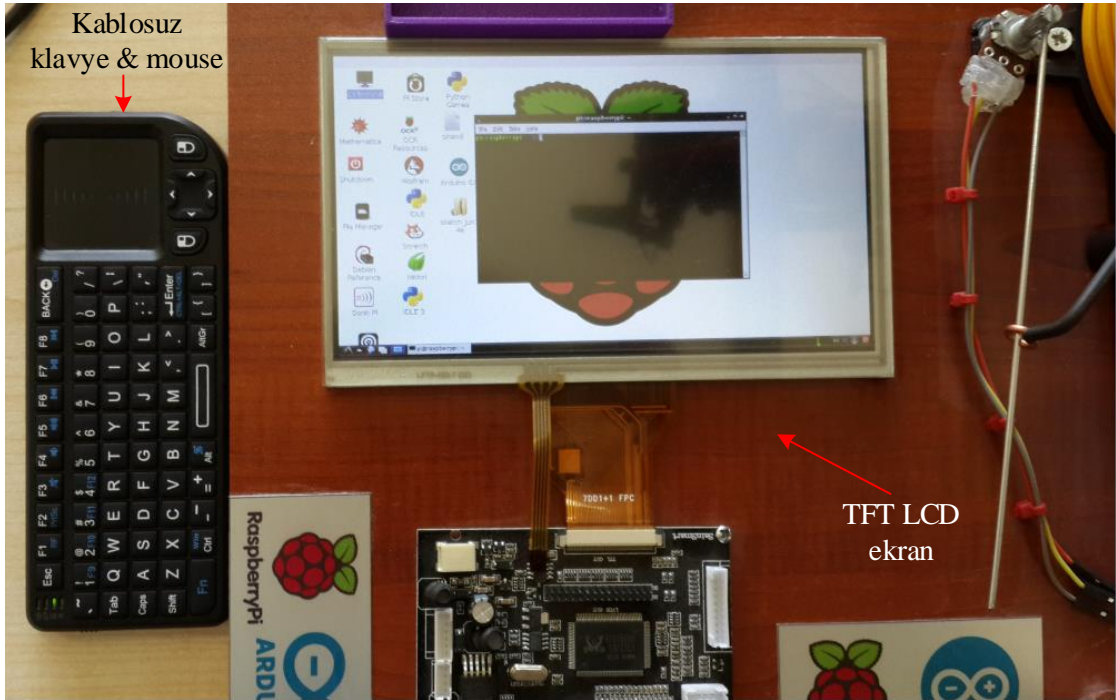


Şekil 5.2. Raspberry Pi B+ modeli

Şekil 5.2’de donanım birimleri gösterilen Raspberry Pi devre kartının B+ modeli, ARMv6 mimarisine sahip tek çekirdekli ARM1176JZ-F 700 MHz işlemci birimini içeren Broadcom BCM2835 SoC (System on Chip–Yongada Sistem) tabanlı olarak çalışmaktadır. Broadcom BCM2835 SoC’inde bulunan VideoCore IV grafik işlem birimi ve 512 MB RAM’e sahip olan bu Raspberry Pi kartı ile yüksek çözünürlüklü video oynatma dahil olmak üzere normal bir bilgisayarda yapılabilecek çoğu işlem yapılabilmektedir.

Klavye, mouse ve ekran bağlanarak mini bir bilgisayar olarak kullanılabilen bu kart üzerinde işletim sisteminin kurulabileceği ve verilerin depolanabileceği dahili bir sabit disk bulunmamaktadır. Bunun için kart üzerinde bulunan microSD kart yuvasına hafıza kartı yerleştirildikten sonra işletim sistemi kurulabilir ve istenen veri depolama işlemi gerçekleştirilebilir. Raspberry Pi B+ modeli, ARMv6 mimarisine sahip olduğu için Windows işletim sistemlerini çalıştıramamaktadır. Bunun için üretici vakfın web sitesinden Raspberry Pi için tasarlanmış Linux işletim sistemleri indirilerek hafıza kartı üzerine kurulmalıdır. Vakfın web sitesinde Raspbian (Debian Wheezy tabanlı) ve Pidora (Fedora tabanlı) gibi Linux işletim sistemleri mevcuttur.

Tez çalışmasında Raspberry Pi kartı için gerekli olan donanımlar ve hafıza kartına kurulan Raspbian işletim sisteminin masaüstü görünümü Şekil 5.3’te görülmektedir.



Şekil 5.3. Raspberry Pi için gerekli donanımlar

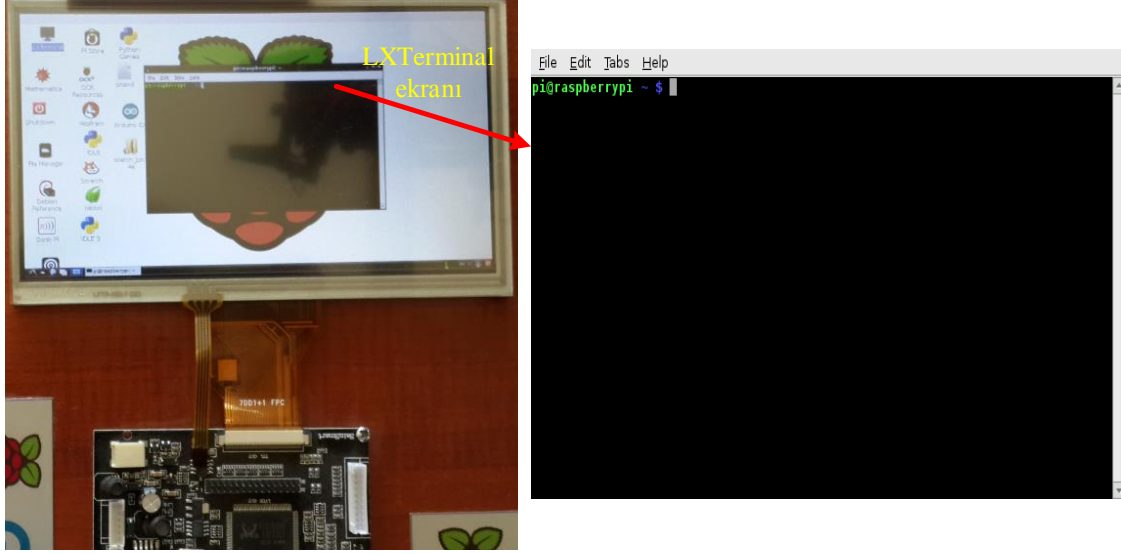
Ayrıca, robot kol çalışma alanındaki görüntülerin alınması için Raspberry Pi devre kartı üzerindeki CSI konektörüne doğrudan bağlanabilecek şekilde tasarlanmış olan 5MP çözünürlüklü Şekil 5.4'teki kamera modülü kullanılmıştır.



Şekil 5.4. Raspberry Pi kamera modülü

Raspberry Pi devre kartına işletim sistemi kurulduktan sonra kamera ile alınan görüntüde bulunan nesnenin tanınması ve yerinin saptanması işlemleri, OpenCV kütüphanesi kullanılarak nesne bulma ve tanıma algoritmalarıyla C++ dilinde geliştirilen yazılım ile gerçekleştirilmiştir. OpenCV, bilgisayarla görme alanındaki akademik ve ticari uygulamalar için 1999 yılında Intel tarafından piyasaya sürülen açık kaynak kodlu bir kütüphanedir. Ücretsiz olarak dağıtılıp geliştirmeye açık olan OpenCV kütüphanesi C diliyle yazılmış olup C++, C, Python ve Java arayüzleriyle Windows, Linux, Mac OS, iOS ve Android gibi platformlarda çalışabilmektedir. OpenCV kütüphanesinin MATLAB paket programı ve ilgili araç kutularının yerine tercih edilmesinin sebebi Raspberry Pi gibi gömülü sistemlerin kısıtlı bir işlemci ve belleğe sahip oluşudur. Ayrıca gömülü sistemler için MATLAB'ın uygun bir çözüm olmaması ve OpenCV kütüphanesinin bilgisayarla görme alanındaki yüksek hızlı performansıdır.

Şekil 5.5'te gösterilen komut satırı arayüzü "LXTerminal" kullanılarak OpenCV kütüphanesi ve GNU GCC derleyicisi internet üzerinden indirilip kurulmuştur.



Şekil 5.5. LXTerminal ekranı

### 5.1.2. OWI-535 Robot Kolu ve Kontrol Ünitesi

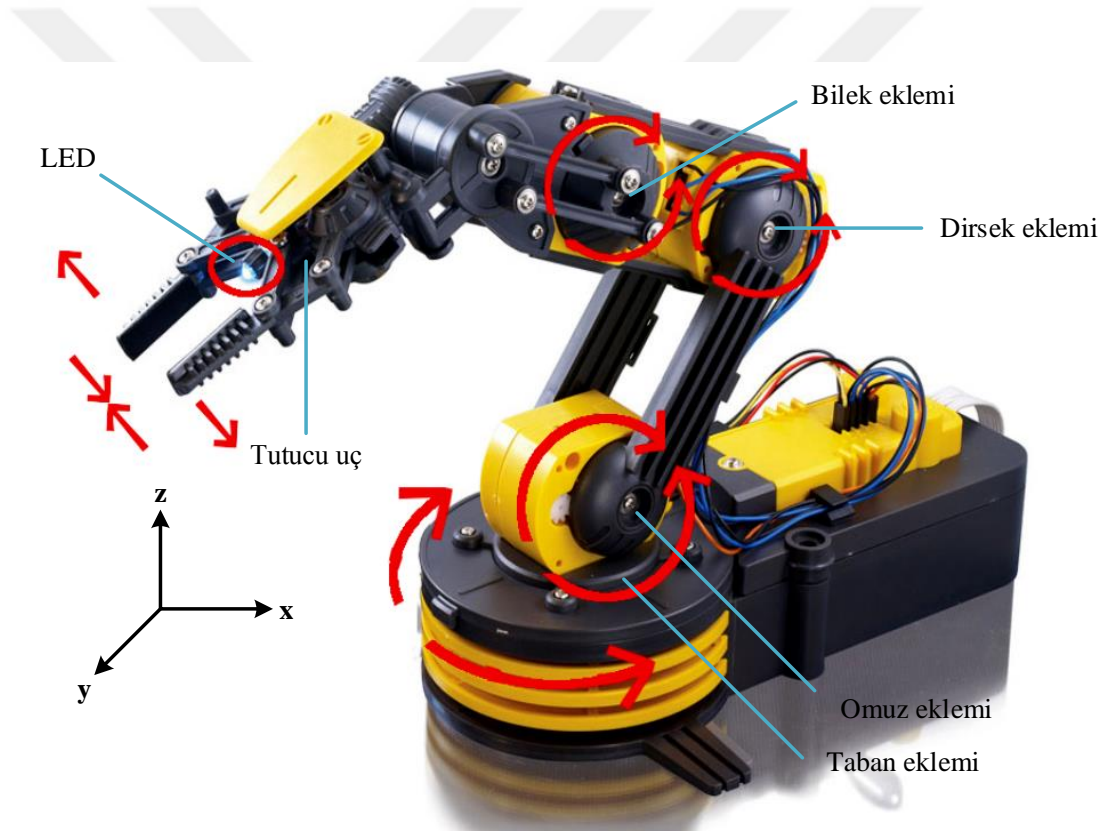
Tez çalışmasında kullanılan robot kolu OWI firmasının ürünlerinden biri olan OWI-535 robot koludur. Piyasaya kablolu uzaktan kumanda kontrollü olarak sürülen robot kolun eklemlerinde yer alan motorlarının kabloları kumandadan sökülerek kumandadan bağımsız bir kontrol ünitesi oluşturulmuştur.

#### 5.1.2.1. OWI-535 Robot Kolu

Eklemlili manipülör sınıfında yer alan Şekil 5.6'daki OWI-535 robot kolu, eğitimsel amaçlar için tasarlanmış olup, düşük maliyetli bir robot koldur. İnsan kolunun yapısına benzeyen bu robot kol, Şekil 5.7'de gösterildiği gibi; taban, omuz, dirsek ve bilek eklemlerinden oluşan 4 serbestlik derecesine ve nesnelere kavramak için tasarlanmış tutucu bir uca sahiptir. 4 döner ekleme sahip olan robot kolun, omuz, dirsek ve bilek eklemleri; kolu x-ekseni etrafında döndürürken, taban eklemleri; z-ekseni etrafında kolu döndürür. Hiçbir eklemler y-ekseni etrafında dönmediğinden kol hareketleri sınırlansa da bu durum kinematik hesaplamaları kolaylaştırır. Robot kolun boyutları  $22.86 \times 16 \times 38$  cm, ağırlığı 658 gr, çalışma alanı dikeyde 38 cm, yatayda 32 cm ve robot kol tutucusu max. 100 gr kaldırabilmektedir.

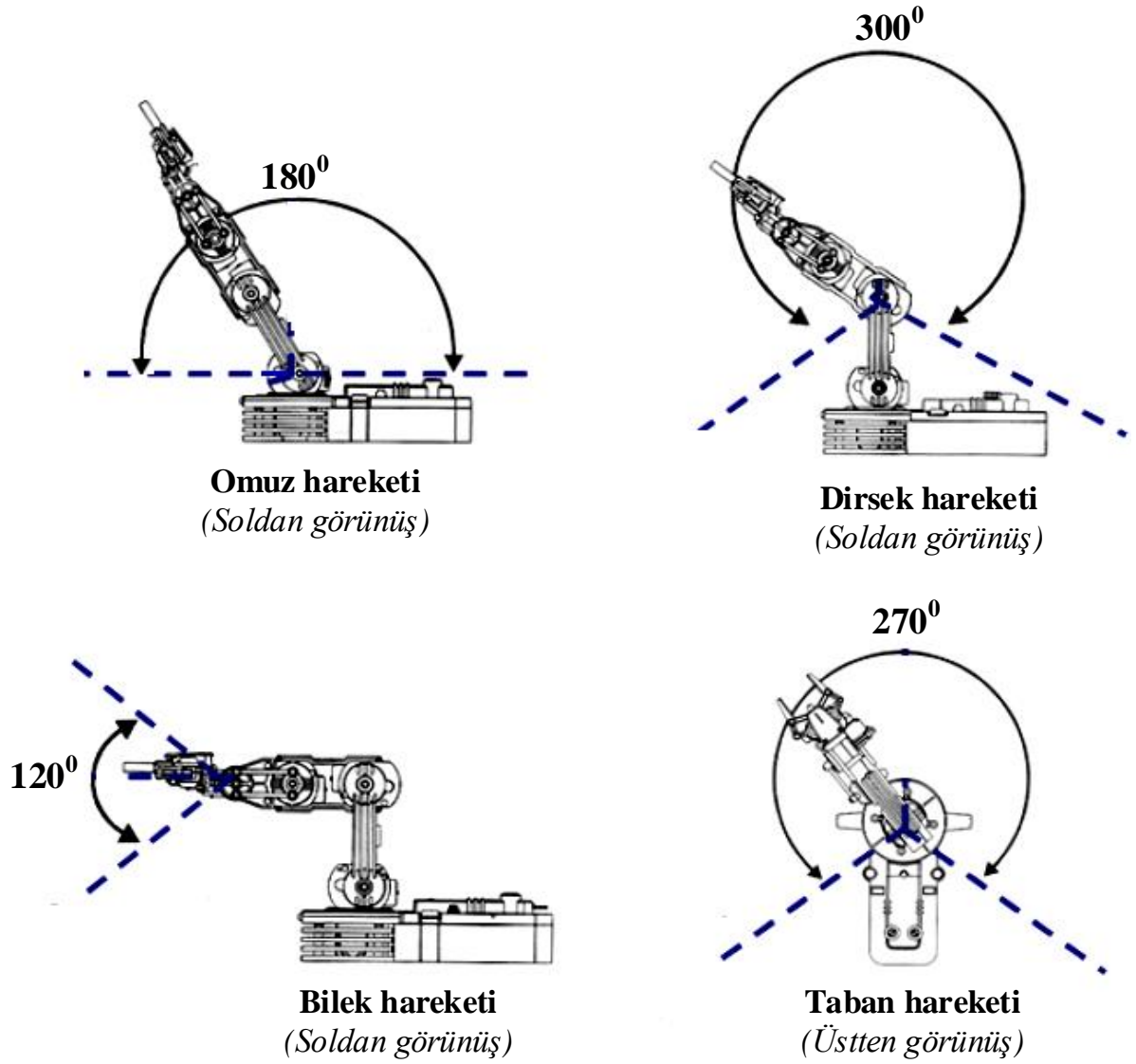


Şekil 5.6. OWI-535 robot kolu



Şekil 5.7. OWI-535 robot kolunun eklemleri ve dönüş yönleri

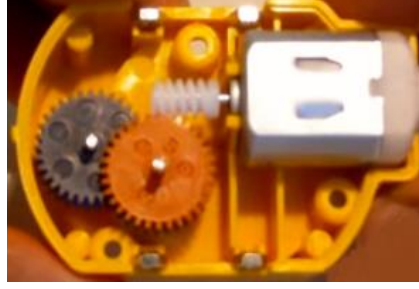
4 döner eklem için robot kolun çalışma alanını belirleyen dönme sınırları vardır. Omuz, dirsek ve bilek eklemleri için ileri ve geri yönlerde, taban için sağa ve sola dönme sınırları bulunmaktadır. Robot kol eklem hareketlerinin açıl dönme sınırları Şekil 5.8'de gösterilmiştir.



Şekil 5.8. Robot kol eklem hareketlerinin açısal dönme sınırları

Robot koldaki her bir eklem, Şekil 5.9’da gösterilen DA (Doğru Akım) motorları ve dişli kutular yardımıyla hareket etmektedir. Robot kol, taban dönmesi, omuz hareketi, bilek hareketi, dirsek hareketi ve tutucunun hareketi ile birlikte 5 hareketi gerçekleştirebilmek için 5 adet DA motoru-dişli kutusu çiftine sahiptir. Dişli kutusu ile motor hızlarının düşürülerek torklarının artması sağlanmaktadır.

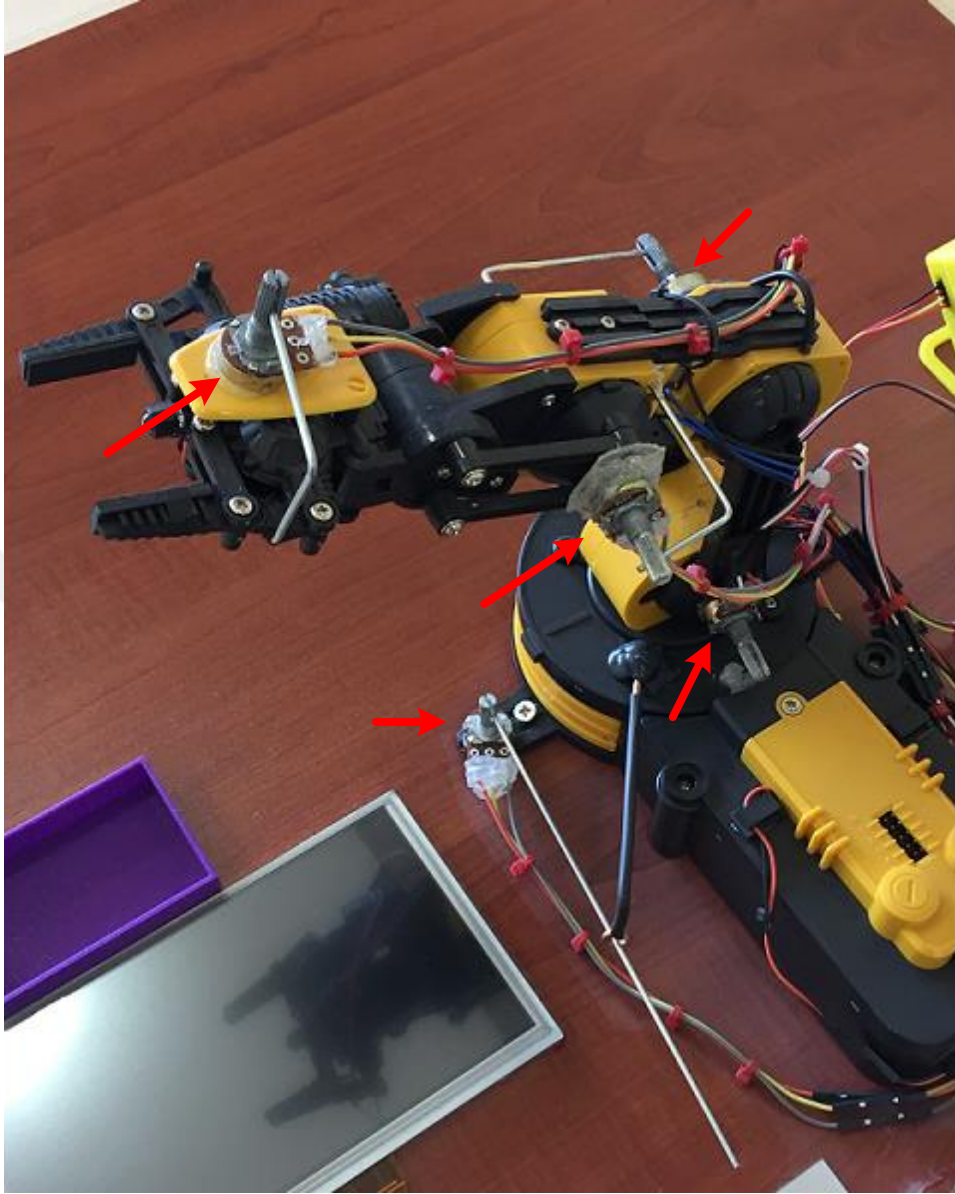




**Şekil 5.9.** DA motoru ve dişli kutusu

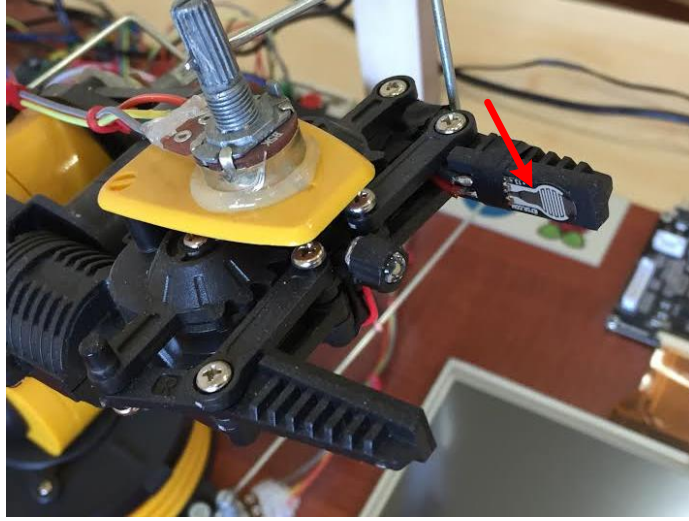
OWI-535 robot kol, endüstride kullanılan robot kollar ile kıyaslandığında hız ve güç açısından çok düşük kalmasına rağmen her bir eklemden yer alan dişli kutularına aşırı yük binmesi durumunda veya eklemin dönüş açısı sınırlarından fazla döndürüldüğünde dişlilerin diş atarak zarar görmemesi şeklinde tasarlanmıştır. Ayrıca, robot kolun dönüş hızı sabit değildir. Örneğin; omuzun aşağı hareketi yukarı hareketinden daha hızlıdır.

Bu robot kol, eklemlerinde yer alan DA motorlarından dolayı eklemin pozisyonlarının belirlenebilmesi için geri beslemeli bir mekanizmaya sahip değildir. Robot kol eklemleri, DA motoru-dişli kutusu çifti için tasarlandığı için eklemin motorlarını servo gibi geri besleme ile pozisyon bilgisinin alınabildiği motorlarla değiştirmek de neredeyse imkânsızdır. Bunun için tez çalışmasında, robot kolun her bir eklemine ve tutucu ucuna açıklık veya kapalılık kontrolünün yapılabilmesi için tutucu ucuna Şekil 5.10'da gösterildiği gibi sensör etkisi yapan potansiyometreler yerleştirilip potansiyometrelerden okunan bilgiler ile eklemin açısı arasında ilişkisi kurularak geri beslemeli bir sistem elde edilip pozisyon kontrolü sağlanmaktadır.



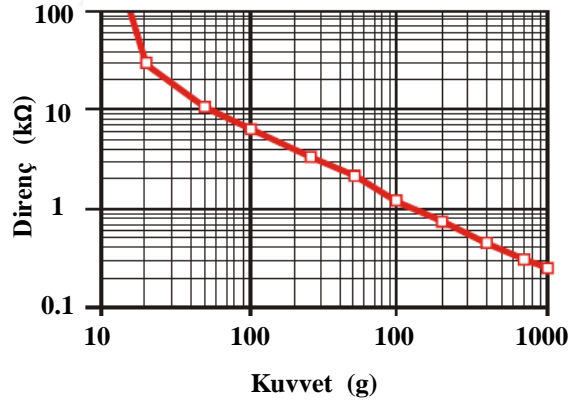
**Şekil 5.10.** Potansiyometrelerin yerleştirilmesi

Ayrıca, robot kol tutucu ucunun nesneyi kavrayacağı esnada ne kadar kapanması gerektiğini belirlemek için tutucu ucun bir tarafına FSR (Force Sensitive Resistor) yani kuvvet algılayıcı direnç Şekil 5.11'deki gibi yerleştirilmiştir.



Şekil 5.11. Kuvvet algılayıcı direncin yerleştirilmesi

Kuvvete duyarlı bu direnç, 6.5 mm çapındaki aktif olan dairesel alana uygulanan kuvvetle orantılı olarak direnç değerinde Şekil 5.12'deki karakteristiğe göre düşüş meydana gelmektedir. Bu düşüş göz önünde bulundurularak eşik bir değer belirlenerek tutucu uçta meydana gelen kuvvet algılanabilmektedir.

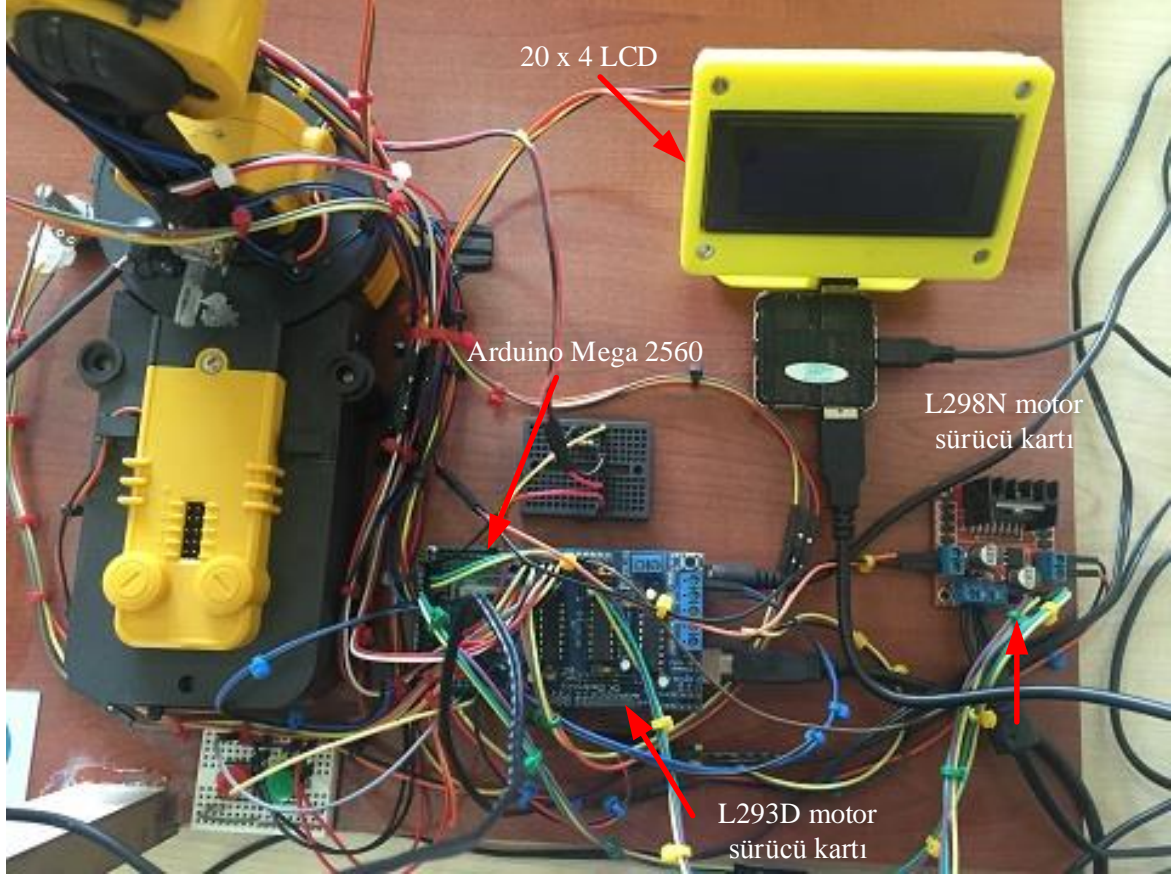


Şekil 5.12. FSR'nin kuvvet-direnç karakteristiği

### 5.1.2.2. Robot Kol Kontrol Ünitesi

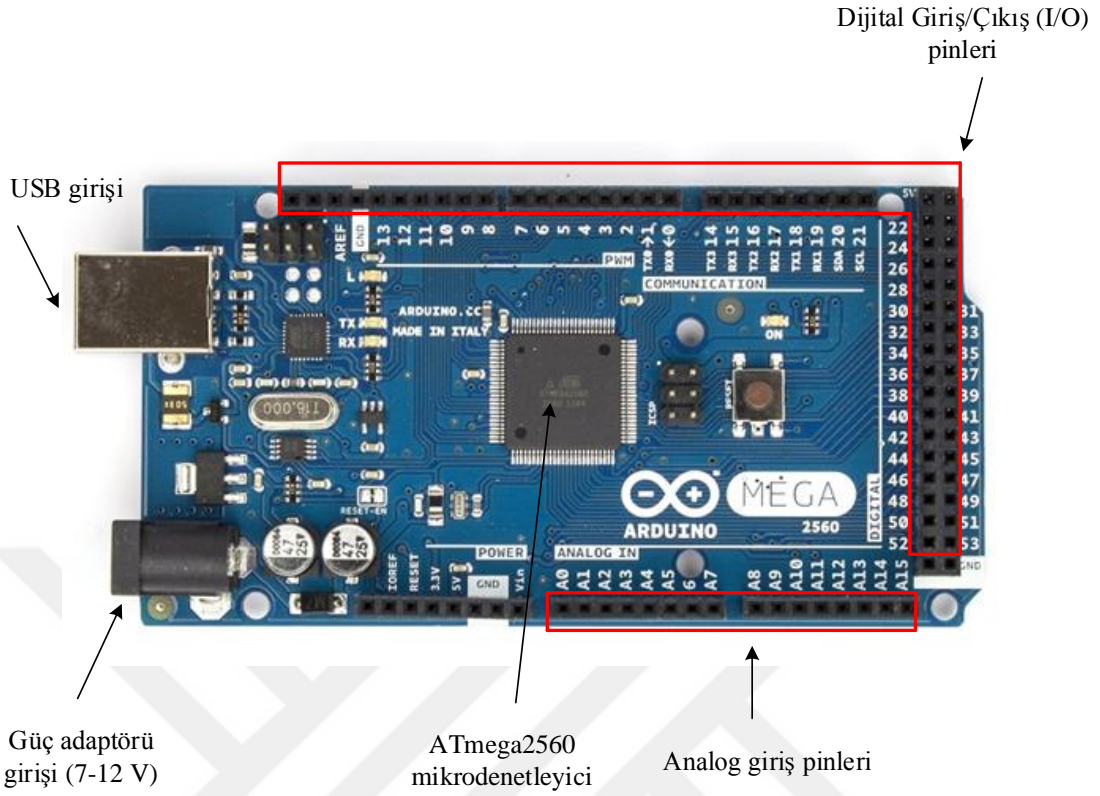
OWI-535 robot kol, kablolu uzaktan kumanda ile piyasaya sürüldüğü için robot kolun kontrolü kumanda üzerinde bulunan switch'ler ile yapılmaktadır. Tez çalışmasında, uzaktan kumandaya bağlı DA motorlarının kabloları sökülüp Arduino Mega 2560 ve DA motor

sürücü kartları ile motorların kontrolünü sağlamak için Şekil 5.13'teki kontrol ünitesi oluşturulmuştur.



Şekil 5.13. Robot kol kontrol ünitesi

Şekil 5.14'teki Arduino Mega 2560 kartı, Atmel firmasının 8 bitlik ATmega2560 mikrodenetleyici tabanlı bir Arduino kartıdır.



Şekil 5.14. Arduino Mega 2560

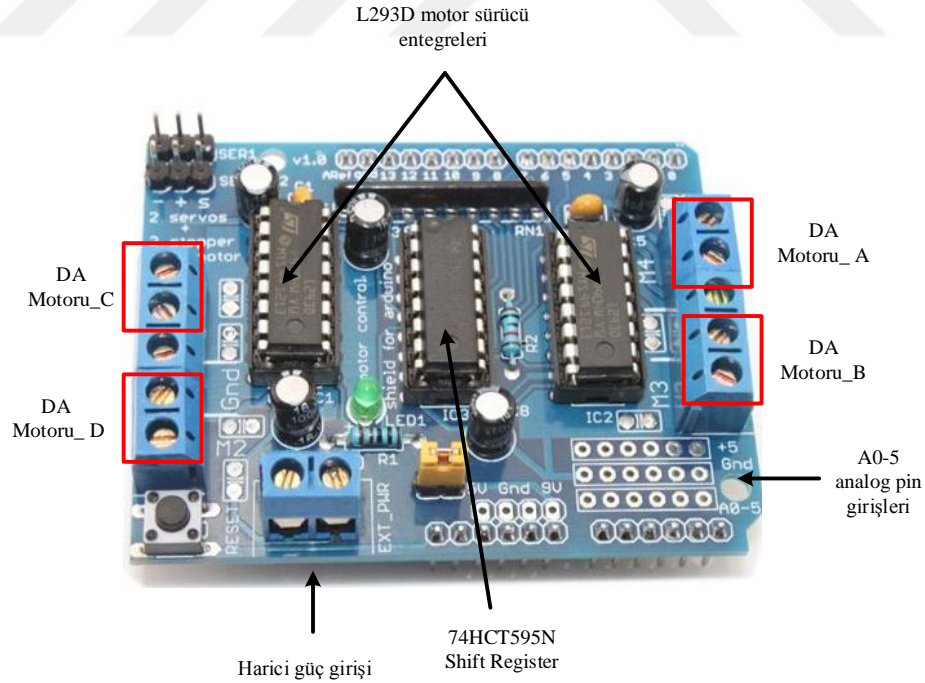
Fazla sayıda pine sahip olmasından dolayı tercih edilen bu Arduino kartında, 54 adet dijital I/O pini bulunmakta olup bu pinlerin 14 tanesi PWM (Pulse Width Modulation–Darbe Genişlik Modülasyonu) çıkışı, 6 tanesi de harici kesme olarak kullanılabilir. Ayrıca, 16 adet analog girişi sağlayan ADC (Analog to Digital Converter–Analog–Dijital Çevirici), 4 adet UART, 16 MHz kristal osilatörü, ICSP çıkışı ve bir reset butonu bulunmaktadır. Sahip olduğu 256 kB flash belleğin 8 kB’ını bootloader kullanmaktadır. Bootloader’ı sayesinde harici bir programlayıcıya ihtiyaç duyulmadan Arduino IDE’si ile programlanabilmektedir.

OWI–535 robot kol kontrol ünitesinde, Arduino kartının kullanılmasının öncelikli sebebi Raspberry Pi devre kartında dahili ADC biriminin bulunmamasıdır. Yani, robot kol eklemlerindeki potansiyometrelerden gelen analog değerlerin okunup dijital değerlere çevrilerek eklem açılarının belirlenmesi ve FSR ile tutucu uçta meydana gelebilecek herhangi bir kuvvetin algılanması işlemleri Raspberry Pi devre kartında doğrudan gerçekleştirilememektedir. Bunun için Raspberry Pi devre kartına harici ADC birimleri eklemek yerine dâhili ADC birimleri bulunan ve Raspberry Pi devre kartı ile kolaylıkla haberleşebilen Arduino kartı tercih edilmiştir. Ayrıca, Raspberry Pi devre kartını kullanarak

robot kol tutucusunun kavrayacağı nesnenin yeri tespit edildikten sonra eklem açılarını belirleyen ters kinematik denklem hesaplamalarında zorlanmaması ve Arduino için tasarlanmış birçok motor sürücü kart ile DA motorlarının rahatça sürülebilmesinden dolayı da Arduino kartının kullanılması fayda sağlamaktadır.

Arduino ile kullanılan Şekil 5.13'te gösterilen 20x4 LCD ise robot kol tutucusunun Raspberry Pi kartında gerçekleşen işlemler ile yeri tespit edilen nesneyi kavrayabilmesi için gitmesi gereken koordinatlara erişip erişemeyeceği bilgisini kullanıcıya bildirmeyi sağlamaktadır.

Robot kol eklemlerinde bulunan motorların sürülebilmesi için Arduino'nun pinleri doğrudan kullanılamaz. Çünkü Arduino'nun her bir pininden maksimum 40 mA çıkış sağlanabilmektedir. Bu yüzden, motorların çektikleri akımlar göz önünde bulundurularak uygun olan DA motor sürücü entegreleri kullanılmalıdır. OWI-535 robot kolundaki dirsek ve bilek eklemlerindeki motorların ileri-geri hareketi ve tutucu ucun açılıp kapanmasını sağlayan bu 3 motorun kontrolü Şekil 5.15'teki DA motor sürücü kartı kullanılarak gerçekleştirilmiştir.

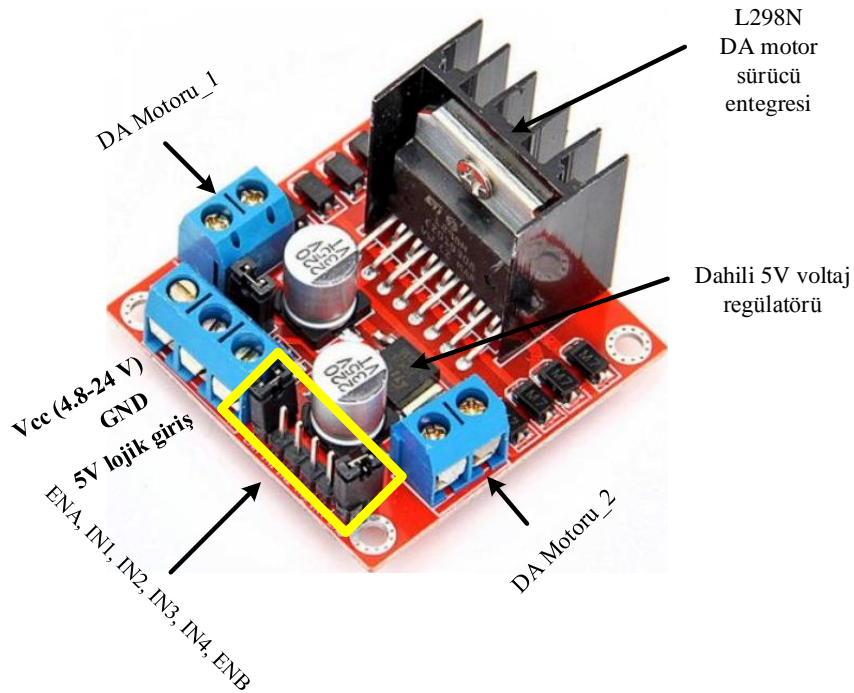


Şekil 5.15. L293D motor sürücü kartı

Şekil 5.15'teki bu motor sürücü kartı, Arduino üzerine yerleştirilebilecek şekilde tasarlanmış olup 4 adet DA motorunu veya 2 adet step motoru (unipolar/bipolar) ve 2 adet

servo motoru kontrol edebilmeyi sağlamaktadır. Aynı anda 4 adet DA motorunun birbirinden bağımsız olarak çift yön kontrolü için 8 pin ve PWM çıkışı ile istenen DA motorunun hız kontrolü için de 4 pin olmak üzere toplam 12 pin, ayrıca 2 adet servo motor sinyal pini de eklenince toplamda Arduino'nun 14 pinini kullanmak gerekmektedir. Kart üzerinde bulunan 74HCT595N Shift Register entegresi sayesinde L293D sürücü entegrelerinin giriş pinleri kontrol edilerek 8 olan motor yön kontrolü için gerekli olan pin sayısı 4'e indirilerek toplamda 14 yerine 10 pin ile bahsedilen motorların kontrolü gerçekleştirilmektedir.

Motor sürücü kart üzerinde yer alan 2 adet L293D motor sürücü entegresi içerisindeki H köprüleri 0.6A sürekli akıma kadar DA motorlarının sürülebilmesine izin vermektedir. OWI-535 robot kolunun dirsek, bilek ve tutucu ucunu tahrik eden DA motorları için gerekli olan akım bu sürücü kart ile karşılanabildiğinden motorların kontrolü başarılı bir şekilde gerçekleştirilebilmektedir. Ancak robot kolun taban ve omuz eklemlerine diğer eklemlerden daha fazla yük bindiği için daha fazla akım çekmekte ve verimli bir şekilde sürülemeyerek L293D sürücü entegrelerinin aşırı şekilde ısınmasına yol açmaktadır. Bu yüzden daha yüksek akım çeken taban ve omuz eklemlerini sağlayan motorların kontrolü Şekil 5.16'daki L298N motor sürücü kartı ile gerçekleştirilmiştir.



Şekil 5.16. L298N motor sürücü kartı

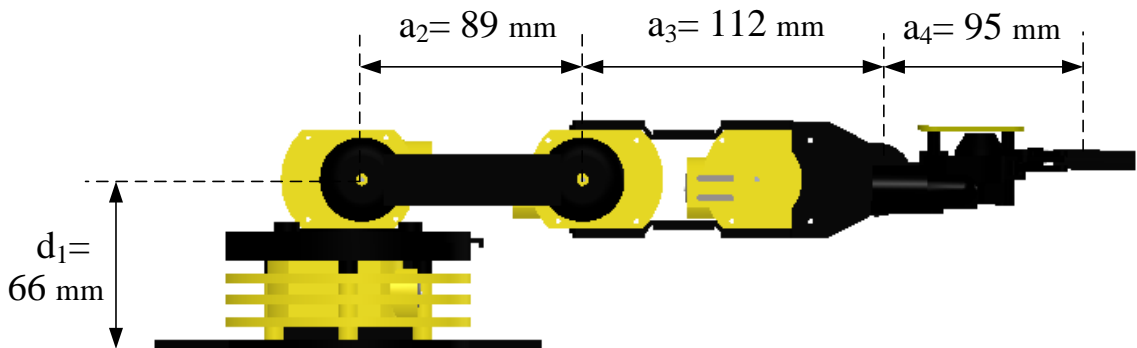
Bu motor sürücü kart, 2 kanallı olup kanal başına 2A'ye kadar akım vererek birbirinden bağımsız 2 adet DA motoru sürülebilmektedir. Şekil 5.16'da gösterilen IN, IN2, IN3 ve IN4 pinleri ile motorların ileri-geri yön kontrolü yapılabilirken ENA ve ENB pinleri ile de hız kontrolü yapılabilmektedir. Kart üzerinde bulunan L298N motor sürücü entegresi ile DA motorlarından ayrı olarak step motor kontrolüne de imkân sağlayabilmektedir.

## 5.2. OWI-535 Robot Kol Kinematik Analizi

OWI-535 robot kol tutucusunun yeri tespit edilen bir nesneyi kavrayıp istenilen bir hedefe bırakabilmesi için önce düz kinematik, daha sonra da ters kinematik analizinin yapılarak eklemlerinin bulunması gereken açıları ifade eden eşitliklerin elde edilmesi gerekmektedir. Bu işlemler için 4. Bölüm'de 4 serbestlik dereceli bir robot kol için anlatılan ve oluşturulan kinematik denklemler, 4 serbestlik dereceli OWI-535 robot kolu için de aynı şekilde uygulanabilmektedir.

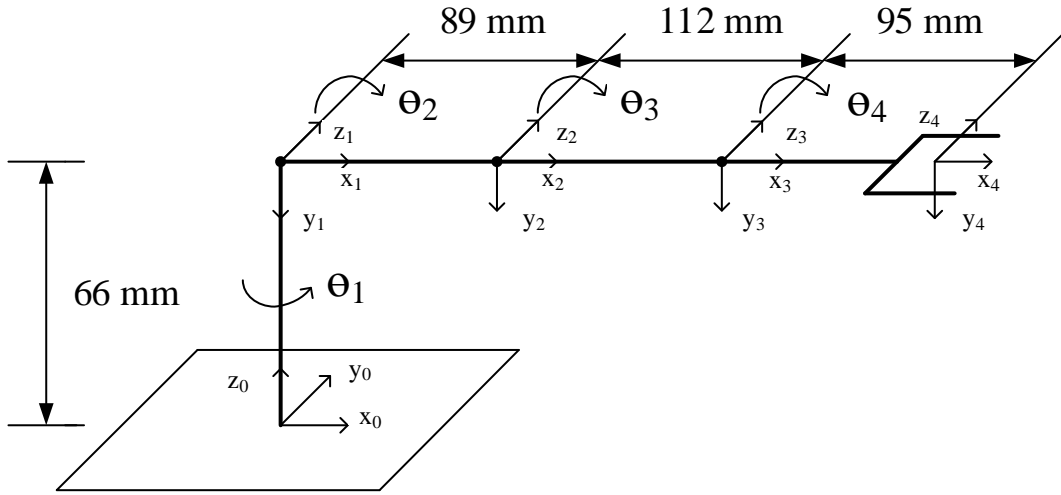
### 5.2.1. OWI-535 Robot Kol Düz Kinematik Analizi

OWI-535 robot kol eklemlerinin aldıkları açılara göre tutucu ucun ulaşacağı konumu hesaplayan düz kinematik analizi için ilk olarak robot kolun dönme eksenleri belirlenip eklemlerine eksen takımları yerleştirilmiştir. Robot kol uzuv uzunlukları Şekil 5.17'de verilen OWI-535 robot kolun eksen takımlarının yerleştirilme işlemi de Şekil 5.18'de gösterilmiştir.



Şekil 5.17. OWI-535 robot kol uzuvlarının uzunlukları





Şekil 5.18. OWI-535 robot kol eklemlerine eksen takımlarının yerleştirilmesi

Eksen takımları yerleştirildikten sonra robot kol uzuv uzunlukları da kullanılarak OWI-535 robot kolu için oluşturulan D-H tablosu Tablo 5.1’de verilmiştir.

Tablo 5.1. OWI-535 robot kol D-H tablosu

Eksen	$\theta$ (rad)	d (mm)	a (mm)	$\alpha$ (rad)
1	$\theta_1$	66	0	$-\pi/2$
2	$\theta_2$	0	89	0
3	$\theta_3$	0	112	0
4	$\theta_4$	0	95	0

D-H tablosundaki parametreler kullanılarak bir eklemin bir önceki ekleme göre konumunu ve yönelimini belirleyen OWI-535 robot koluna ait homojen dönüşüm matrisleri elde edilmiştir. Sırasıyla taban, omuz, dirsek ve bilek eklemleri için elde edilen homojen dönüşüm matrisleri (5.1), (5.2), (5.3) ve (5.4)’te verilmiştir.

$$A_1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 66 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$A_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 89\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 89\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$$A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 112\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 112\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$$A_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 95\cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & 95\sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

Her bir eklem için homojen dönüşüm matrisleri elde edildikten sonra bu matrisler çarpılarak OWI-535 robot kol tutucusunun ana koordinat sistemine göre temel dönüşüm matrisi (5.5)'teki gibi elde edilmiştir.

$$T_0^4 = A_1 A_2 A_3 A_4 \quad (5.5)$$

OWI-535 robot kol eklemleri için oluşturulan homojen dönüşüm matrisleri ve aşağıdaki trigonometrik kısaltmalar kullanılarak (5.6) elde edilmiştir.

$$s_1 = \sin \theta_1, s_2 = \sin \theta_2, s_3 = \sin \theta_3, s_4 = \sin \theta_4$$

$$c_1 = \cos \theta_1, c_2 = \cos \theta_2, c_3 = \cos \theta_3, c_4 = \cos \theta_4$$

$$T_0^4 = \begin{bmatrix} c_1(c_2(c_3c_4 - s_3s_4) - s_2(c_3s_4 + c_4s_3)) & -c_1(c_2(c_3s_4 + c_4s_3) + s_2(c_3c_4 - s_3s_4)) \\ s_1(c_2(c_3c_4 - s_3s_4) - s_2(c_3s_4 + c_4s_3)) & -s_1(c_2(c_3s_4 + c_4s_3) + s_2(c_3c_4 - s_3s_4)) \\ -c_2(c_3s_4 + c_4s_3) - s_2(c_3c_4 - s_3s_4) & s_2(c_3s_4 + c_4s_3) - c_2(c_3c_4 - s_3s_4) \\ 0 & 0 \end{bmatrix}$$

$$\left. \begin{array}{l} -s_1 \quad c_1(89c_2 + c_2(112c_3 + 95c_3c_4 - 95s_3s_4) - s_2(112s_3 + 95c_3s_4 + 95c_4s_3)) \\ c_1 \quad s_1(89c_2 + c_2(112c_3 + 95c_3c_4 - 95s_3s_4) - s_2(112s_3 + 95c_3s_4 + 95c_4s_3)) \\ 0 \quad 66 - c_2(112s_3 + 95c_3s_4 + 95c_4s_3) - s_2(112c_3 + 95c_3c_4 - 95s_3s_4) - 89s_2 \\ 0 \quad 1 \end{array} \right\} \quad (5.6)$$

Temel dönüşüm matrisinin elemanlarını ifade eden (5.7) kullanılarak OWI-535 robot kol tutucusunun konum koordinatları olan  $p_x$ ,  $p_y$  ve  $p_z$ , (5.8), (5.9) ve (5.10)'daki gibi elde edilerek düz kinematik analiz işlemi tamamlanmıştır.

$$T_0^4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

$$p_x = c_1(89c_2 + c_2(112c_3 + 95c_3c_4 - 95s_3s_4) - s_2(112s_3 + 95c_3s_4 + 95c_4s_3)) \quad (5.8)$$

$$p_y = s_1(89c_2 + c_2(112c_3 + 95c_3c_4 - 95s_3s_4) - s_2(112s_3 + 95c_3s_4 + 95c_4s_3)) \quad (5.9)$$

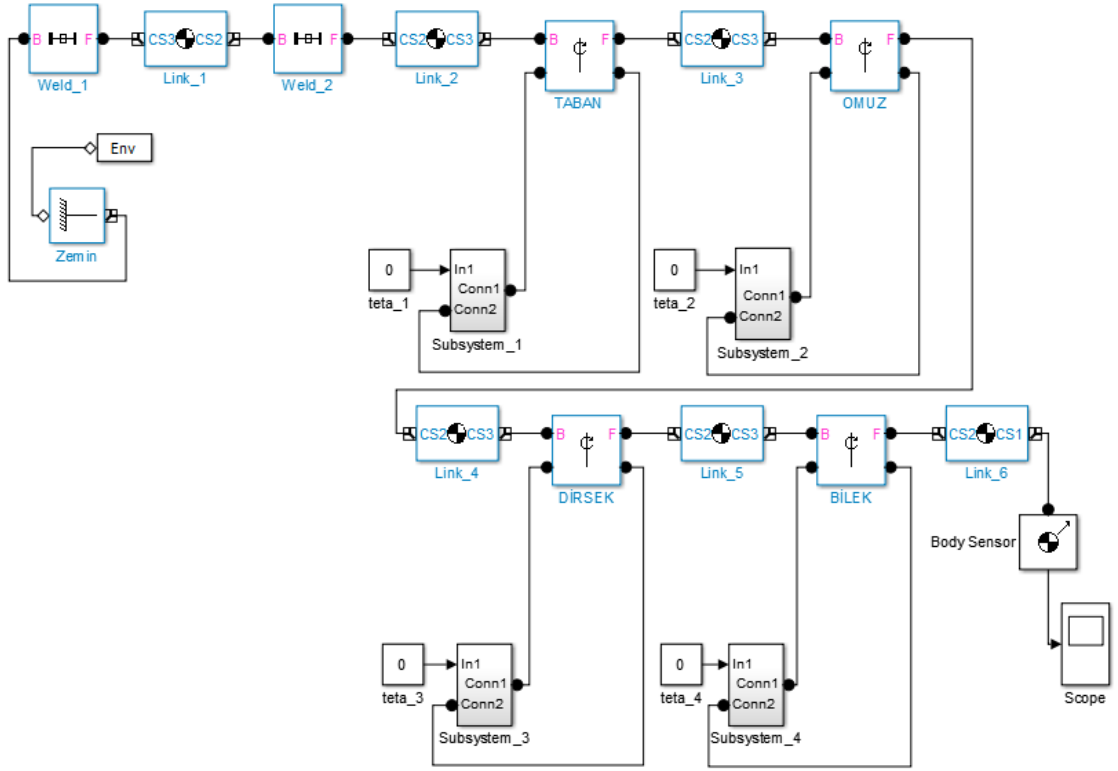
$$p_z = 66 - c_2(112s_3 + 95c_3s_4 + 95c_4s_3) - s_2(112c_3 + 95c_3c_4 - 95s_3s_4) - 89s_2 \quad (5.10)$$

Tablo 5.2’de örnek olarak girilen robot kol eklem açıları ( $\theta_1, \theta_2, \theta_3, \theta_4$ ) sonucunda hesaplanan tutucu uç koordinatları verilmiştir.

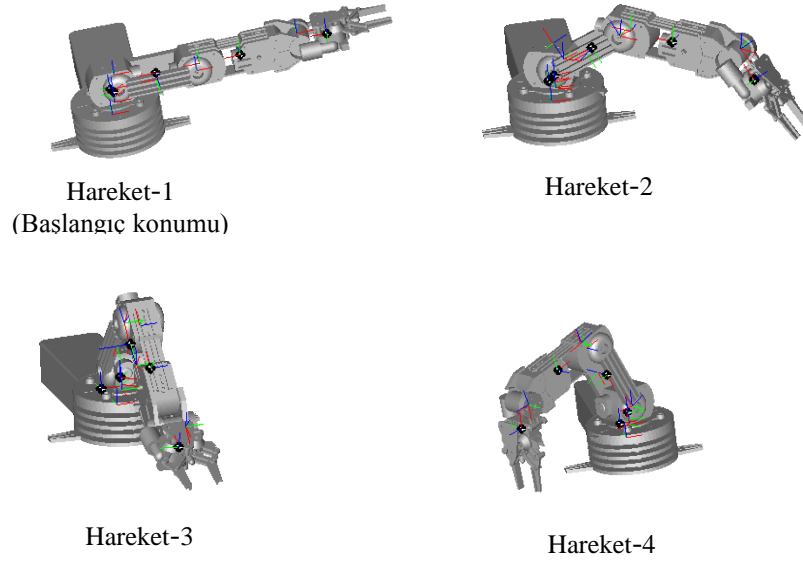
**Tablo 5.2.** OWI-535 robot kolunun 4 farklı hareketi için eklem açıları ve tutucu ucun konumu

Hareketler	Eklem Açıları (derece)				Tutucu uç konumu (mm) ( $p_x, p_y, p_z$ )
	Taban açısı ( $\theta_1$ )	Omuz açısı ( $\theta_2$ )	Dirsek açısı ( $\theta_3$ )	Bilek açısı ( $\theta_4$ )	
Hareket-1	0	0	0	0	(296, 0, 66)
Hareket-2	-30	-30	40	45	(209, -121, 13)
Hareket-3	-90	-50	100	-30	(0, -218, 16)
Hareket-4	-150	-65	80	80	(-119, -69, 23)

OWI-535 robot kol eklemlerinin istenilen açı değerlerine getirilerek robot kolun yapacağı hareketi benzetim yapabilmek için robot kol SolidWorks’te modellendikten sonra MATLAB/Simulink/SimMechanics araç kutusu aracılığıyla Simulink ortamında robot kolun mekanik modeli Şekil 5.19’da gösterildiği gibi oluşturulmuştur. Oluşturulan bu mekanik model sayesinde robot kol eklemlerinin Tablo 5.2’deki açılara göre aldığı son görüntüler Şekil 5.20’de gösterilmiştir.



Şekil 5.19. OWI-535 robot kol mekanik modelinin MATLAB/Simulink ile oluşturulan blok diyagramı



Şekil 5.20. Robot kolun 4 farklı hareketi

## 5.2.2. OWI-535 Robot Kol Ters Kinematik Analizi

OWI-535 robot kolunun düz kinematik analizi yapıldıktan sonra robot kol tutucu ucunun istenen koordinat değerlerine ulaşabilmesi için eklemlerinin bulunması gereken açı değerleri ters kinematik analiz ile elde edilen denklemlerin çözümü ile belirlenmektedir. Denklem çözümlerinde 4. Bölüm'de yapılan ara işlemleri kullanarak OWI-535 robot kolun eklem açılarını ifade eden eşitlikler elde edilmiştir.

OWI-535 robot kolunun ters kinematik analizi için öncelikle (5.11)'de verilen eşitlikteki temel dönüşü matrisinin elemanlarını ifade eden matris (5.12)'deki gibi yerine koyulup eşitlik sadeleştirilmiştir.

$$A_1^{-1}T_0^4 = A_1^{-1}A_1A_2A_3A_4 \quad (5.11)$$

$$A_1^{-1} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_2A_3A_4 \quad (5.12)$$

İleri yön (düz) kinematik analizde elde edilen homojen dönüşü matrislerini ve aşağıdaki trigonometrik kısaltmaları kullanarak (5.12) gerçekleştirilmiş ve (5.13) elde edilmiştir.

$$s_{234} = \sin(\theta_2 + \theta_3 + \theta_4), c_{234} = \cos(\theta_2 + \theta_3 + \theta_4), s_{23} = \sin(\theta_2 + \theta_3), c_{23} = \cos(\theta_2 + \theta_3)$$

$$\begin{bmatrix} n_x c_1 + n_y s_1 & o_x c_1 + o_y s_1 & a_x c_1 + a_y s_1 & p_x c_1 + p_y s_1 \\ -n_z & -o_z & -a_z & 66 - p_z \\ n_y c_1 - n_x s_1 & o_y c_1 - o_x s_1 & a_y c_1 - a_x s_1 & p_y c_1 - p_x s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{234} & -s_{234} & 0 & 95c_{234} + 112c_{23} + 89c_2 \\ s_{234} & c_{234} & 0 & 95s_{234} + 112s_{23} + 89s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.13)$$

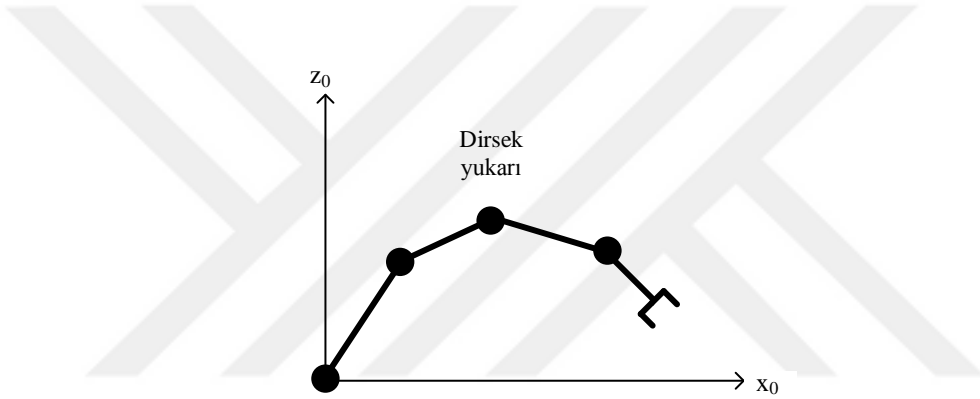
(5.13)'ten (5.14), (5.15) ve (5.16) eşitlikleri elde edilerek eklem açılarını ifade eden denklemler elde edilmiştir.

$$p_x c_1 + p_y s_1 = 95c_{234} + 112c_{23} + 89c_2 \quad (5.14)$$

$$66 - p_z = 95s_{234} + 112s_{23} + 89s_2 \quad (5.15)$$

$$p_y c_1 - p_x s_1 = 0 \quad (5.16)$$

Bu denklemlerin birden fazla çözüm kümesi olduğu için tez çalışmasında OWI-535 robot kolun hedefe uygun olan yönelimi Şekil 5.21'deki gibi belirlenmiş ve eklem açılarını ifade eden eşitlikler (5.17), (5.18), (5.19) ve (5.20)'de verildiği gibi tek çözüm kümesine düşürülerek kinematik analiz tamamlanmıştır.



Şekil 5.21. OWI-535 robot kolun hedefe uygun yönelimi

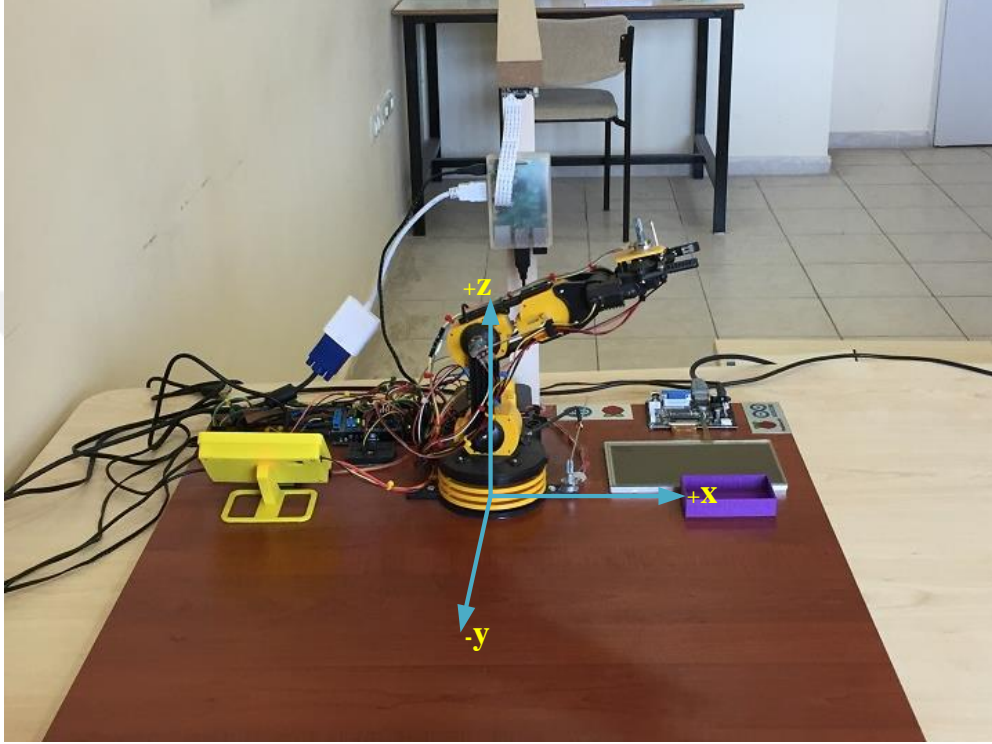
$$\theta_1 = \tan^{-1} \left( \frac{p_y}{p_x} \right) \quad (5.17)$$

$$\theta_3 = \tan^{-1} \left( \frac{\sqrt{1 - \left( \frac{((p_x c_1 + p_y s_1 - 95c_{234})^2 + (66 - p_z - 95s_{234})^2 - 89^2 - 112^2)}{(2 \times 89 \times 112)} \right)^2}}{((p_x c_1 + p_y s_1 - 95c_{234})^2 + (66 - p_z - 95s_{234})^2 - 89^2 - 112^2) / (2 \times 89 \times 112)} \right) \quad (5.18)$$

$$\theta_2 = \tan^{-1} \left( \frac{(89 + 112c_3)(66 - p_z - 95s_{234}) - 112s_3(p_x c_1 + p_y s_1 - 95c_{234})}{(89 + 112c_3)(p_x c_1 + p_y s_1 - 95c_{234}) + 112s_3(66 - p_z - 95s_{234})} \right) \quad (5.19)$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (5.20)$$

Şekil 5.22’de ana koordinat sistemi gösterilen OWI-535 robot kol eklemlerinin açı sınırları göz önünde bulundurularak tutucu ucunun istenen bir  $p_x$ ,  $p_y$ ,  $p_z$  koordinatına gitmesi için  $\theta_{234} = 40$  kabul edilerek örnek 4 hareket için kol eklemlerinin alması gereken açı değerleri Tablo 5.3’te verilmiştir.

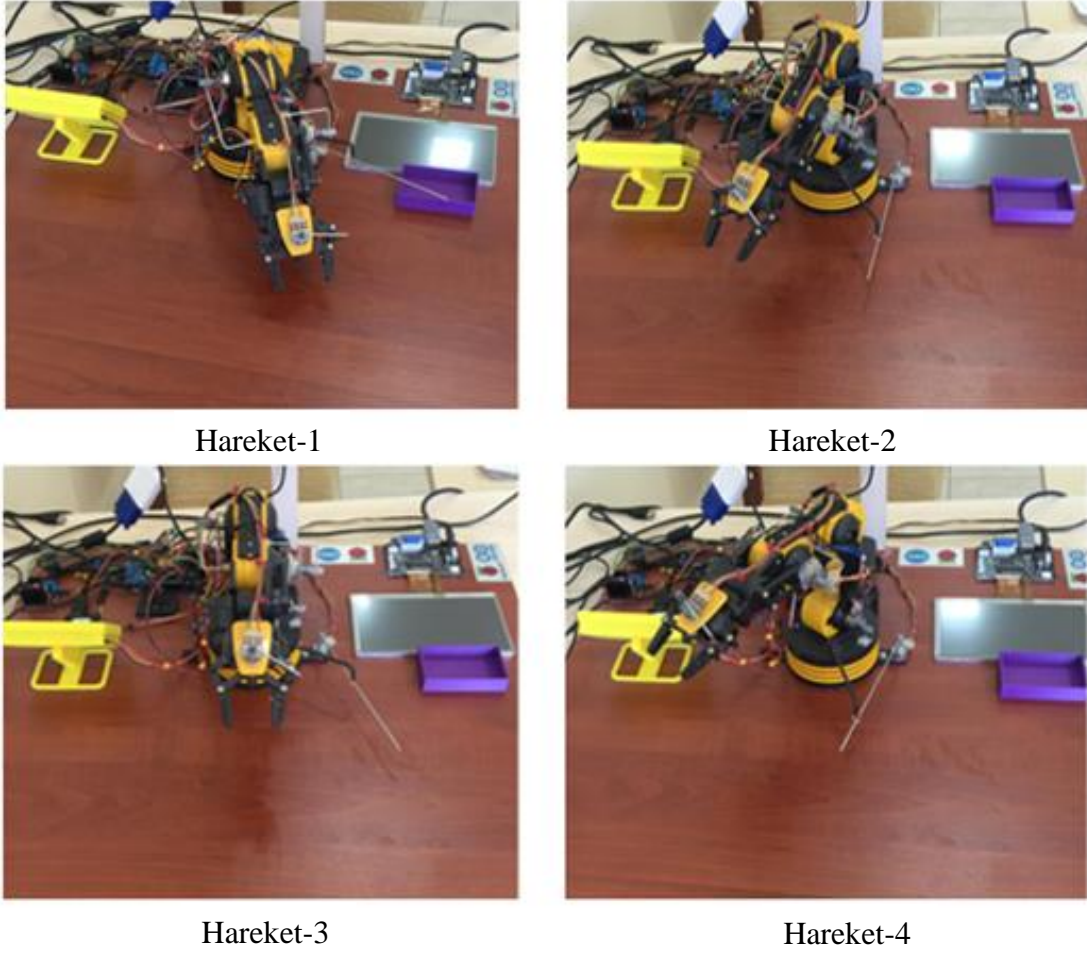


Şekil 5.22. Robot kol ana koordinat sistemi

Tablo 5.3. OWI-535 robot kol tutucu ucunun bulunduğu konuma göre eklemlerinin açıları

Hareket	Tutucu uç konumu (mm) ( $p_x, p_y, p_z$ )	Eklemlerinin Açılımları (derece)			
		Taban açısı ( $\theta_1$ )	Omuz açısı ( $\theta_2$ )	Dirsek açısı ( $\theta_3$ )	Bilek açısı ( $\theta_4$ )
1	(100,-200,30)	-63.43	-55.87	81.59	14.27
2	(-80,-150,50)	-118.07	-93.78	116.77	17.01
3	(0,-170,80)	-90	-99.07	105.64	33.42
4	(-120,-160,70)	-126.86	-78.69	90.12	28.56

Tablo 5.3'e göre gerçekleştirilen bu 4 hareket sonucunda, robot kolun aldığı pozisyon görünümleri Şekil 5.23'te verilmiştir.

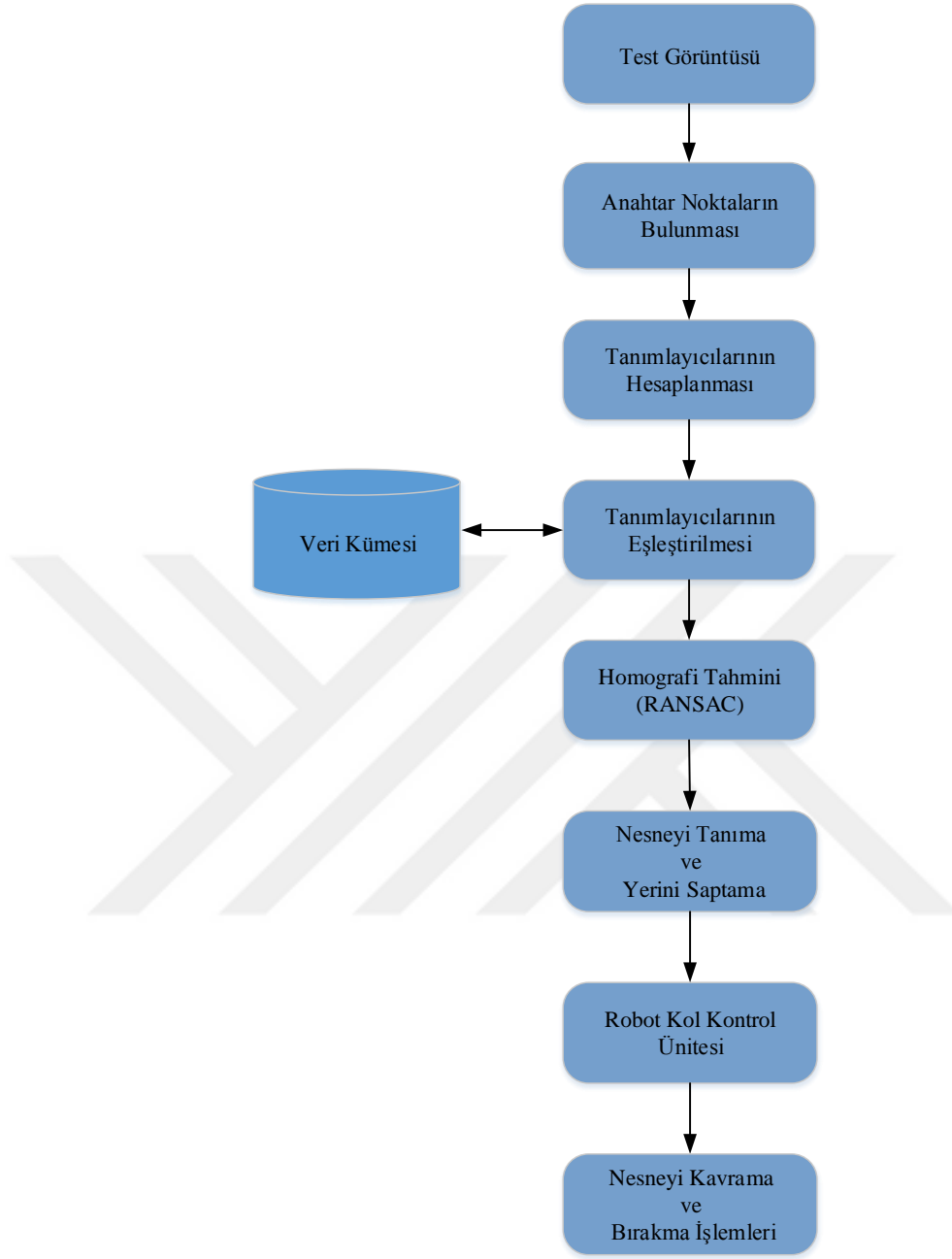


Şekil 5.23. 4 farklı hareket sonucu robot kolun görünümü

### 5.3. Deneysel Çalışmalar

Tez kapsamındaki deneysel çalışmalarda, bilimsel yazında en iyi bilinen anahtar nokta bulucu ve anahtar nokta tanımlayıcılarını hesaplayan algoritmaların performansları karşılaştırılarak nesne tanıma ve yerini tespit işleminden sonra robot kolun tutucusu ile nesnenin kavranma işlemleri gerçekleştirilmiştir. Yapılan deneysel çalışmalar için Şekil 5.24'te verilen işlem adımları gerçekleştirilmiştir.





Şekil 5.24. Deneysel çalışmalar için işlem adımları

Şekil 5.24'teki işlem adımlarından robot kol kontrol ünitesi adımına kadar olan tüm adımlarda, 700 MHz tek çekirdekli işlemci ve 512 MB RAM'e sahip Raspberry Pi B+ devre kartı kullanılmıştır. Bu adımlar, kart üzerinde çalışan OpenCV 2.4.9. kütüphanesi kullanılarak C++ dilinde yazılan yazılımlar ile gerçekleştirilmiştir.

Nesnenin tanınması ve yer tespiti için oluşturulan sistemin eğitimi için öncelikle öznelik vektörlerini içeren veri kümesi oluşturulur. Daha sonra, OWI-535 robot kolunun çalışma alanını gösterecek ve robot kolun çarpmayacağı şekilde yerleştirilen kameradan test

görüntüsü alınıp gri seviyeye çevrildikten sonra nesne bulma ve tanıma algoritmaları kullanılıp görüntüye ait anahtar noktalar tespit edilir. Tespit edilen anahtar noktaların tanımlayıcıları hesaplandıktan sonra daha önceden oluşturulan veri kümesindeki öznitelikler ile eşleştirilerek en fazla eşleşmenin olduğu nesne görüntüsü bulunup nesnenin ait olduğu sınıf belirlenir. RANSAC algoritması ile en fazla eşleşmenin olduğu nesne ile test görüntüsü arasında eğer varsa aykırı eşleşmeler de elenerek nesne sınırlayıcı kutu içine alınarak nesnenin ağırlık merkezi piksel cinsinden bulunur. Daha sonra, robot kol tutucusunun yönelimi için gerekli olan x ve y koordinatlarına dönüştürülerek Raspberry Pi devre kartı üzerinden, robot kolun kontrolünün gerçekleştirildiği Arduino Mega 2560 kartına seri haberleşme ile gönderilir. Bu Arduino kartındaki yazılım ile de robot kol tutucu ucunun belirlenen koordinat değerlerine hareketini gerçekleştirebilmesi için her bir eklemının alması gereken açılar ters kinematik analiz eşitlikleri kullanılıp hesaplanarak robot kolun nesneyi kavrayıp nesne bırakma haznesine bırakması sağlanır.

### **5.3.1. Nesne Tanıma ve Yer Tespiti Çalışmaları**

Gerçekleştirilen nesne tanıma işlemi için yapılan deneysel çalışmalarda görüntülerdeki öznitelikleri çıkarmak için SIFT, SURF, FAST ve ORB algoritmaları kullanılmıştır. SIFT, SURF ve ORB algoritmalarının hem anahtar nokta tespiti hem de anahtar noktalarının tanımlayıcılarını hesaplayabildikleri için tek başına kullanılmışlardır. Ancak, FAST algoritmasının kendi anahtar nokta tanımlayıcısı bulunmadığından SURF algoritması ile tanımlayıcıları hesaplanarak kullanılmıştır. Anahtar noktaların tespitinde çalışmaya uygun olarak OpenCV kütüphanesindeki algoritma parametreleri ayarlanmıştır. SIFT algoritması için oktav katmanı 3 ve kontrast değeri 0.04, SURF algoritması için oktav katmanı 2 ve Hessian sınır değeri 200, FAST algoritması için merkez piksel ve bu piksel etrafındaki çemberde yer alan piksellerin parlaklık değerlerinin farkı için sınır değeri 20 olarak belirlenmiştir. ORB algoritması için ise, tespit edilmesi istenen en fazla anahtar nokta sayısı 500 olarak belirlenmiştir.

#### **5.3.1.1. Veri Kümesinin Oluşturulması**

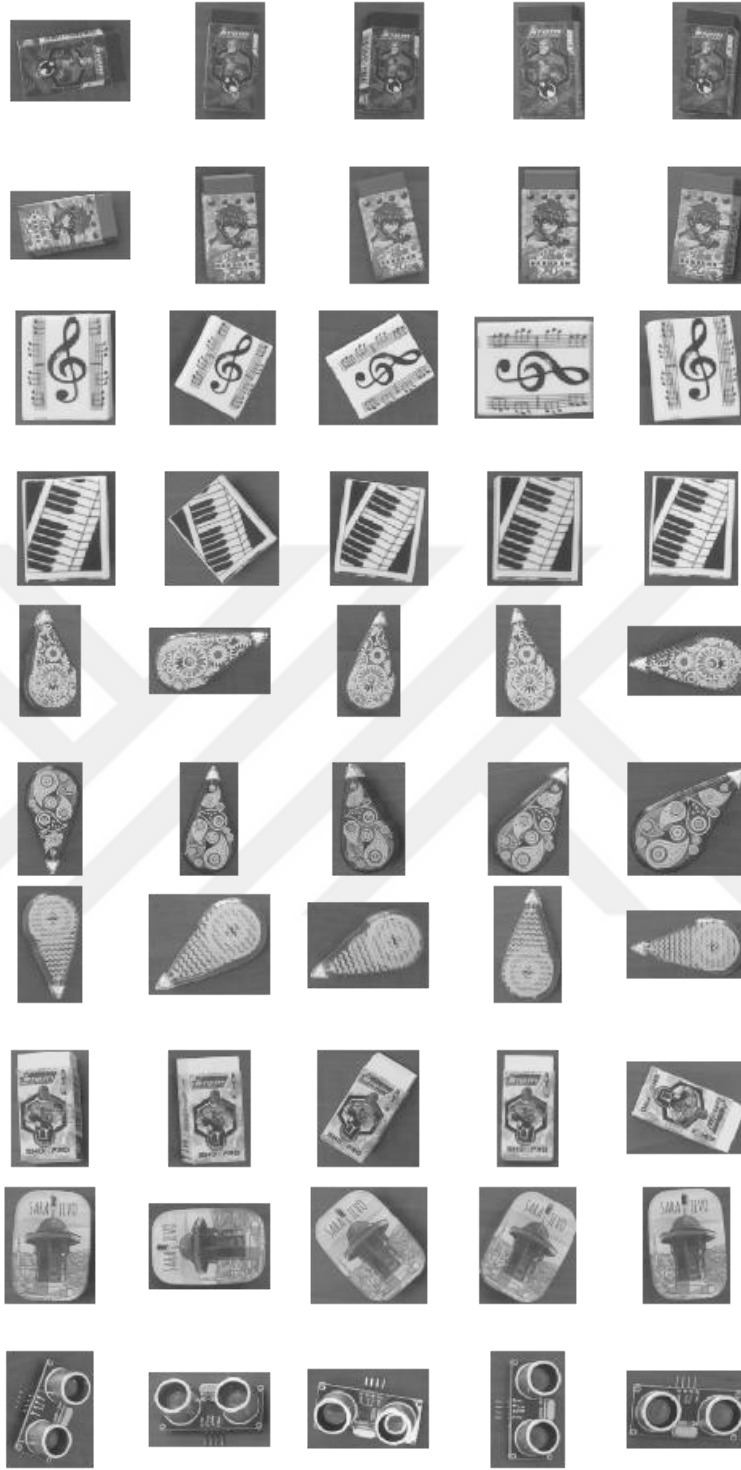
Deneysel çalışmalar için sistemin eğitim aşamasında kullanılan nesnelere, robot kol tutucu ucunun kavrayabileceği boyutta nesnelere olduğu için veri kümesi, küçük boyuttaki nesne

görüntülerinin öznitelik vektörlerini içermektedir. Ayrıca, kameranın iki boyutlu olmasından dolayı nesnelerin yükseklikleri birbirine çok yakın seçilerek tutucu ucun sabit z konum koordinatında ( $p_z$ ) kavrama işlemini gerçekleştirmesi sağlanmıştır. Robot kol çalışma alanına uygun olarak rastgele yerleştirilen nesnelerin kamera ile 1024x768 çözünürlükte RGB olarak görüntüleri alındıktan sonra, sadece nesnelere ait olan görüntüler kırpılmıştır. Görüntülerin yüksek çözünürlükte alınmasının sebebi, nesnelerin boyutlarının küçük olmasıdır. Bu nesnelerin görüntülerinin düşük çözünürlük ile alınması durumunda nesne görüntüsü üzerindeki anahtar noktaların tespiti zorlaşmakta ve sayısı azalmaktadır. Şekil 5.25'te verildiği gibi veri kümesi 10 nesne ile oluşturulmuştur. Deneysel çalışmalarda, kamera görüş açısı, aydınlanma, dönme ve ölçek değişiklikleri gibi zor şartları da içermek için her bir sınıftan 5 farklı nesne görüntüsü kullanılmıştır.



Şekil 5.25. Veri kümesi için kullanılan nesnelerin RGB görüntüleri

Görüntüler RGB olarak alındıktan sonra, nesne bulma ve tanıma algoritmalarını kullanarak özniteliklerini çıkarabilmek için Şekil 5.26'da verildiği gibi gri seviye görüntülere dönüştürülmüştür.

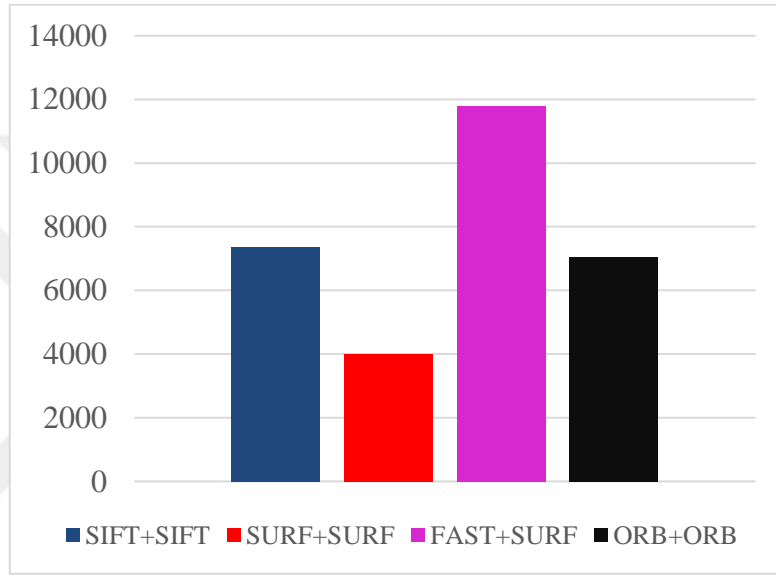


Şekil 5.26. Veri kümesi için kullanılan nesnelerin gri seviye görüntüleri

Şekil 5.26’da yer alan nesne sınıflarının etiketleri sırasıyla *mavi karakterli silgi*, *kırmızı karakterli silgi*, *nota sembolü silgi*, *piyano sembolü silgi*,  *pembe daksil*, *mavi daksil*,  *yeşil daksil*,  *beyaz karakterli silgi*,  *Sarajevo maskotu* ve  *ultrasonik sensör* olarak verilmiştir.

Toplamda 50 olan nesne görüntüsünün tamamının anahtar noktaları tespit edilip tanımlayıcıları hesaplanmıştır. Bu tanımlayıcılar sınıf etiketleriyle birlikte saklanarak eğitim aşaması için gerekli olan öznelik veri kümesi elde edilmiştir.

Anahtar nokta bulucu ve anahtar nokta tanımlayıcı algoritmalar; SIFT ile SIFT, SURF ile SURF, FAST ile SURF ve ORB ile ORB şeklinde birlikte kullanılarak Şekil 5.26'daki eğitim görüntülerine uygulanmıştır. Bu algoritmalar ile elde edilen veri kümelerinin toplam öznelik vektörlerinin sayıları Şekil 5.27'de verilmiştir.



Şekil 5.27. Farklı algoritmalar ile çıkarılan öznelik vektörlerinin toplam sayıları

### 5.3.1.2. Test İşlemleri

Veri kümeleri elde edilip eğitim aşaması tamamlandıktan sonra sisteme dışardan verilen test görüntüsünde yer alan nesneyi tanımak ve tanınan nesneyi sınırlayıcı kutu içerisine almak için çeşitli test görüntüleri üzerinde çalışmalar yapılmıştır.

Deneysel çalışmalarda kullanılan test görüntüleri, hem eğitimde kullanılan ve hem de farklı nesnelere içerisinde barındırmaktadır. Bu görüntülerin her birinde tanınması istenen bir nesne bulunmakta olup her bir nesne sınıfı için ikişer adet test görüntüsü olmak üzere toplam 20 adet test görüntüsü alınmıştır. 1024x768 çözünürlükte RGB olarak alınan bu test görüntüleri gri seviye dönüştürülerek test işlemi için hazır hale getirilmiştir. Test görüntülerinden bazılarını Şekil 5.28, Şekil 5.29 ve Şekil 5.30'da yer verilmiştir.



Şekil 5.28. Örnek test görüntüsü-1



Şekil 5.29. Örnek test görüntüsü-2



Şekil 5.30. Örnek test görüntüsü-3

Test görüntüleri belirlendikten sonra herhangi bir test görüntüsü içerisindeki nesnenin tanınması için öncelikle test görüntüsünün anahtar noktaları tespit edilmiş ve her bir anahtar noktaya ait tanımlayıcılar hesaplanmıştır. Bu işlemler, veri kümesi oluştururken kullanılan SIFT+SIFT, SURF+SURF, FAST+SURF ve ORB+ORB algoritmaları kullanılarak gerçekleştirilmiştir. Test görüntüsünün anahtar nokta tanımlayıcıları hesaplandıktan sonra daha önce oluşturulan veri kümesindeki tanımlayıcılar ile FLANN tabanlı eşleştirici ve BF eşleştiricileri kullanılarak eşleştirilmiştir.

SIFT ve SURF sonucunda elde edilen tanımlayıcıların eşleştirilmesinde FLANN tabanlı eşleştirici, ORB tanımlayıcılarının eşleştirilmesinde BF eşleştiriciler kullanılmıştır. ORB ile hesaplanan ikili tanımlayıcılarının eşleştirilmesinde Hamming metriği kullanılmıştır.

20 test görüntüsünün her biri için anahtar nokta tanımlayıcıları elde edilip, veri kümesindeki tanımlayıcılarla eşleştirilmiştir. Eşleştirilen tanımlayıcılar arasındaki uzaklıklar hesaplandıktan sonra en küçük uzaklık değeri ile belli bir eşik değeri mesafesi belirlenmiştir. Eşik değeri altında kalan uzaklıklara sahip olan eşleşmeler en yakın eşleşmeleri temsil etmekte olup en fazla eşleşmeye sahip olan nesne görüntüsü test görüntüsünün sınıfı olarak belirlenmiştir. Nesne sınıfı belirlendikten sonra RANSAC algoritmasıyla da test görüntüsü ile nesne görüntüsü arasında aykırı eşleşmeler varsa elenip, nesne sınırlayıcı kutu içerisine alınmıştır. Tanınan nesnenin sınıf etiketi görüntü üzerine



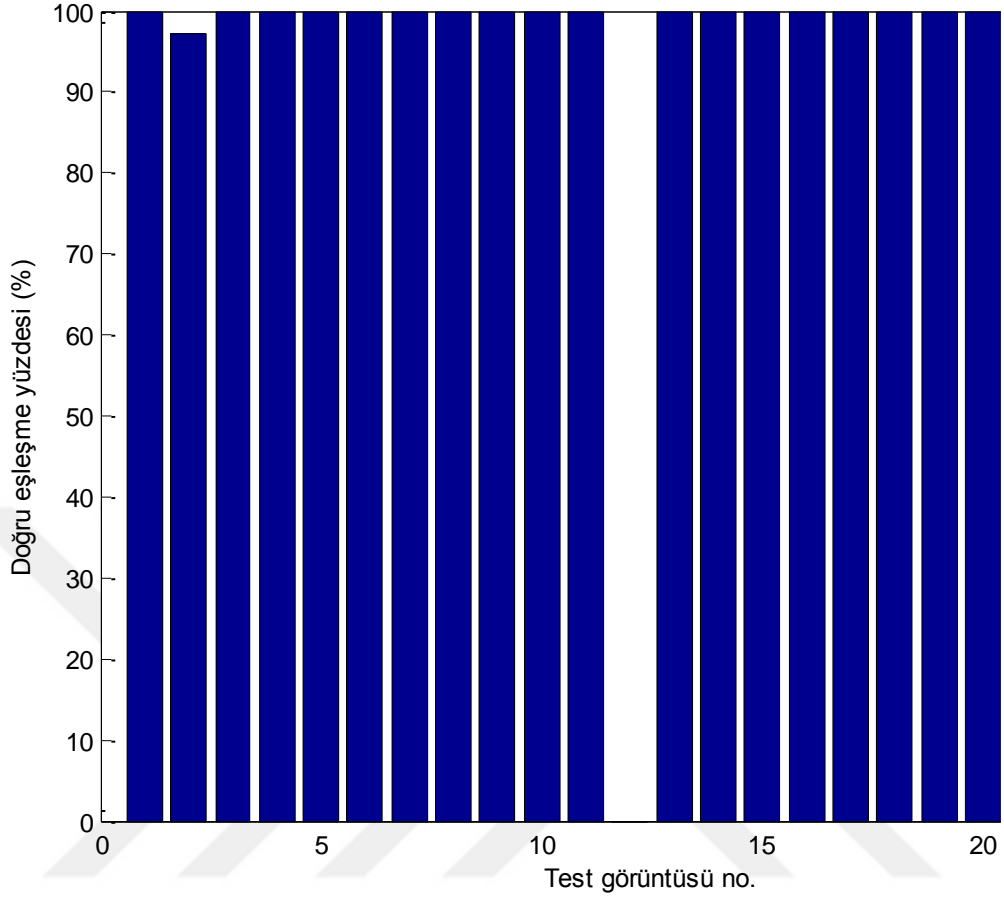
yazılıp, sınırlayıcı kutunun ağırlık merkezinin de belirlenmesiyle nesne tanıma ve yer tespiti işlemleri tamamlanmıştır.

SIFT+SIFT algoritmaları kullanılarak gerçekleştirilen test işlemleriyle elde edilen sonuçlar Tablo 5.4'te verilmiştir.

**Tablo 5.4.** Test işlemleri (SIFT+SIFT)

Test görüntüsü no.	Anahtar nokta sayısı	Öznitelik çıkarımı için geçen süre (ms)	Eşleştiği nesne görüntüsünün anahtar nokta sayısı	Eşleşen tanımlayıcı çifti sayısı	Aykırı eşleşme sayısı
1	460	23283.3	321	95	0
2	603	23778.0	281	104	3
3	518	23704.9	166	52	0
4	369	22955.6	166	69	0
5	539	23634.5	281	144	0
6	541	24063.4	281	82	0
7	270	22623.0	112	25	0
8	371	22897.9	112	39	0
9	473	23363.8	125	19	0
10	399	23055.7	117	35	0
11	219	22335.3	71	6	0
12	403	22994.3	100	36	36
13	460	23393.2	178	69	0
14	359	22803.1	178	53	0
15	337	22725.9	101	24	0
16	332	22518.1	97	37	0
17	308	22568.7	84	36	0
18	380	23111.5	84	32	0
19	345	22794.3	78	8	0
20	221	22347.5	80	11	0

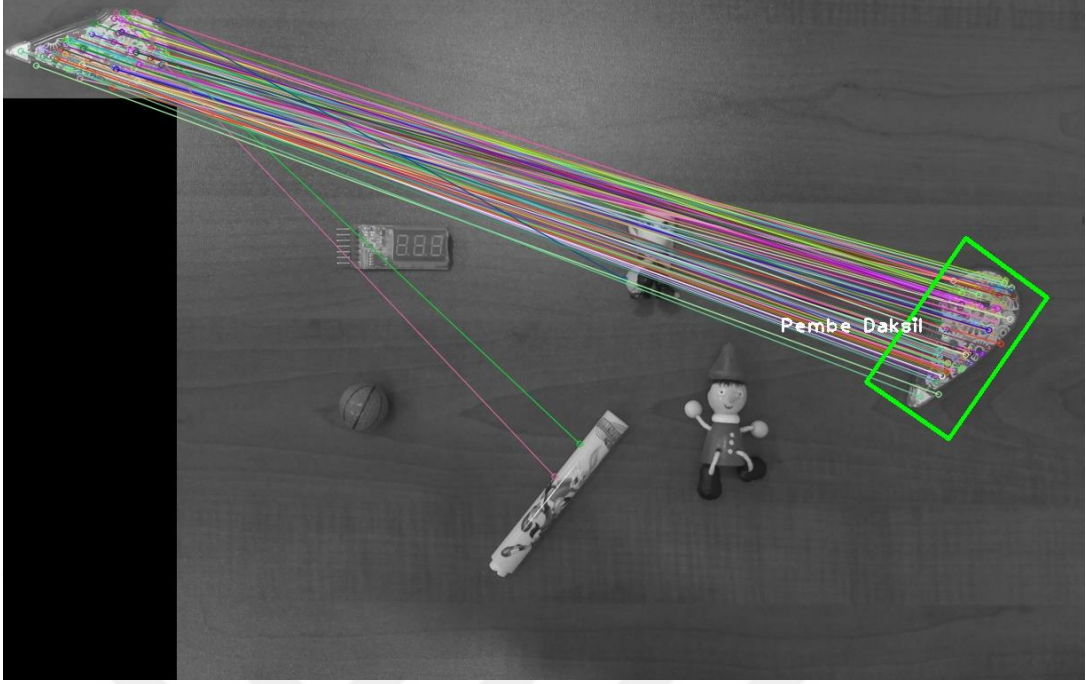
Tablo 5.4'teki sonuçlara göre 20 test görüntüsünün 19'u doğru sınıflandırılmış olup sadece 12 no'lu test görüntüsündeki nesnenin yanlış sınıflandırıldığı, bunun için de tamamen aykırı eşleşmelerin meydana geldiği görülmüştür. Test görüntülerinin her biri için doğru eşleşme yüzdesi Şekil 5.31'de verilmiştir.



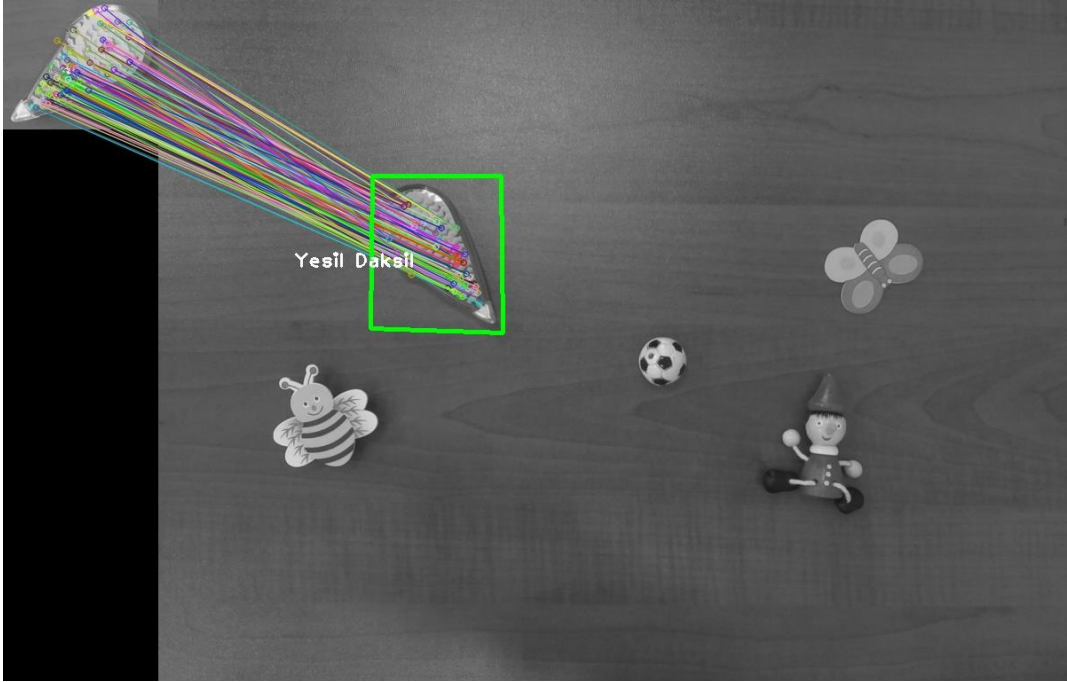
**Şekil 5.31.** Test görüntülerinin doğru eşleşme yüzdeleri (SIFT+SIFT)

Şekil 5.31'e göre SIFT+SIFT algoritması ile elde edilen tanımlayıcıların doğru eşleşme yüzdesi 12 no'lu test görüntüsünün tamamen yanlış eşleşmesi hariç diğer görüntülerde %97 ile 100 arasındadır.

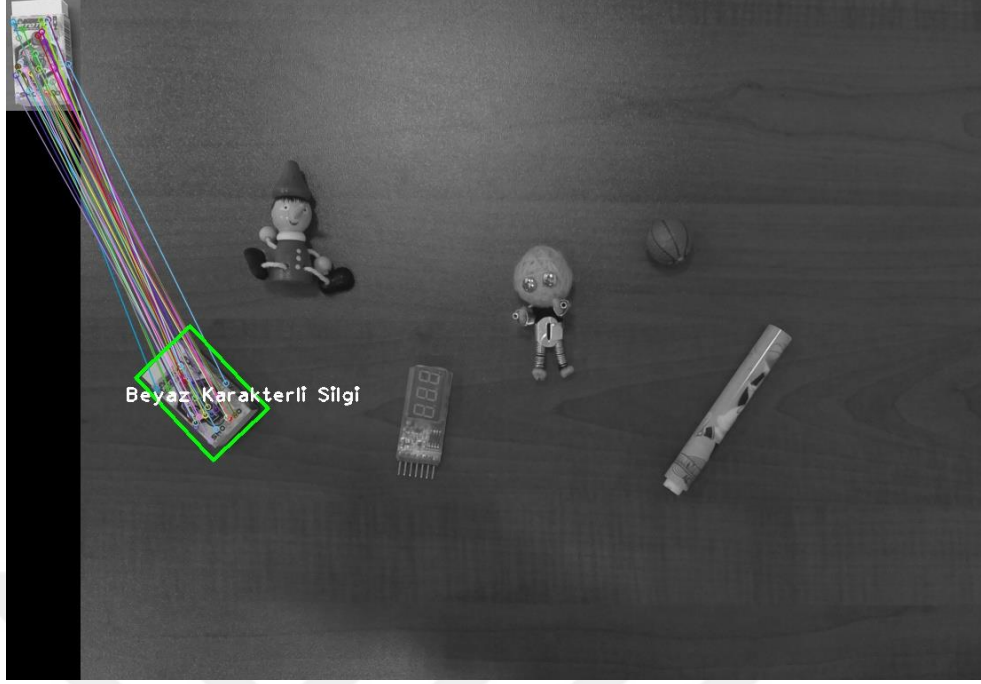
Gerçekleştirilen test işlemlerinden bazılarında Şekil 5.32, Şekil 5.33, Şekil 5.34 ve Şekil 5.35'te yer verilmiştir.



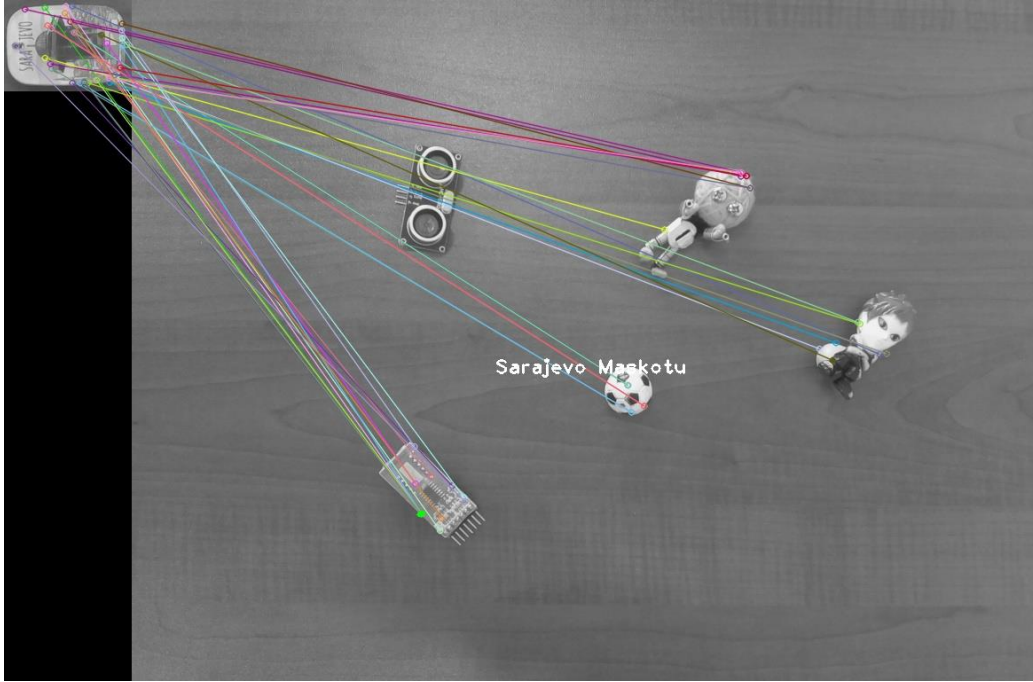
Şekil 5.32. 2 no'lu test görüntüsü için test işlemi (SIFT+SIFT)



Şekil 5.33. 5 no'lu test görüntüsü için test işlemi (SIFT+SIFT)



Şekil 5.34. 10 no'lu test görüntüsü için test işlemi (SIFT+SIFT)



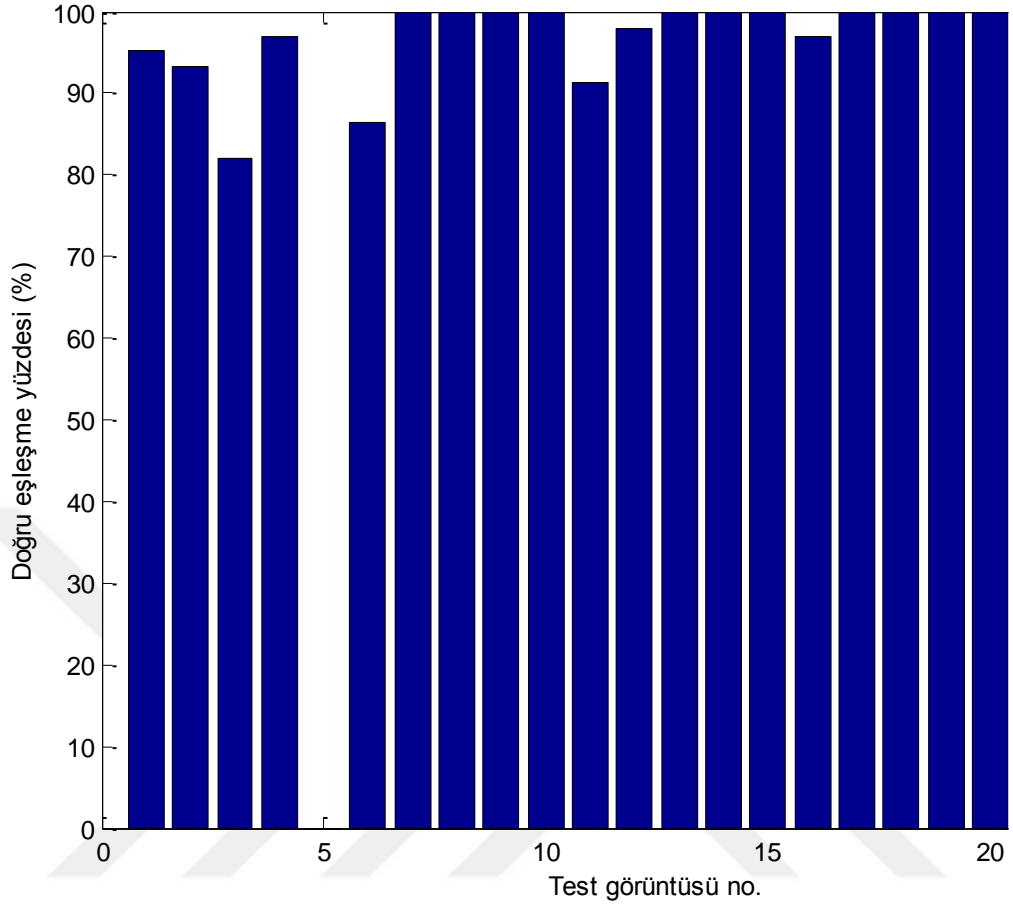
Şekil 5.35. 12 no'lu test görüntüsü için test işlemi (SIFT+SIFT)

SURF+SURF algoritmaları kullanılarak gerçekleştirilen test işlemleriyle elde edilen sonuçlar Tablo 5.5'te verilmiştir.

**Tablo 5.5.** Test işlemleri (SURF+SURF)

Test görüntüsü no.	Anahtar nokta sayısı	Öznitelik çıkarımı için geçen süre (ms)	Eşleştiği nesne görüntüsünün anahtar nokta sayısı	Eşleşen tanımlayıcı çifti sayısı	Aykırı eşleşme sayısı
1	306	3928.2	134	129	6
2	376	4705.5	129	45	3
3	459	4853.5	109	50	9
4	301	4155.0	124	32	1
5	327	4174.6	134	134	134
6	329	4426.0	97	88	12
7	258	3877.5	82	31	0
8	320	4405.1	76	25	0
9	375	4630.6	73	12	0
10	354	4605.0	67	22	0
11	253	3687.4	65	46	4
12	344	4785.4	65	51	1
13	334	4247.0	64	24	0
14	279	4020.8	77	18	0
15	311	4353.9	83	32	0
16	301	4757.5	60	32	1
17	308	4232.0	87	38	0
18	331	4301.8	98	25	0
19	283	4008.5	45	13	0
20	179	3337.9	47	9	0

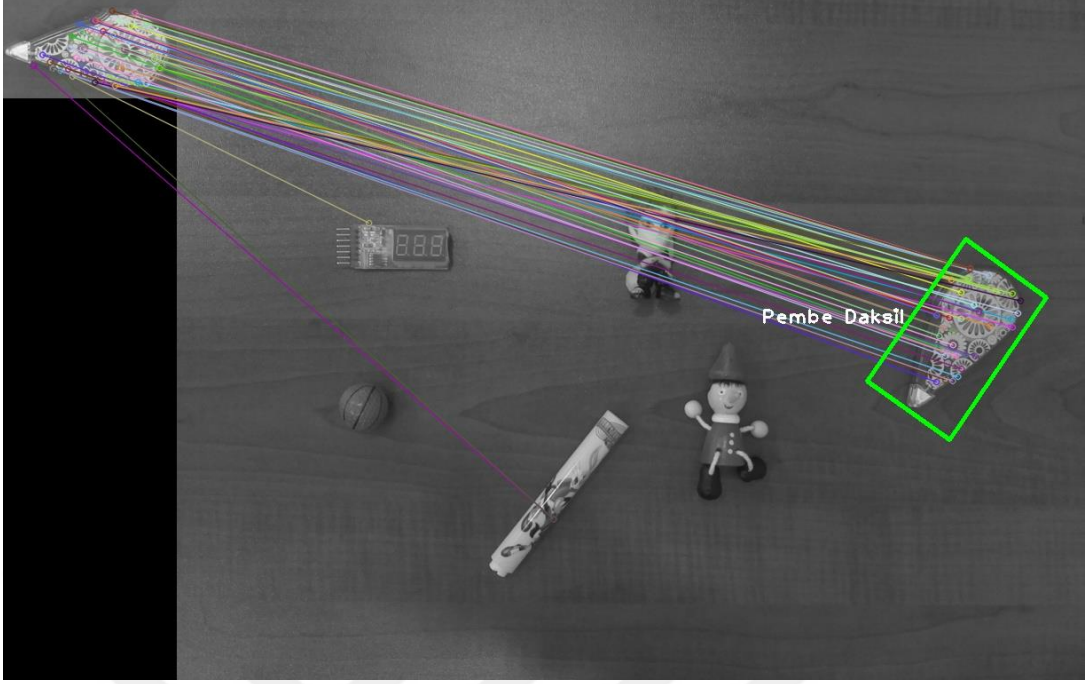
Tablo 5.5'teki sonuçlara göre 20 test görüntüsünün 19'u doğru sınıflandırılmış olup sadece 5 no'lu test görüntüsündeki nesnenin yanlış sınıflandırıldığı, bunun için de tamamen aykırı eşleşmelerin meydana geldiği görülmüştür. Test görüntülerinin her biri için doğru eşleşme yüzdesi Şekil 5.36'da verilmiştir.



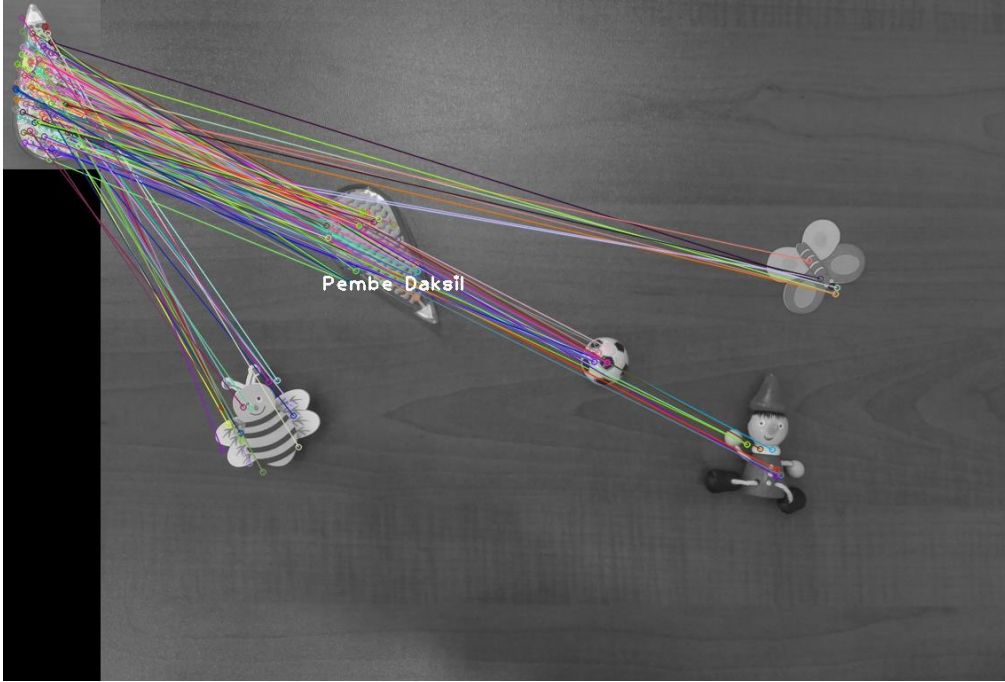
**Şekil 5.36.** Test görüntülerinin doğru eşleşme yüzdeleri (SURF+SURF)

Şekil 5.36'ya göre SURF+SURF algoritması ile elde edilen tanımlayıcıların doğru eşleşme yüzdesi 5 no'lu test görüntüsünün tamamen yanlış eşleşmesi hariç diğer görüntülerde %82 ile %100 arasındadır.

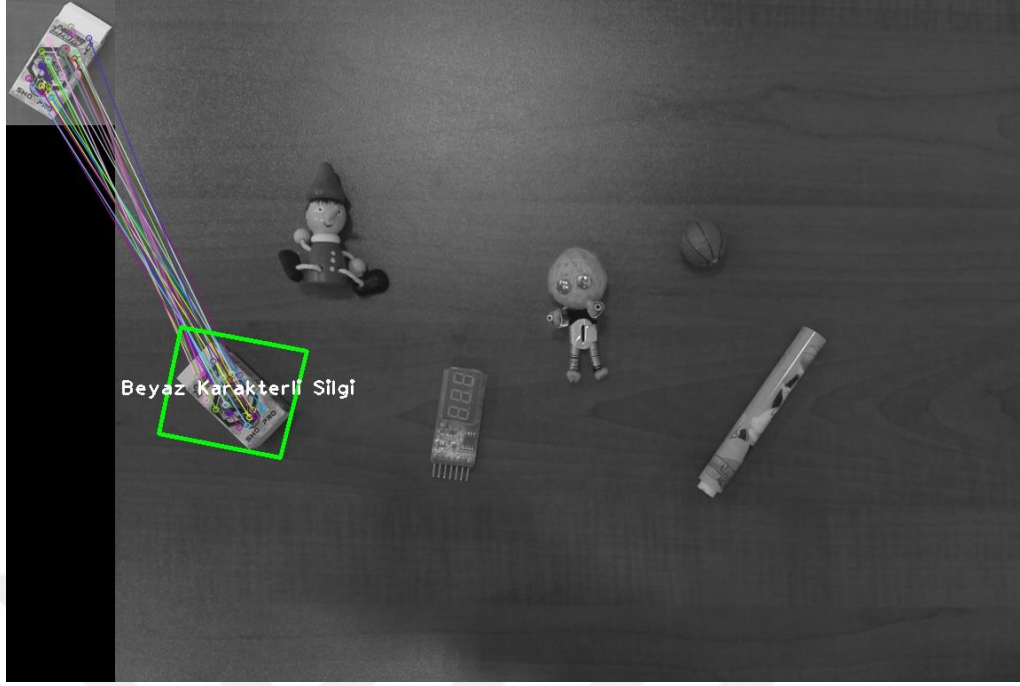
Gerçekleştirilen test işlemlerinden bazılarında Şekil 5.37, Şekil 5.38, Şekil 5.39 ve Şekil 5.40'ta yer verilmiştir.



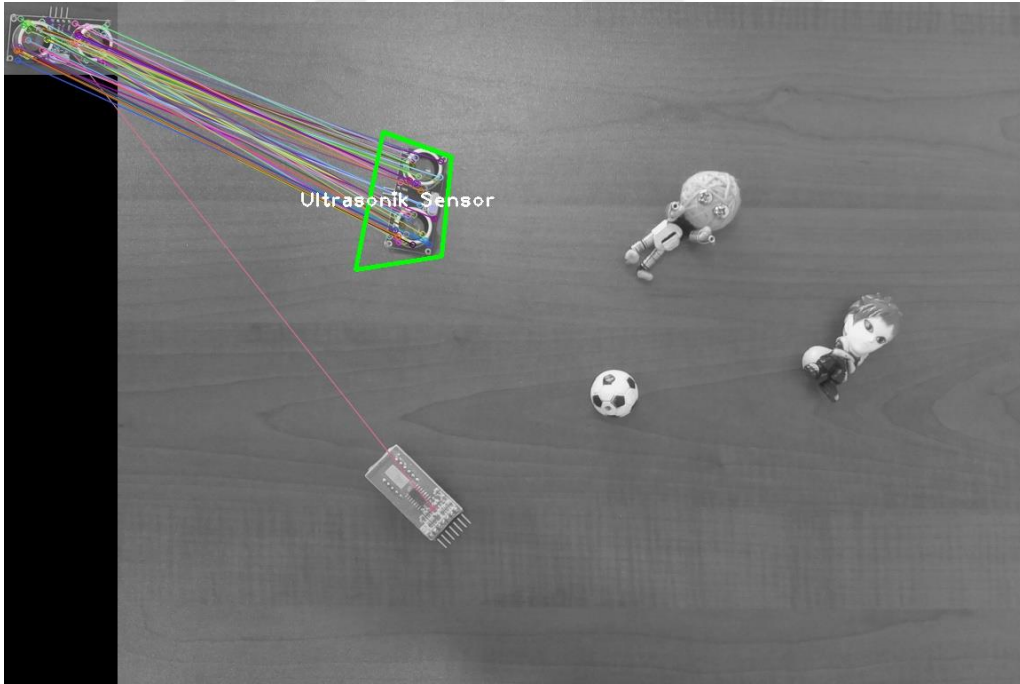
Şekil 5.37. 2 no'lu test görüntüsü için test işlemi (SURF+SURF)



Şekil 5.38. 5 no'lu test görüntüsü için test işlemi (SURF+SURF)



Şekil 5.39. 10 no'lu test görüntüsü için test işlemi (SURF+SURF)



Şekil 5.40. 12 no'lu test görüntüsü için test işlemi (SURF+SURF)

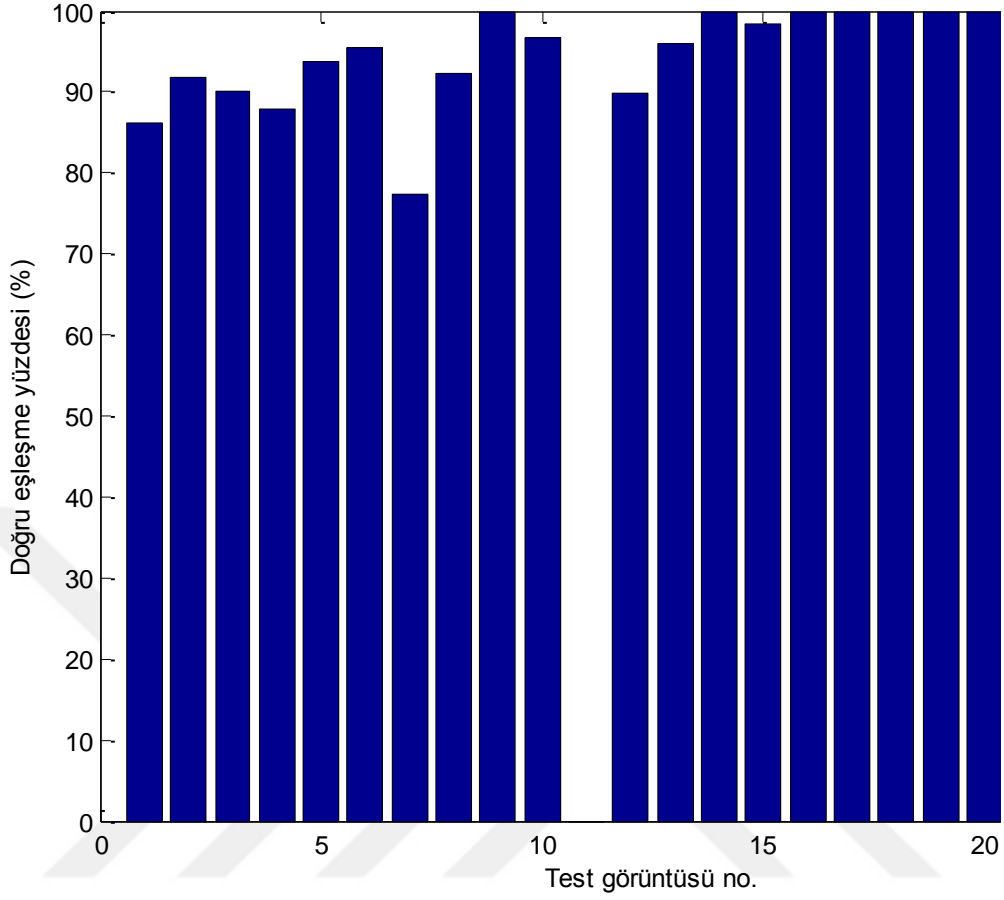
FAST+SURF algoritmaları kullanılarak gerçekleştirilen test işlemleriyle elde edilen sonuçlar Tablo 5.6'da verilmiştir.



**Tablo 5.6.** Test işlemleri (FAST+SURF)

Test görüntüsü no.	Anahtar nokta sayısı	Öznitelik çıkarımı için geçen süre (ms)	Eşleştiği nesne görüntüsünün anahtar nokta sayısı	Eşleşen tanımlayıcı çifti sayısı	Aykırı eşleşme sayısı
1	682	515.1	363	233	32
2	797	578.7	414	145	12
3	758	610.0	282	140	14
4	580	456.6	282	90	11
5	687	545.9	315	162	10
6	742	576.8	315	152	7
7	428	418.6	165	53	12
8	492	419.5	163	26	2
9	738	536.4	213	16	0
10	612	457.0	213	31	1
11	472	380.1	273	152	152
12	658	501.2	146	69	7
13	726	531.6	222	49	2
14	569	436.1	222	45	0
15	610	472.7	197	66	1
16	539	445.1	174	34	0
17	611	516.7	242	31	0
18	616	518.8	242	14	0
19	510	394.1	170	23	0
20	387	318.4	142	40	0

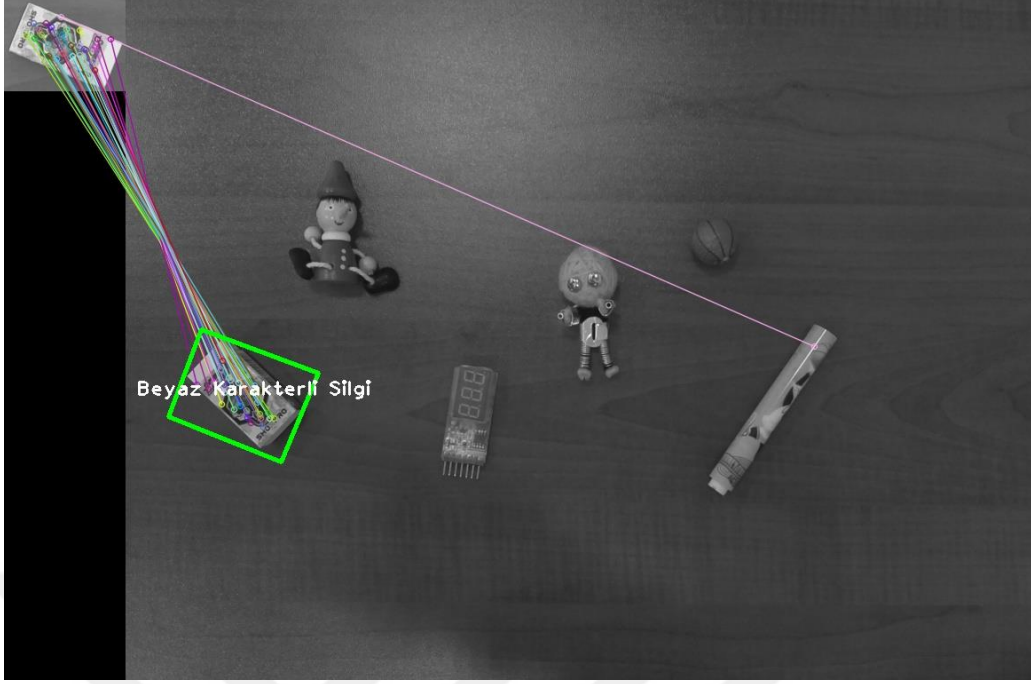
Tablo 5.6'daki sonuçlara göre 20 test görüntüsünün 19'u doğru sınıflandırılmış olup sadece 11 no'lu test görüntüsündeki nesnenin yanlış sınıflandırıldığı, bunun için de tamamen aykırı eşleşmelerin meydana geldiği görülmüştür. Test görüntülerinin her biri için doğru eşleşme yüzdesi Şekil 5.41'de verilmiştir.



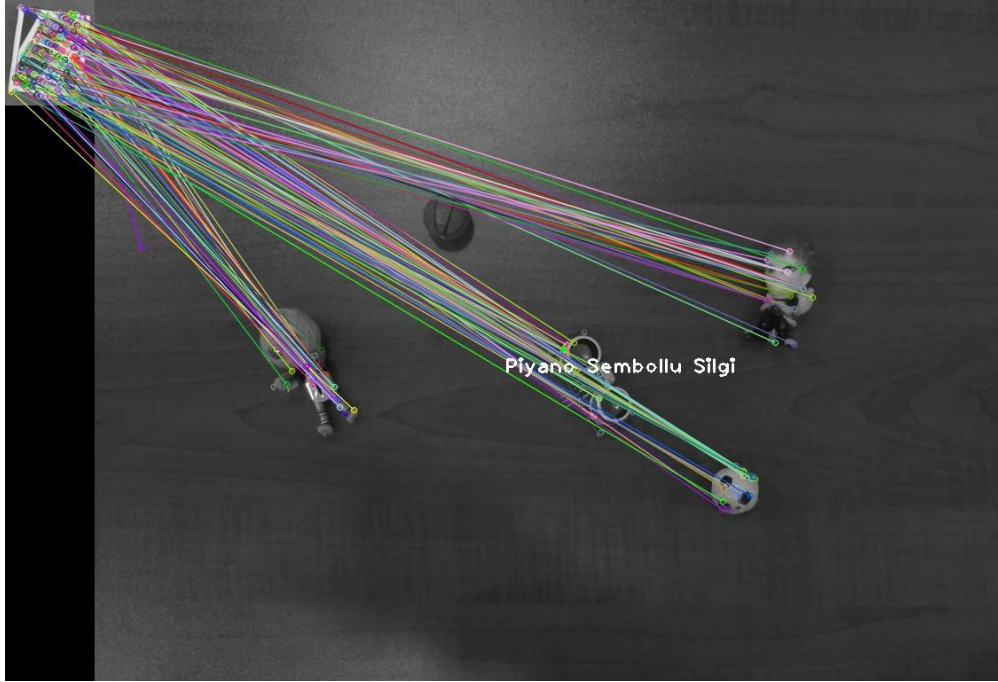
**Şekil 5.41.** Test görüntülerinin doğru eşleşme yüzdeleri (FAST+SURF)

Şekil 5.41'e göre FAST+SURF algoritması ile elde edilen tanımlayıcıların doğru eşleşme yüzdesi 11 no'lu test görüntüsünün tamamen yanlış eşleşmesi hariç diğer görüntülerde %77 ile %100 arasındadır.

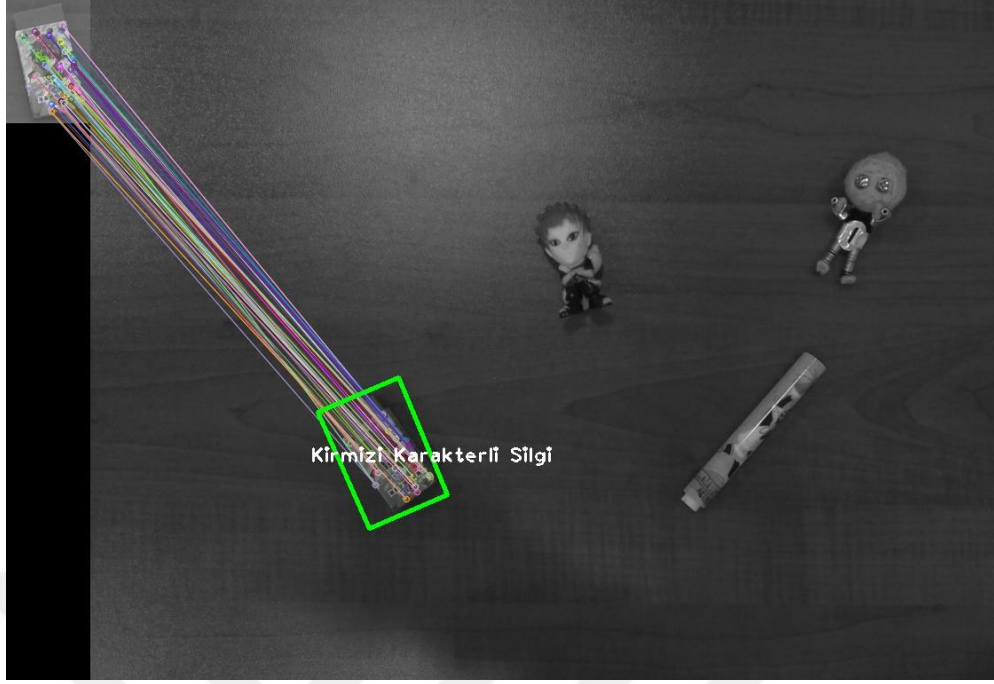
Gerçekleştirilen test işlemlerinden bazılarında Şekil 5.42, Şekil 5.43, Şekil 5.44 ve Şekil 5.45'te yer verilmiştir.



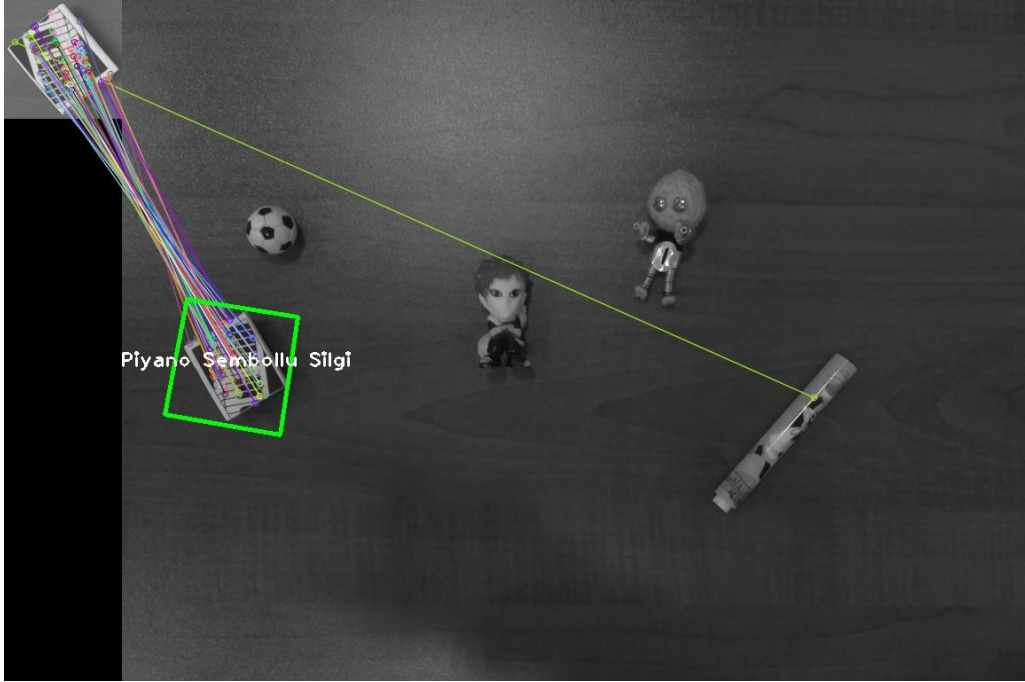
Şekil 5.42. 10 no'lu test görüntüsü için test işlemi (FAST+SURF)



Şekil 5.43. 11 no'lu test görüntüsü için test işlemi (FAST+SURF)



Şekil 5.44. 14 no'lu test görüntüsü için test işlemi (FAST+SURF)



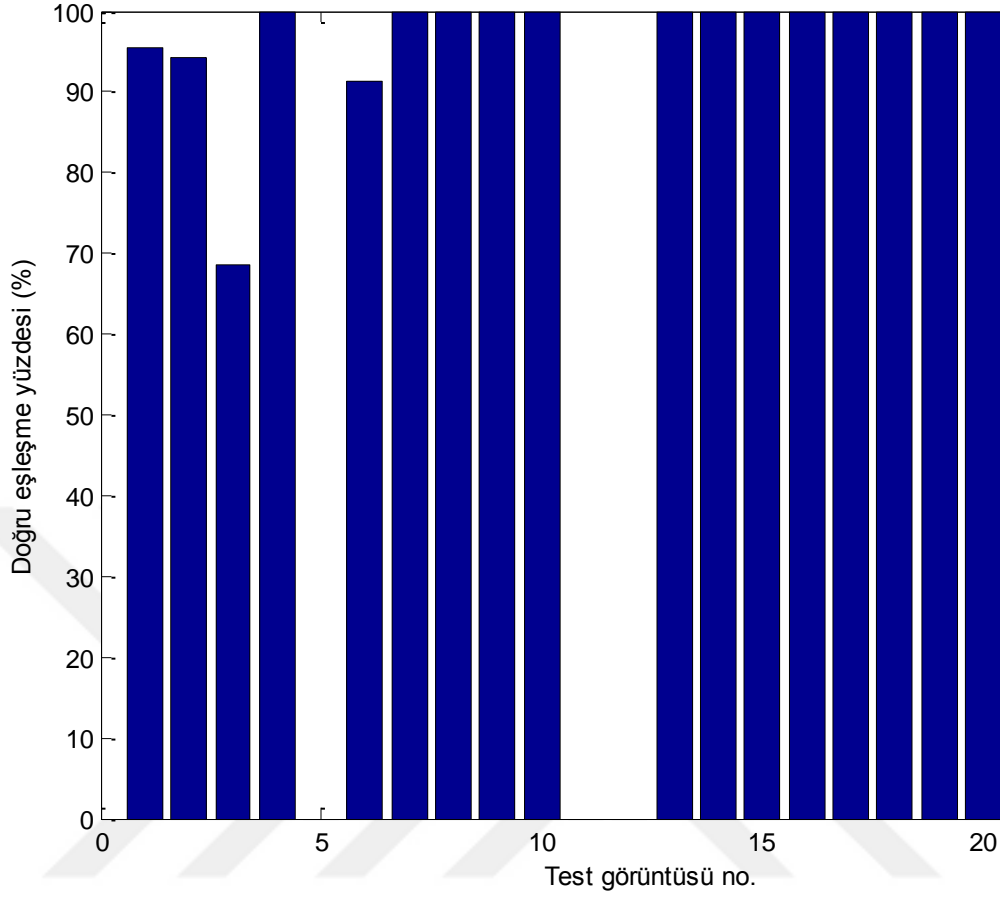
Şekil 5.45. 15 no'lu test görüntüsü için test işlemi (FAST+SURF)

ORB+ORB algoritmaları kullanılarak gerçekleştirilen test işlemleriyle elde edilen sonuçlar Tablo 5.7'de verilmiştir.

**Tablo 5.7.** Test işlemleri (ORB+ORB)

Test görüntüsü no.	Anahtar nokta sayısı	Öznitelik çıkarımı için geçen süre (ms)	Eşleştiği nesne görüntüsünün anahtar nokta sayısı	Eşleşen tanımlayıcı çifti sayısı	Aykırı eşleşme sayısı
1	500	798.5	253	45	2
2	500	841.6	253	35	2
3	500	852.3	239	134	42
4	500	796.9	239	5	0
5	500	812.7	313	178	178
6	500	817.2	284	46	4
7	500	781.4	179	6	0
8	500	801.6	179	7	0
9	500	806.9	93	7	0
10	500	844.6	222	19	0
11	500	812.8	313	313	313
12	500	813.0	313	313	313
13	500	820.3	130	5	0
14	500	806.0	126	10	0
15	500	793.7	209	20	0
16	500	802.8	209	27	0
17	500	802.9	239	8	0
18	500	797.7	239	13	0
19	500	790.5	72	16	0
20	500	783.9	58	8	0

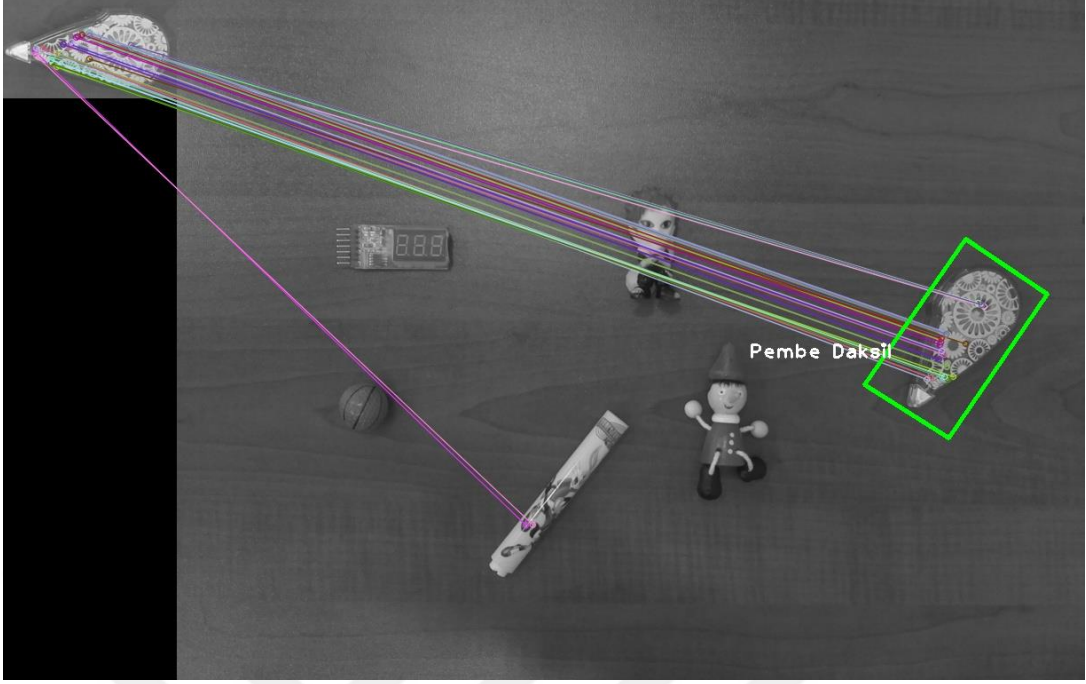
Tablo 5.7'deki sonuçlara göre 20 test görüntüsünün 17'si doğru sınıflandırılmış olup 5, 11 ve 12 no'lu test görüntülerindeki nesnenin yanlış sınıflandırıldığı, bunun için de tamamen aykırı eşleşmelerin meydana geldiği görülmüştür. Test görüntülerinin her biri için doğru eşleşme yüzdesi Şekil 5.46'da verilmiştir.



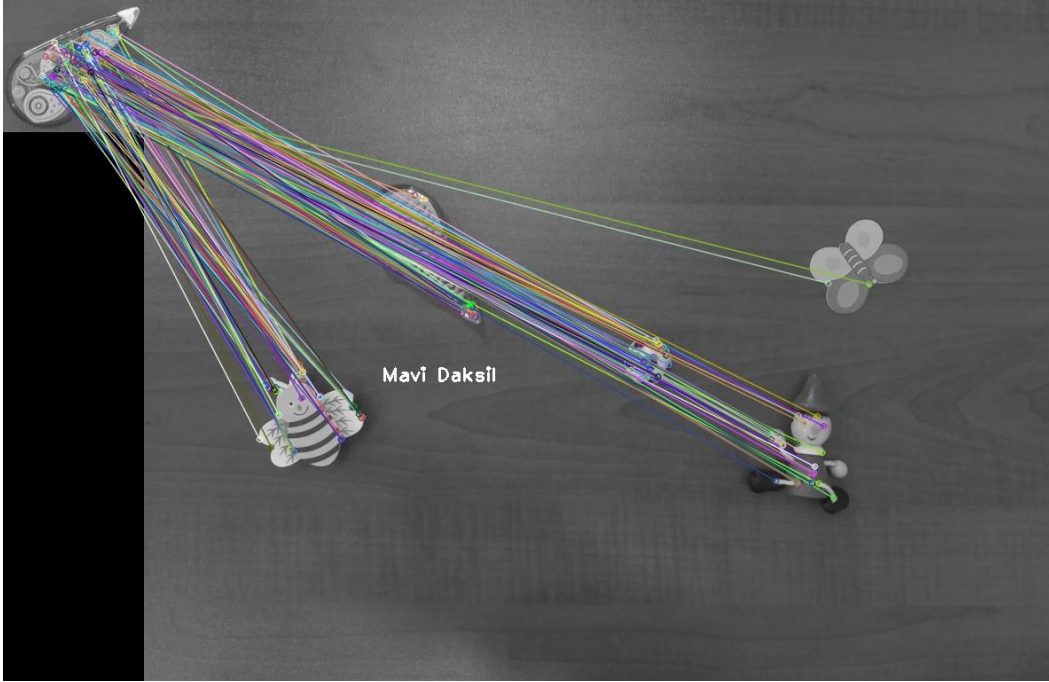
**Şekil 5.46.** Test görüntülerinin doğru eşleşme yüzdeleri (ORB+ORB)

Şekil 5.46'ya göre ORB+ORB algoritması ile elde edilen tanımlayıcıların doğru eşleşme yüzdesi 5, 11 ve 12 no'lu test görüntülerinin tamamen yanlış eşleşmeleri hariç diğer görüntülerde %68 ile %100 arasındadır.

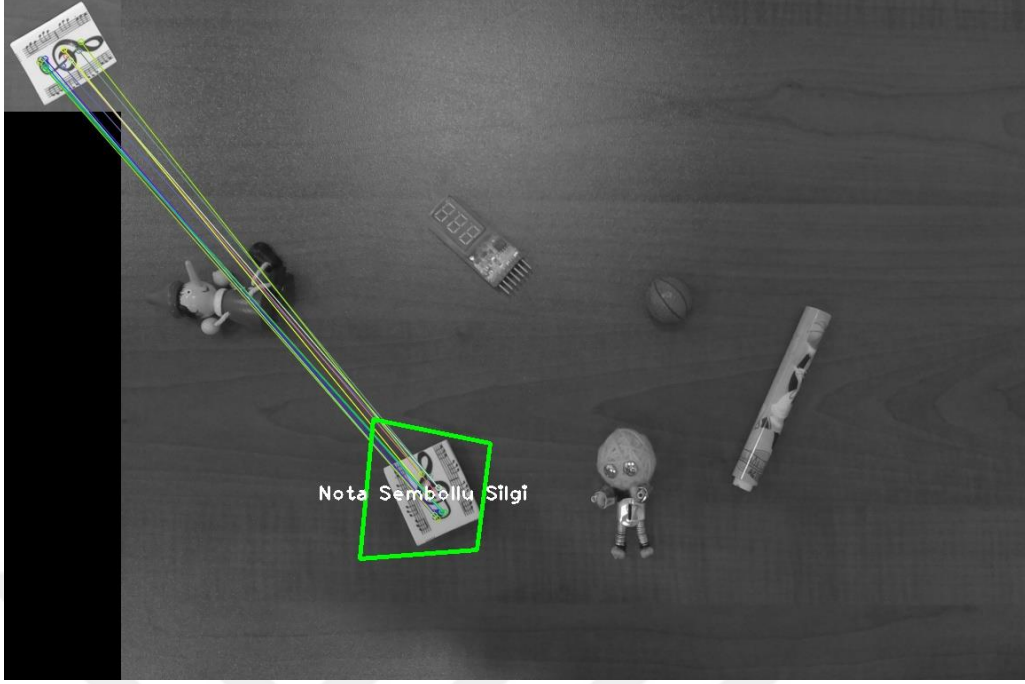
Gerçekleştirilen test işlemlerinden bazılarında Şekil 5.47, Şekil 5.48, Şekil 5.49 ve Şekil 5.50'de yer verilmiştir.



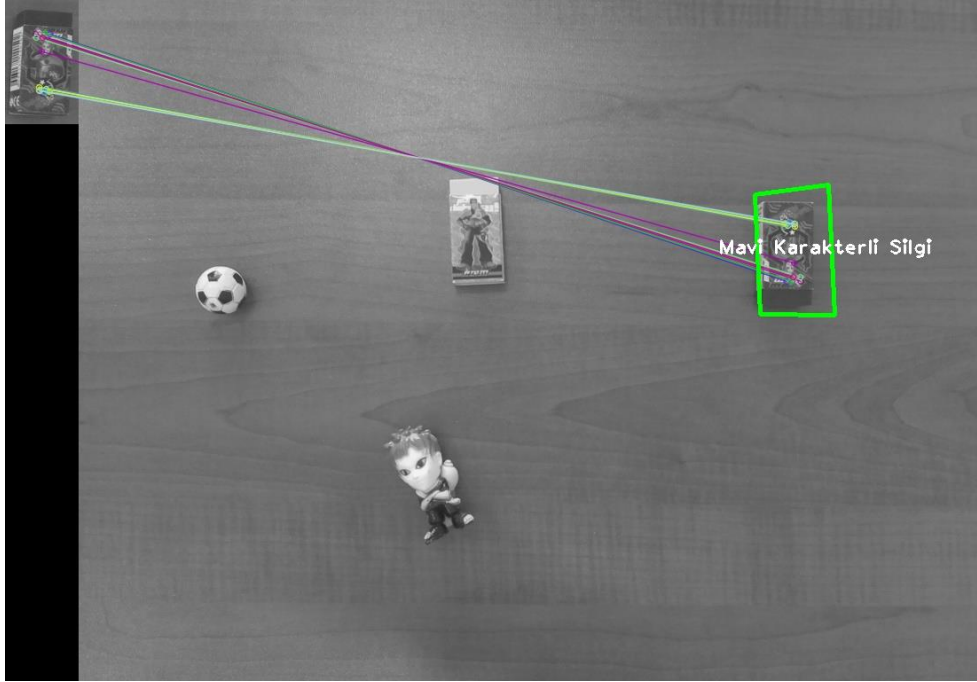
Şekil 5.47. 2 no'lu test görüntüsü için test işlemi (ORB+ORB)



Şekil 5.48. 5 no'lu test görüntüsü için test işlemi (ORB+ORB)



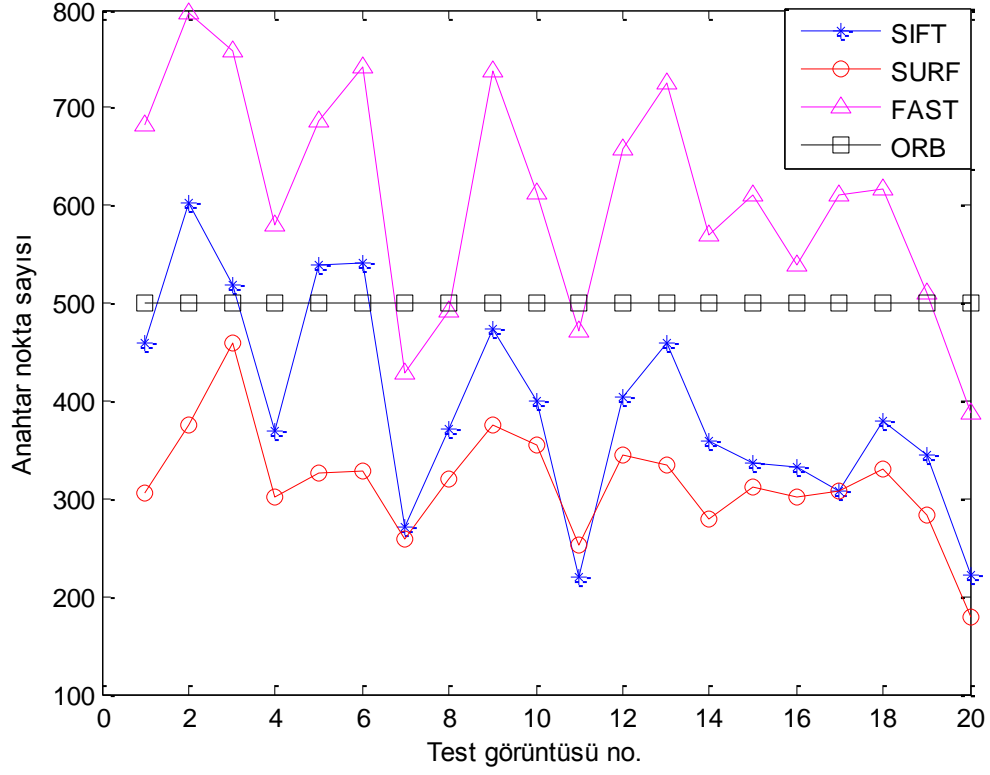
Şekil 5.49. 18 no'lu test görüntüsü için test işlemi (ORB+ORB)



Şekil 5.50. 19 no'lu test görüntüsü için test işlemi (ORB+ORB)

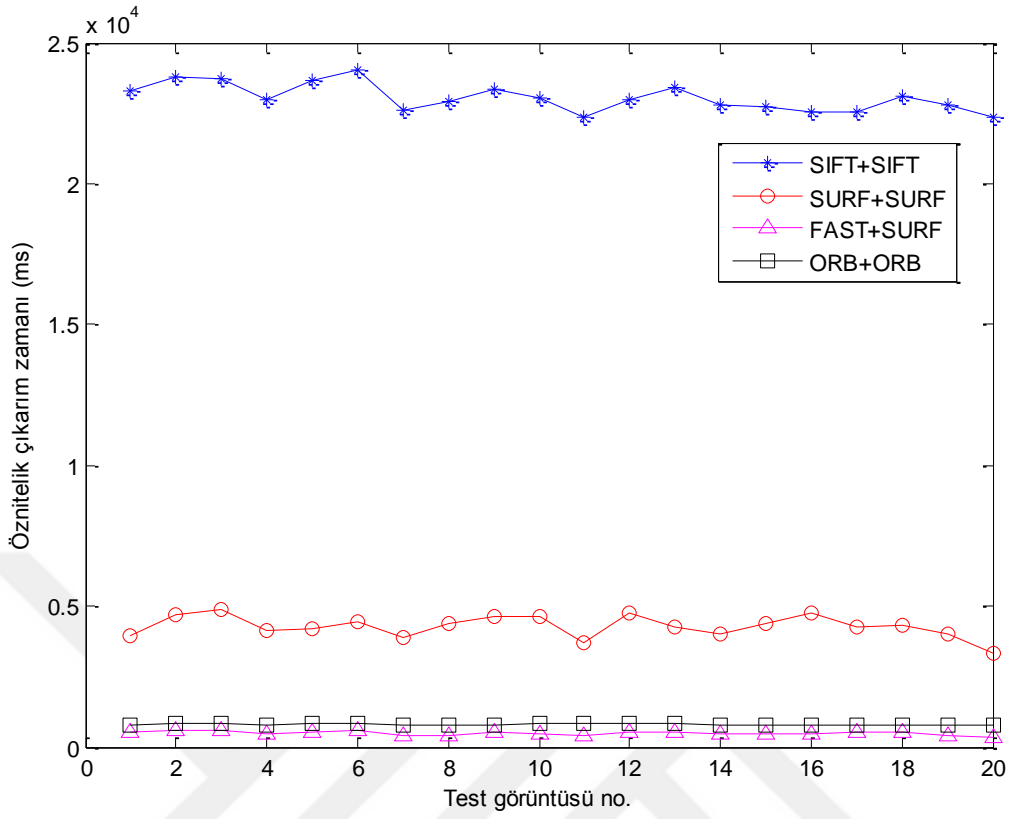
Tablo 5.4, Tablo 5.5, Tablo 5.6 ve Tablo 5.7'ye göre test görüntülerinden farklı algoritmalar ile tespit edilen anahtar nokta sayılarının karşılaştırılması Şekil 5.51'de gösterilmiştir.





Şekil 5.51. Tespit edilen anahtar nokta sayılarının karşılaştırılması

Tablo 5.4, Tablo 5.5, Tablo 5.6 ve Tablo 5.7'ye göre farklı algoritmalar ile test görüntülerindeki anahtar noktaların tespit edilip tanımlayıcılarının hesaplanması için geçen toplam zamanı ifade eden öznelilik çıkarım zamanlarının karşılaştırılması Şekil 5.52'de gösterilmiştir.



Şekil 5.52. Öznelik çıkarım zamanlarının karşılaştırılması

Kullanılan öznelik çıkarım algoritmalarının çalışma sürelerini birbirleriyle karşılaştırmak için anahtar nokta başına düşen öznelik çıkarım süreleri göz önünde bulundurulmuştur. Bunun için oluşturulan Tablo 5.8’de, test işleminde kullanılan 20 test görüntüsünün her biri için elde edilen Şekil 5.51’deki anahtar nokta sayılarının ve Şekil 5.52’deki öznelik çıkarım sürelerinin ortalamaları alınmıştır.

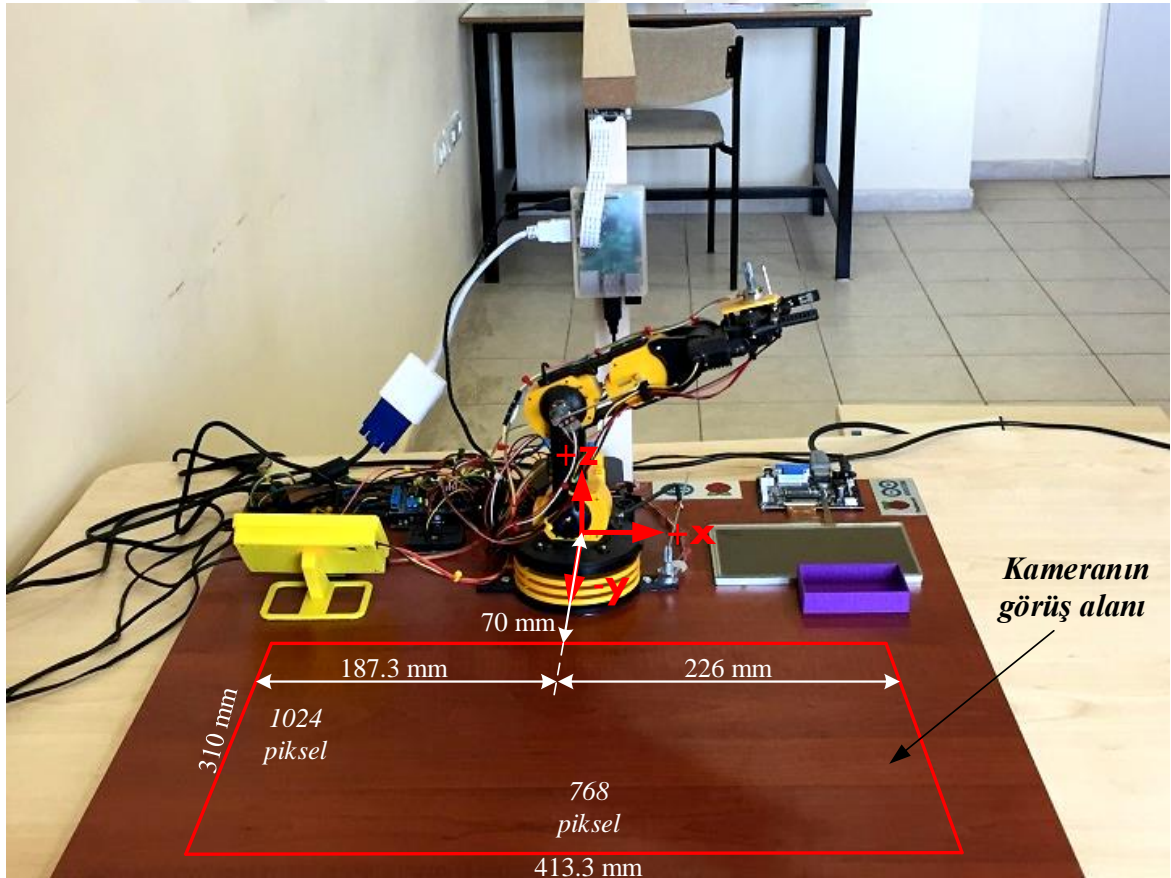
Tablo 5.8. Algoritmaların çalışma sürelerinin karşılaştırılması

Algoritma	Ortalama anahtar nokta sayısı	Ortalama süre (ms)	Süre/Anahtar nokta sayısı
SIFT+SIFT	395.35	23048	58.2977
SURF+SURF	316.4	4274.7	13.5104
FAST+SURF	610.7	481.47	0.7884
ORB+ORB	500	808.865	1.6177

Tablo 5.8'e göre, FAST+SURF'ün açık ara en hızlı çalışan algoritma olduğu, ORB+ORB'nin de FAST+SURF'e hız bakımından iyi bir seçenek olabileceği gözlemlenmiştir.

### 5.3.2. Tanınan Nesnenin Kavranıp Taşınmasına Yönelik Çalışmalar

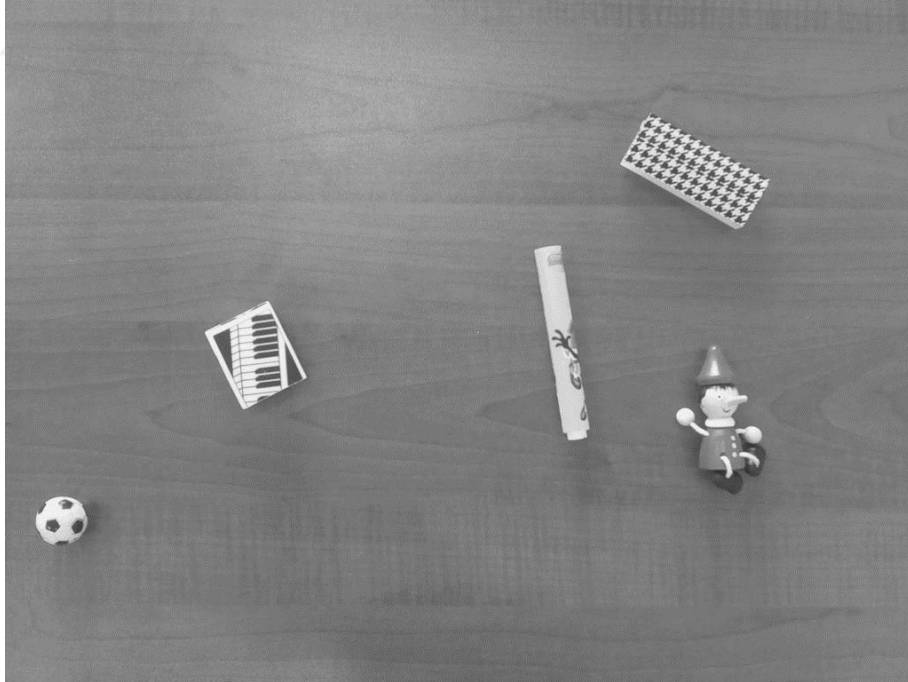
Robot kolun çalışma alanını kapsayan kameranın görüş alanı içerisine yerleştirilen nesnelere, tanınarak sınırlayıcı kutu ile yeri belirlenen nesnenin ağırlık merkezinin koordinatları piksel cinsinden elde edildikten sonra, robot kol tutucusunun nesneyi kavrayabilmesi için yönelmesi gereken x ve y konum koordinatları, Şekil 5.53'teki bilgiler ile piksel-mm dönüşümü yapılarak hesaplanmıştır.



Şekil 5.53. Piksel-mm dönüşümü için gerekli bilgiler

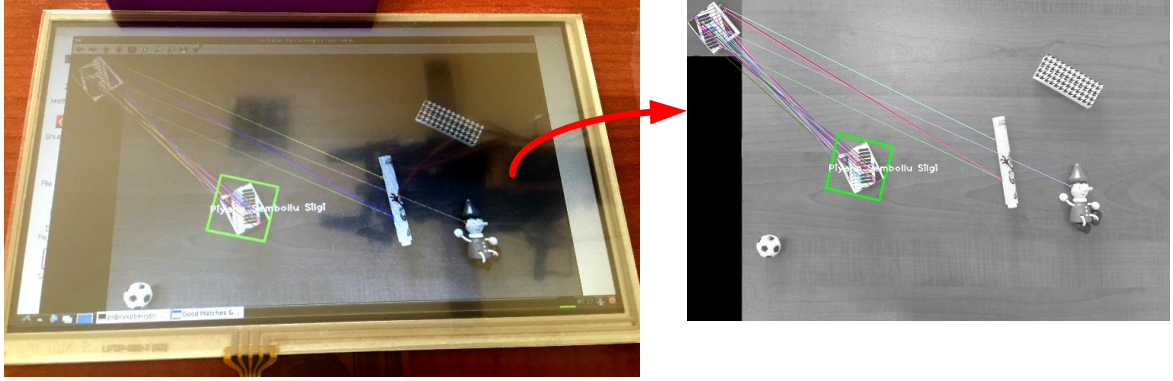
z konum koordinatı ise, daha önce belirtildiği gibi nesnelerin yükseklikleri birbirine çok yakın seçildiğinden sabit bir değer belirlenerek robot kol tutucu ucunun yönelimi gerçekleştirilmiştir. Tez çalışması için bu değer, 30 mm olarak belirlenmiş ve uygun olduğu görülmüştür. Ayrıca, robot kol tutucu ucunda 5. eksen bulunmadığı için kavranması istenen nesnelerin robot kol tutucusunun kavrayabileceği şekilde yerleştirilmiştir.

Robot kol tutucusunun tanınan nesneyi kavrayabilmesi için tutucu ucun ulaşması gereken  $p_x$ ,  $p_y$  ve  $p_z$  konum koordinatları belirlendikten sonra bu koordinat değerleri robot kol kontrol ünitesindeki Arduino Mega kartına seri haberleşmeyle gönderilmiştir. Ters kinematik eşitlikler ile de robot kol eklemlerinin bulunması gereken açı değerleri elde edilmiştir. Daha sonra robot kol eklemlerinin bu açı değerlerine hareketi gerçekleştirilip nesneyi kavraması sağlanmıştır. Nesne kavrandıktan sonra nesne bırakma haznesine bırakılıp robot kolun harekete başlamadan önceki konumunu almasıyla birlikte işlemler tamamlanmıştır. Deneysel olarak nesnenin tanınip taşınmasına yönelik gerçekleştirilen bu işlemleri, tezin metninde görsel olarak verebilmek için Şekil 5.54'teki örnek bir test görüntüsü kullanılmıştır.



Şekil 5.54. Kavranıp taşınması istenen nesnenin içinde bulunduğu örnek bir test görüntüsü

Şekil 5.54'teki bu test görüntüsü için SURF+SURF algoritması kullanılarak gerçekleştirilen nesne tanıma ve yer tespiti işlemleri Şekil 5.55'te gösterilmiştir.



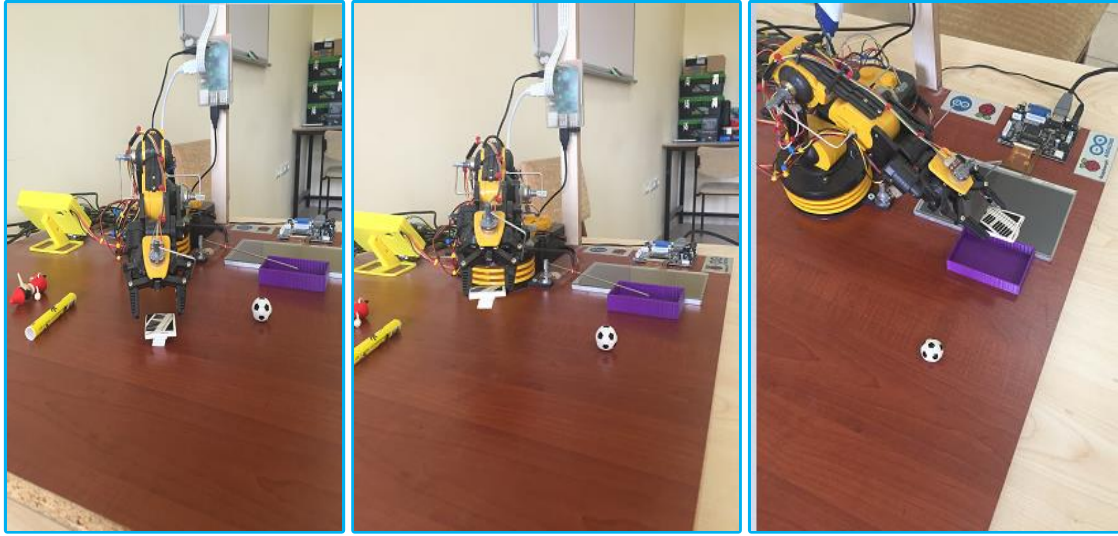
Şekil 5.55. Nesnenin tanınması ve yer tespiti işlemi (SURF+SURF)

Şekil 5.55'te tanınan nesnenin ağırlık merkezinin koordinatları piksel cinsinden tespit edilmiş ve gerekli hesaplamalar ile robot kol tutucu ucunun ulaşması gereken konum koordinatları Tablo 5.9'da gösterildiği gibi elde edilmiştir.

**Tablo 5.9.** Ağırlık merkezi ve tutucu ucun ulaşması gereken konum koordinatları

Ağırlık merkezi (piksel)	Konum koordinatları (mm)
(339,408)	$P_x= 89.18$ mm, $P_y= -215.31$ mm $P_z= 30$ mm (sabit)

Robot kol tutucu ucunun elde edilen bu konum koordinat değerlerine ulaşmasından sonra nesneyi kavraması, taşıması, bırakması ve robot kolun harekete başlamadan önceki pozisyonuna gelmesi için gerçekleştirdiği hareketler sırasıyla Şekil 5.56'da gösterildiği gibidir.



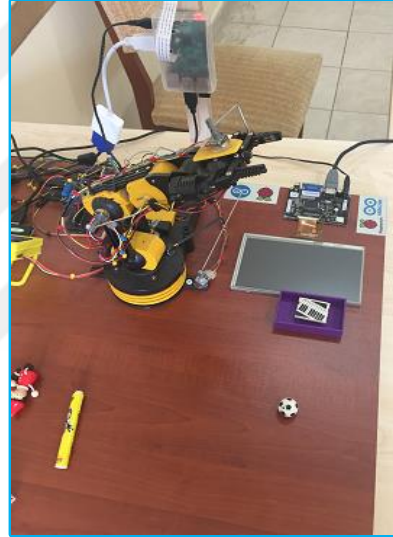
(a)

(b)

(c)



(d)



(e)

**Şekil 5.56.** Robot kolun hareket sırası: (a) nesneye ulaşma, (b) nesneyi kavrama, (c) nesneyi taşıma, (d) nesneyi bırakma ve (e) harekete başlamadan önceki pozisyonuna gelme

## 6. SONUÇLAR VE ÖNERİLER

Bu tezde bir nesnenin tanınması, nesne konumunun belirlenmesi ve bir robot kolun tutucusu yardımıyla nesnenin kavranıp taşınması işlemlerini gerçekleştirmek için bir deney düzeneği oluşturulmuştur. Deney düzeneğinde, robot kolun çalışma alanını görebilecek şekilde bir kamera yerleştirilmiştir. Kameradan alınan görüntülere nesne tanıma ve yerini tespit etme işlemlerini uygulamak için Raspberry Pi devre kartı kullanılmıştır. Tanınan nesneyi kavrayıp istenilen bir yere bırakmak için 4 serbestlik derecesi ve bir tutucu ucu bulunan OWI-535 robot kolu kullanılmıştır. Robot kol eklemlerinde bulunan motorları kontrol edebilmek için Arduino ve DA motor sürücü kartları kullanılmıştır.

Nesne tanıma ve yerini tespit etme işlemleri, Raspberry Pi devre kartına gömülü çalışan ve OpenCV kütüphanesi kullanılarak C++ dilinde yazılan yazılım ile başarılmıştır. Bu tezde, OpenCV kütüphanesinde mevcut olan, yerel öznitelik tabanlı anahtar nokta bulucu ve anahtar nokta tanımlayıcı algoritmalarından olan SIFT, SURF, FAST ve ORB kullanılmıştır.

Nesne tanıma ve yerini saptama işlemlerini deneysel olarak gerçekleştirmek için hem eğitim aşaması hem de test aşaması için farklı görüntüler alınmıştır. Alınan bu görüntüler üzerinde belirtilen yerel öznitelik tabanlı algoritmaların performansları karşılaştırılmıştır.

Çalışma için algoritmaların uygun parametreleri belirlendikten sonra hem eğitim hem de test görüntüleri için en fazla anahtar noktanın FAST algoritması ile, en az ise SURF algoritması ile tespit edildiği görülmüştür.

Anahtar noktaların tespitinden sonra eğitim ve test görüntülerine ait anahtar noktaların her biri için tanımlayıcılar hesaplanmıştır. Eğitim ve test görüntülerindeki bu tanımlayıcılar eşleştirilerek birbirine en yakın olanları belirlenerek nesne tanıma işlemleri gerçekleştirilmiştir. Deneysel çalışmalar için anahtar nokta bulucu ve anahtar nokta tanımlayıcı algoritmalar; SIFT+SIFT, SURF+SURF, FAST+SURF ve ORB+ORB şeklinde birlikte kullanılmış olup bu algoritmalar ile gerçekleştirilen nesne tanıma işlemlerinin sonuçlarının oldukça başarılı olduğu görülmüştür. 20 test görüntüsü için gerçekleştirilen test işlemlerinde sırasıyla 19, 19, 19 ve 17 doğru tanıma gerçekleşmiştir. Tanıma işlemi için gerçekleştirilen tanımlayıcı eşleştirme işleminde en fazla doğru eşleşme %97 ile %100 arasındaki başarıyla SIFT+SIFT algoritması ile elde edilmiştir. SIFT+SIFT algoritmasının ardından SURF+SURF gelmektedir. Bilimsel yazında belirtildiği gibi SIFT ve SURF tanımlayıcıları için FLANN tabanlı eşleştirici, ORB tanımlayıcıları için ise BF eşleştiricinin

Hamming metriği ile kullanılmasının uygun olduğu bir kez daha görülmüştür. Ayrıca eşleşmelerde meydana gelebilecek aykırı eşleşmeler RANSAC algoritması ile filtrelenmiş ve nesnenin yeri çok doğru bir şekilde tespit edilmiştir.

Algoritmaların çalışma sürelerini birbirleriyle karşılaştırmak için anahtar nokta başına düşen öznitelik çıkarım süreleri referans alınmıştır. Bunun için anahtar noktaların tespit edilip tanımlayıcılarının hesaplanması için geçen sürenin ortalamasının, tespit edilen anahtar noktaların sayısının ortalamasına oranı hesaplanmıştır. FAST+SURF ve ORB+ORB'nin bariz bir şekilde en hızlı çalışan algoritmalar olduğu görülmüştür. SIFT+SIFT algoritmasının ise en iyi eşleşme performansının aksine hız bakımından diğer algoritmalara göre çok yavaş çalıştığı görülmüştür. Elde edilen bu sonuçlar doğrultusunda genel olarak yerel öznitelik tabanlı algoritmalar ile nesne tanıma ve yer tespiti işlemlerinin başarılı bir şekilde gerçekleştiği sonucuna varmak mümkündür. Çünkü, robot kol çalışma alanı içerisine yerleştirilen nesneler kamera görüş açısı, dönme veya aydınlanma gibi dış etkenlere maruz kalabilmektedir.

Tanınan ve yeri tespit edilen nesnelerin kavranması ve taşınması için kullanılan OWI-535 robot kolunun düz kinematik analizi yapılmış ve daha sonra ters kinematik analiziyle de robot kol tutucusunun ulaşması gereken konumlar için eklemlerinin bulunması gereken açı değerleri başarılı bir şekilde elde edilmiştir.

Robot kolun eklemlerinde bulunan motorlar servo mekanizması gibi kendinden geri beslemeli bir sisteme sahip olmadığından geri besleme için eklemlere potansiyometreler yerleştirilmiştir. Potansiyometrelerin hassasiyetinden ve potansiyometre içindeki direnci oluşturan kalın film tabakasının zamanla aşınmasından dolayı gerçekleştirilmesi istenen hareketin küçük sapmalarla gerçekleştiği bazı durumlar olmuş ve bu yüzden nesneler kavranamamıştır. Ayrıca, nesnelerin ağırlık merkezlerinin bazen tam doğrulukta belirlenememesi ve kavranması istenen nesnenin robot kol tutucusu için uygun bir şekilde yerleştirilememesi gibi durumlar da nesnelere kavrama problemlerine neden olmasına rağmen genel olarak başarılı işlemler gerçekleştirilebilmiştir.

Tez kapsamında oluşturulan deney düzeneği, robotik ve bilgisayarla görmeye yönelik uygulamalar için endüstriyel ve eğitim amaçlı kullanılmaya elverişli olup elde edilen deneysel sonuçların bilimsel yazına katkı sağlayacağı düşünülmüştür.

Bu çalışmalarda uygulamalara yönelik bazı öneriler de yapılabilir. Örneğin; nesne tanıma ve yer tespiti işlemleri için oluşturulan sistemin eğitim ve test aşamasında kullanılan nesneler daha fazla ayırt edici özellik içerebilir. Bu sayede her bir nesneye özgü daha belirgin



öznitelikler çıkarılarak eşleşme ve bu sayede tanıma doğruluğu arttırılabilir. Ayrıca, OpenCV kütüphanesi ile geliştirilen yazılımın çalıştığı Raspberry Pi devre kartının işlemci hızı ve RAM kapasitesinin daha yüksek olduğu versiyonları seçilerek algoritmaların çalışma süresi kısaltılabilir.

Nesneleri kavrayıp taşımada kullanılan robot kol yerine de eklem hareketleri yüksek hassasiyetli servo motorlarla sağlanan ve tutucu ucunda ayrı bir eksene sahip bir robot kol kullanılması daha verimli sonuçların alınmasını sağlayabilir.



## KAYNAKLAR

- [1] **Akec, J. A., Steiner, S. J. and Stenger, F.,** 1998. An experimental visual feedback control system for tracking applications using a robotic manipulator, Industrial Electronics Society (IECON), Proceedings of the IEEE 24th Annual Conference, Aachen, Germany, **2**, 1125-1130.
- [2] **Carusone, J. and D'Eleuterio, G. M. T.,** 1998. The "feature CMAC": a neural-network-based vision system for robotic control, Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Leuven, Belgium, **4**, 2959-2964.
- [3] **Modi, K. P, Sahin, F. and Saber, E.,** 2005. An application of human robot interaction: development of a ping-pong playing robotic arm, IEEE International Conference on System, Man and Cybernetics (SMC), Waikoloa, HI, USA, **2**, 1831-1836.
- [4] **Kert, M.,** 2006. Gerçek görüntüden elde edilen koordinatlarla robot kol hareket optimizasyonu, *Yüksek Lisans Tezi*, Mustafa Kemal Üniversitesi Fen Bilimleri Enstitüsü, Antakya, Türkiye.
- [5] **Çiçek, S.,** 2006. Renge göre (kırmızı, yeşil, mavi) malzeme taşıyan robot kolu tasarımı ve uygulaması, *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [6] **Bustamante, L. and Gu, J.,** 2007. Localization of electrical outlet for a mobile robot using visual servoing, Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, Canada, **1**, 1211-1214.
- [7] **Dinh, B. H., Dunnigan, M. W. and Reay, D. S.,** 2008. A practical approach for position control of a robotic manipulator using a radial basis function network and a simple vision system, WSEAS Transaction on Systems and Control, **3(4)**, 289-298.
- [8] **Bayrak, A. ve Sarıtaş, M.,** 2008. Beş eksenli bir robot kolu simülasyonu ve engel/hedef takibi, ELECO, Bursa, Türkiye, 1-5.
- [9] **Luo, R. C., Tsai, C. H., Lai, C. C. and Chang, C. M.,** 2008. Autonomous security robot services using eye-in-hand visual servo system, IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO), Taipei, Taiwan, 1-6.
- [10] **Erşan, E.,** 2009. Bir robot kolunun görmeye dayalı kontrolü, *Yüksek Lisans Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [11] **Arserim, M. A.,** 2009. Akıllı yöntemlerle nesne tanınması ve robot kolu kontrolü, *Doktora Tezi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, Türkiye.

- [12] **Anh, L. T. and Song, J. B.**, 2010. Object tracking and visual servoing using features computed from local feature descriptor, International Conference on Control, Automation and Systems (ICCAS), Gyeonggi-do, Korea, 1044-1048.
- [13] **Seelye, M., Gupta, G. S., Seelye, J. and Mukhopadhyay, S. C.**, 2010. Camera-in-hand robotic system for remote monitoring of plant growth in a laboratory, IEEE Instrumentation and Measurement Technology Conference (I2MTC), Austin, TX, USA, 809-814.
- [14] **Huang, G. S.**, 2011. Robotic arm grasping and placing using edge detection system, IEEE 8th Asian Control Conference (ASCC), Kaohsiung, Taiwan, 960-964.
- [15] **Rapp, H. H.**, 2011. A ping-pong ball catching and juggling robot, a real-time framework for vision guided acting of an industrial robot arm, IEEE 5th International Conference on Automation Robotics and Applications (ICARA), Wellington, New Zealand, 430-435.
- [16] **Ali, H., Seng, T. C., Hoi, L. H. and Elshaikh, M.**, 2012. Development of vision-based sensor of smart gripper for industrial applications, IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA), Meleka, Malaysia, 300-304.
- [17] **Erdoğan, T.**, 2012. Hareketli konveyör üzerinde kamera görüntüsü ile nesne tanıma ve nesnelere yerine koyma uygulaması, *Yüksek Lisans Tezi*, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, İzmir, Türkiye.
- [18] **Basile, F., Caccavale, F., Chiacchio, P., Coppola, J. and Curatella, C.**, 2012. Task-oriented motion planning for multi-arm robotic systems, *Robotics and Computer-Integrated Manufacturing*, **28(5)**, 569-582.
- [19] **Şenel, F. A.**, 2013. Görüntü işleme ve beş eksenli robot kol ile üretim bandında nesne denetimi, *Yüksek Lisans Tezi*, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta, Türkiye.
- [20] **Prabhu, V. and Soman, K. P.**, 2011. Voice interfaced arduino robotic arm for object detection and classification, *International Journal of Scientific & Engineering Research (IJSER)*, **4(7)**, 445-449.
- [21] **İnan, T.**, 2013. Hareketli hedefi takip eden robot kolu sistemi, *Yüksek Lisans Tezi*, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [22] **Wohlking, W. and Vincze, M.**, 2011. Ensemble of shape functions for 3D object classification, IEEE International Conference on Robotics and Biomimetics (ROBIO), Phuket, Thailand, 2987-2992.
- [23] **Wohlking, W. and Vincze, M.**, 2011. Shape distributions on voxel surfaces for 3D object classification from depth images, IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 115-120.

- [24] **Redondo-Cabrera, C., Lopez-Sastre, R. J., Acevedo-Rodriguez J. and Maldonado-Bascon, S.**, 2012. SURFing the point clouds: Selective 3D spatial pyramids for category-level object recognition, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, 3458-3465.
- [25] **Gal, R. and Cohen-Or, D.**, 2006. Salient geometric features for partial shape matching and similarity, ACM Transactions on Graphics (TOG), **25(1)**, 130-150.
- [26] **Jain, A. K.**, 1989. Fundamentals of digital image processing, Prentice Hall Inc., Prentice Hall International Edition.
- [27] **Aytekın, C.**, 2011. Geo-spatial object detection using local descriptors, *Master of Science Thesis*, METU Institute of Science, Ankara, Turkey.
- [28] **Mingqiang, Y., Kidiyo, K. and Joseph, R.**, 2008. A survey of shape feature extraction techniques, Pattern Recognition, IN-TECH, 43-90.
- [29] **Morevec, H.**, 1981. Rover visual obstacle avoidance, Proceeding of the 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, Canada, **2**, 785-790.
- [30] **Harris, C. and Stephens, M.**, 1988. A combined corner and edge detector, 4th Alvey Vision Conference (AVC), Manchester, UK, **15**, 147-151.
- [31] **Harris, C.**, 1993. Geometry from visual motion, Active vision, MIT Press, Cambridge, MA, USA, 263-284.
- [32] **Schmid, C. and Mohr, R.**, 1997. Local grayvalue invariants for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), **19(5)**, 530-534.
- [33] **Nelson, R. C. and Selinger, A.**, 1998. Large-scale tests of a keyed, appearance-based 3-D object recognition system, Vision Research, **38(15)**, 2469-2488.
- [34] **Shree, K. N.**, 1995. Visual learning and recognition of 3-D objects from appearance, International Journal of Computer Vision (IJCV), **14(1)**, 5-24.
- [35] **Swain, M. and Ballard, D.**, 1991. Color indexing, International Journal of Computer Vision (IJCV), **7(1)**, 11-32.
- [36] **Schiele, B. and Crowley, J. L.**, 1996. Object recognition using multidimensional receptive field histograms, 4th European Conference on Computer Vision (ECCV), Cambridge, UK, **1**, 610-619.
- [37] **Lowe, D. G.**, 1999. Object recognition from local scale-invariant features, IEEE 7th International Conference on Computer Vision (ICCV), Corfu, Greece, **2**, 1150-1157.

- [38] **Ledwich, L. and Williams, S.**, 2004. Reduced SIFT features for image retrieval and indoor localisation, Australian Conference on Robotics and Automation (ACRA), Canberra, Australia, **322**, 3.
- [39] **Guan, F., Liu, X., Feng, W. and Mo, H.**, 2013. Multi target recognition based on SURF algorithm, 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, **1**, 444-453.
- [40] **Heo, H., Lee, J. Y., Lee, K. Y. and Lee, C. H.**, 2013. FPGA based Implementation of FAST and BRIEF algorithm for object recognition, TENCON 2013-2013 IEEE Region 10 Conference (31194), Xi'an, China, 1-4.
- [41] **Rublee, E., Rabaud, V., Konolige and K., Bradski, G.**, 2011. ORB: an efficient alternative to SIFT or SURF, International Conference on Computer Vision (ICCV), Barcelona, Spain, 2564-2571.
- [42] **Roberts, L. G.**, 1965. Machine perception of three-dimensional solids, Optical and Electro-Optical Information Processing, MIT Press, Cambridge, MA, USA, 159-197.
- [43] **Dickinson, S. J.**, 1999. Object representation and recognition, What is cognitive science?, 172-207.
- [44] **Lowe, D. G.**, 2004. Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision (IJCV), **60(2)**, 91-110.
- [45] **Joshi, H. and Sinha, M. K.**, 2013. A Survey on image mosaicing techniques, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), **2(2)**, 365-369.
- [46] **Brown, M., Lowe, D. G.**, 2002. Invariant features from interest point groups, Proceeding of British Machine Vision Conference (BMVC), Cardiff, UK, 253-262.
- [47] **Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L.**, 2008. Speeded-up robust features (SURF), Computer Vision and Image Understanding, **110(3)**, 346-359.
- [48] **Mikolajczyk, K. and Schmid, C.**, 2005. A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), **27(10)**, 1615-1630.
- [49] **Rosten, E. and Drummond, T.**, 2006. Machine learning for high speed corner detection, Proceeding of the 9th European Conference on Computer Vision (ECCV), Graz, Austria, **1**, 430-443.
- [50] **Viswanathan, D. G.**, 2009. Features from accelerated segment test (FAST), [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/AV1011/AV1FeaturefromAcceleratedSegmentTest.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/AV1FeaturefromAcceleratedSegmentTest.pdf).

- [51] **Calonder, M., Lepetit, V., Strecha, C. and Fua, P.,** 2010. BRIEF: Binary robust independent elementary features, European Conference on Computer Vision (ECCV), Springer Berlin, Heidelberg, 778-792.
- [52] **Rosin, P. L.,** 1999. Measuring corner properties, Computer Vision and Image Understanding, **73(2)**, 291-307.
- [53] **Muja, M. and Lowe, D. G.,** 2009. Fast approximate nearest neighbors with automatic algorithm configuration, International Conference on Computer Vision Theory and Applications (VISAPP), Lisboa, Portugal, **1**, 331-340.
- [54] **Pratt, L. Y.,** 1989. Comparing biases for minimal network construction with back-propagation, Part of: Advances in neural information processing systems 1, Morgan Kaufmann Publishers Inc., 177-185.
- [55] **Silpa-Anan, C. and Hartley, R.,** 2008. Optimised KD-trees for fast image descriptor matching, IEEE Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, USA, 1-8.
- [56] **Vaghela, D. and Naina, P.,** 2014. A review of image mosaicing techniques, International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS), **2(3)**, 431-437.
- [57] **Fischler, M. A. and Bolles, R. C.,** 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM, **24(6)**, 381-395.
- [58] **Derpanis, K. G.,** 2010. Overview of the RANSAC Algorithm, Image Rochester NY, **4(1)**, 2-3.
- [59] **Tonbul, T. S. ve Sarıtaş, M.,** 2003. Beş eksenli bir EDUBOT robot kolunda ters kinematik hesaplamalar ve yörünge planlaması, Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 145-167.
- [60] **Bingül, Z. ve Küçük, S.,** 2005. Robot Tekniği I, Birsen Yayınevi, İstanbul.
- [61] **Maxwell, E. A.,** 1951. General homogeneous coordinates in space of three dimensions, Cambridge University Press, New York.
- [62] **Denavit, J. and Hartenberg, R. S.,** 1955. A kinematic notation for lower-pair mechanisms based on matrices, Journal of Applied Mechanics, **22(2)**, 215-221.
- [63] **Güzel, M. S.,** 2008. Altı eksenli robot kolun hareketsel karakteristiğinin görsel programlanması ve gerçek zamanlı uygulamalar, *Yüksek Lisans Tezi*, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye.

## ÖZGEÇMİŞ

Çağrı KAYMAK, 1989 yılında Ankara’da doğdu. Orta öğrenimini Uşak Orhan Deniz Anadolu Lisesi’nde tamamladı. 2009 yılında girdiği Fırat Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü’nden 2013 yılında mezun oldu. 2013 yılında girdiği Fırat Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda yüksek lisans eğitimine başladı. 2014 yılının ocak ayından itibaren Fırat Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü’nde araştırma görevlisi olarak çalışmaktadır.

