

# Bir Otomasyon Hattında Parça Seçim ve Ayrıştırma İşlemlerinde Kullanılan Çoklu Robotlar için Görüntü İşleme Temelli Görev Ataması Yapan Sistem Önerisi

<sup>1</sup>Yasin AK ve <sup>2</sup>Serkan DERELİ

<sup>1</sup>Lisansüstü Eğitim Enstitüsü, Mekatronik Mühendisliği Ana Bilim Dalı, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye

<sup>2</sup>Sakarya Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye

## Özet

Bu çalışmada aynı otomasyon hattında farklı biçimde ve türde üretilen parçaların tespit edilmesi ve ayrıştırılması işlemi için birden fazla robot kolun görev paylaşımı için görüntü işleme temelli bir yöntem önerilmektedir. Yüksek hızda ve doğrulukta üretim, insan kaynaklı hataları en aza indirme ve insan gücü yerine makine gücü kullanma gibi avantajların birden fazla robot kol kullanarak robotların iş birliği ile yaptırılıyor olması bu avantajları en üst seviyeye çekmektedir. Bunun için çok sayıda robot kolun bulunduğu iş hattında her bir robot kolun seçmesi gereken nesneyi görüntü işleme ile tespit edilen nesne özelliklerine göre robot kollara atanması işlemi gerçekleştirilmiştir. Birden fazla robot kol kullanımı ile robotların iş birliği anlamında önemli olan bu çalışmanın ön hazırlık çalışması Matlab GUI arabiriminde gerçekleştirilmiştir. Oluşturulan kullanıcı arayüzü ile kullanıcı tabanlı renk, şekil ve boyut seçimi sonucunda parça tespit işlemi yaptırılmaktadır.

**Anahtar Kelimeler:** görüntü işleme, çoklu robotlar, robot kol, robot ve otomasyon

## Abstract

In this study, an image processing-based method is proposed for the task sharing of more than one robot arm for the detection and separation of parts produced in different forms and types in the same automation line. High speed and precision production, minimizing human errors by using machine power instead of human power, and performing tasks by using more than one robot arm with the cooperation of robots maximize these advantages. Object selection of each robot arm in the automation line with more than one robot arm for this is made process of assignment to robot arms according to its properties of detected object with image processing. The simulation tests of this study, which is important for the cooperation of robots with the use of more than one robot arm, were carried out using the Matlab GUI. Object detection is made as a result of the choice of color, shape and size with created user interface.

**Key words:** image processing, multi-robot, robot arm, robot and automation

## 1. Giriş

Günümüzde teknolojinin ilerlemesiyle birlikte endüstriyel sanayi teknolojilerinde hızlı üretim, iş güvenliği ve zaman gibi gereksinimlerden dolayı makine gücüne ihtiyaç duyulmaktadır. Makine gücü kullanılması insan kaynaklı oluşabilecek hataların ortadan kaldırılmasını hedeflemektedir.

\*Corresponding author: Address: Faculty of Technology, Department of Mechatronic Engineering, Sakarya University, 54187, Sakarya TURKEY.

Görüntü işleme esaslı geliştirilen sistemler; insan gücü ve algısının üretim hızına yeterli gelmemesi, insanların sağlık ve güvenlik açısından riskli bölgelerde çalışması ve insan kaynaklı oluşabilecek verim kaybı gibi sorunları minimize etmektedir. Birden fazla robot kullanımı ile yüksek hızda ve doğrulukta üretim, insan kaynaklı hataları en aza indirmeye ve insan gücü yerine makine gücü kullanma gibi avantajların robotların iş birliği ile yaptırılıyor olması bu avantajları en üst seviyeye çekmek hedeflenmektedir.

Basile, yaptığı çalışmada istenen nesnelerin konumlarını bularak birden fazla robot kolunu eş zamanlı olarak çalıştırmışlardır. Nesnenin konumları bulunduğundan sonra ters kinematik analiz yapıp robot kollarından en uygunu hangisi ise onun robot kola hareketi sağlanmıştır [1]. Erdoğan ve Sarel, bir çoklu robot sisteminin bir görevi yerine getirmesi üzerine görev atama, iş paylaşımı ve yürütme gibi işlemleri ve bu işlemleri tümüyle otonom bir şekilde gerçekleştirmek üzere bir çalışma gerçekleştirmiştir [2]. Dönmez ve Kocamaz, yaptıkları çalışmada birden fazla robot ile birden fazla hedefin gezilmesi bu durumlarına göre görev paylaşımı yapılması ve yük dengeleme sistemini gerçekleştirmiştir. Yapılan görev dağılımı yük dengeleme sisteminin aktif veya pasif olma durumlarına göre gerçekleştirilmiştir [3]. Çayırpunar, yaptığı çalışmada birden fazla robotun bir arama görevinde haberleştirilmesi işlemini gerçekleştirmiştir [4]. Cuşkun ve arkadaşları, görüntü işleme tekniklerini kullanarak farklı özellikteki plakaları tespit ederek 4 serbestlik derecesine sahip bir robot mekanizması ile taşıma işlemini yapan çalışmayı gerçekleştirmiştir [5]. Thenmozhi ve Reddy, yaptıkları çalışmada makine görüşünü ve görüntü analizini kullanarak şeker kamışı mahsulündeki böceklerin şeklini tespit etme işlemini gerçekleştirmiştir [6]. Rahman ve arkadaşları, nesnelere rengine göre ayırabilen ve bu nesnelere ile arasındaki mesafeyi belirleyebilen 6 serbestlik derecesine sahip bir robot kol çalışmasını gerçekleştirmiştir [7]. Kaymak ve arkadaşları, yaptıkları çalışmada bir elma bahçesindeki kırmızı elmaların tespit edilmesi ve sayılması işlemini gerçekleştirmiştir [8]. Alitappeh ve Jeddisaravi, yaptıkları çalışmada çoklu robot sistemlerinde görev tahsisi konusundaki zorluklara yönelik bir çözüm gerçekleştirmişlerdir [9]. Dos Reis ve arkadaşları yaptıkları çalışmada robotlara görev dağıtımını üzerine bir çözüm önermektedir [10]. Woosley ve arkadaşları, otonom robotların bilmedikleri bir ortamda bilgi toplama ve keşif üzerine yaptıkları çalışmada yol çakışmalarını önlemek için çözüm önerisinde bulunmuşlardır [11]. Marcotte ve arkadaşları, robotların işbirliği üzerine gerçekleştirdiği çalışmada belirli aralıklarda iletişimi iyileştirme odaklı öneride bulunmuşlardır [12]. Alsayegh ve arkadaşları, gerçek dünya kısıtlamalarından kaynaklı oluşabilecek iletişim sorunları üzerine robotlar arası haberleşme protokolleri üzerine bir çalışma önermişlerdir [13]. Serra-Gomez ve arkadaşları, yaptıkları çalışmada robotlar arası koordineli çalışmayı sağlayarak çarpışmayı önlemek üzerine bir iletişim yöntemi sunmaktadırlar [14]. Fan ve arkadaşları, yaptıkları çalışmada aynı ortamda bulunan birden fazla robotun çarpışmadan kaçınmasını sağlamak geliştirdikleri yöntemi sunmaktadırlar [15]. Patel ve arkadaşları, yaptıkları çalışmada birden fazla insan ve birden fazla robotun iletişimi için bir öneride bulunmuşlardır [16]. Thabit ve Mohades, yaptıkları çalışmada çok robotlu çok amaçlı parçacık sürü optimizasyonu ile çoklu robotların yol planlaması üzerine bir öneride bulunmuşlardır [17].

## 2. Materyal ve Yöntem

Birden fazla robot kolun bir otomasyon hattında gerçek zamanlı bir şekilde nesne tespit işleminde ki görev dağılımı için yapılacak olan çalışmanın ön hazırlık kısmında görüntü üzerinden nesnelere renk, şekil ve boyuta bağlı özelliklerine göre nesne tespiti yapılmıştır. Birden fazla robotun haberleşmesi ile birlikte bir otomasyon hattında bulunan nesnelere referans olarak alınan renk, şekil ve boyut bilgisine göre taşınması işlemi gerçekleştirilecektir. Örneğin bir robot kolun kırmızı renkte olan nesnelere taşınması istenilirken diğer bir robot kolun mavi renkte olan nesnelere taşınması istenildiği durumda olacaktır. Bu durumlarda her bir robot kolu için çalışma alanı belirlenecek ve robot kollar için belirlenen nesnelere çalışma alanı dışındaysa robotlar arasında haberleşme gerçekleştirilerek nesnelere çalışma alanı içine taşınacaktır. Nesne tespit işlemi için Matlab programı kullanılmıştır. Oluşturulan yazılımda bir ana fonksiyon ve üç alt fonksiyon bulunmaktadır. Renk tespiti için renkTespit, şekil tespiti için şekilTespit ve boyut tespiti için boyutTespit alt fonksiyonları yazılmıştır. Oluşturulan fonksiyonlar ile nesne tespiti yapıldıktan sonra Matlab GUI ile bir arayüz oluşturulmuştur. Yazılan fonksiyonlar arayüzde bulunan buton vb. nesnelere entegre edilmiştir. Oluşturulan arayüzün SimMechanics ile bağlantısı sağlanarak robot kolların hareketleri simüle edilmiştir.

### 2.1. Görüntü İşleme

Yapılan bu çalışmada nesne tespit işlemi için görüntü işleme kullanılmıştır. Görüntü işleme uygulamalarında kameradan alınan görüntü eşikleme, aşındırma, açma gibi birçok morfolojik işlemle geçirilerek nesne tespit aşamasına getirilmektedir. Görüntü işleme için Matlab kullanılmıştır. Matlab ile yapılan bu çalışmada karmaşık görüntü içerisinde bir görüntü segmentasyonu yapılmıştır. Bu görüntü segmentasyonu içerisinde kullanıcı giriş parametreleri belirleyecektir. Giriş parametresi olarak kırmızı, yeşil ve mavi olmak üzere 3 farklı renk, yuvarlak, kare ve üçgen olmak üzere 3 farklı şekil ve küçük ve büyük olmak üzere 2 farklı boyut referans olarak alınmıştır. Alınan bu bilgiler doğrultusunda istenilen nesnenin tespiti yapılabilmektedir.

#### 2.1.1. Renk Algılama

Matlab ile yapılmış olan görüntü segmentasyonu işleminde ilk olarak görüntü okuma işlemi gerçekleştirildi. Ardından kullanıcının girdi olarak gönderdiği renkte bulunan nesnelere tespit yapıldı. Renk tespiti için alınan görüntüye HSV dönüşümü uygulandı ve istenilen rengin eşik değerleri uygulandı. HSV renk uzayı, renkleri sırasıyla renk özü, doygunluk ve parlaklık olarak tanımlar. İnsan görü düzenine daha yakın bir yapıda olan HSV renk uzayı, bu metodu kullanım nedeni olarak öne çıkarmaktadır. HSV renk uzayı ile dönüşüm yapıldıktan sonra eşik değeri uygulanarak istenilen renk tespit edilmiştir.

### 2.1.2. Şekil Algılama

Yapılan bu çalışmada renk tespiti işleminin ardından şekil tespiti yapılmaktadır. Renk tespiti işleminin sonucunda elde edilen binary görüntü üzerinde kullanıcıdan gelen şekil bilgisine göre şekil tespit edilmektedir. Denklem 2.1’de yer alan formül kullanılarak elde edilen görüntü üzerindeki şekillerin daireselliği hesaplanmıştır. Hesaplamalar sonucunda elde edilen değerlere göre şekillerin durumu tespit edilmiştir. Şekillerin durumu için eğer hesaplanan değer 0.3-0.6 aralığında ise üçgen, 0.6-0.9 aralığında ise kare/dörtgen 0.9’dan büyük ise yuvarlak olarak belirlenmiştir. İstenilen şekil için bulunan nesnelere aynı boyutlarda binary görüntü olarak başka bir görsele yazdırılmıştır.

$$\text{Dairesellik} = ((4 * \pi * A) / P^2)^{1/2} \quad (1)$$

### 2.1.3. Boyut Tespiti

Renk tespiti ve şekil bulma işleminin ardından kullanıcıdan gelen boyut bilgisine göre istenilen renk, şekil ve boyuttaki nesnenin alan ve koordinat bilgileri buldurulmaktadır. Bu işlem için şekil buldurma kısmında elde edilen bilgiler kullanılarak kullanıcıdan gelen boyut bilgisine göre bulunan şekillerde alanların karşılaştırılması işlemi yapılarak istenilen şekil tespit edilebilmektedir. Öncelikli olarak bulunan nesnelere alanları buldurularak bulunan alanların karşılaştırılması yapılmış ve kullanıcıdan gelen boyut bilgisinin büyük veya küçük olma durumuna göre nesne tespit işlemi yapıp bulunan nesne aynı boyutlarda başka bir görsele çizdirilmiştir.

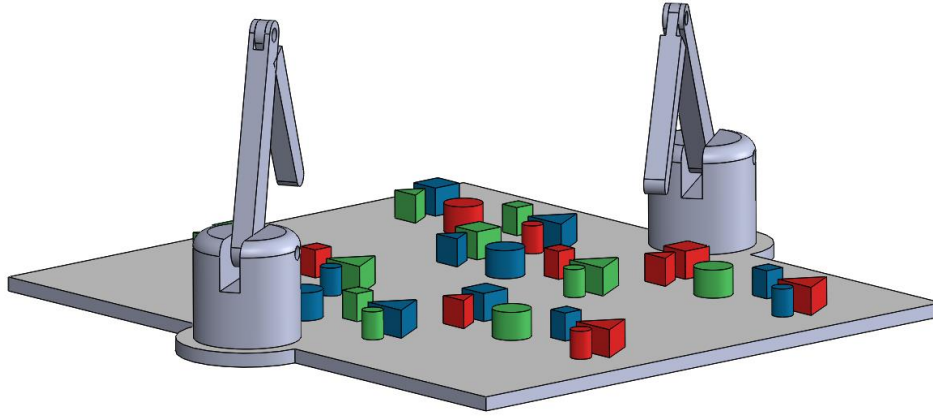
## 2.2. Modelleme

Bu çalışmada birden fazla robot kolun görev ataması gerçekleştirildikten sonra belirlenen nesnelere konumlarını görüntü işleme ile buldurulduktan sonra bulunan bu konumlara göre robot kollarının hareketlerini gözlemlemek için simülasyon ortamı oluşturularak kolların hareketleri gözlemlenmiştir. Simülasyon için Matlab Simmechanics kullanılmıştır. Mekanik tasarım programında robot kol çizimleri yapıldıktan sonra Matlaba aktarılmıştır. Matlab GUI ile oluşturulan arayüz ve simülasyon ortamı arasında bağlantı kurularak oluşturulan arayüz programı ile istenilen nesnelere tespit edilip, koordinatları bulunmuş ve bu koordinatlara göre kinematik analiz yapılarak hesaplanan açılar robot kollarına gönderilmiştir. Simülasyon ortamında kolların hareketi için PID control uygulanmıştır.

### 2.2.1. Simülasyon Ortamının Oluşturulması

Simülasyon ortamında robot kolların hareketini gözlemleyebilmek için öncelikle robot kol tasarımları yapılmıştır. Robot kol tasarımı için SolidWorks programı kullanılmıştır. Oluşturulan tasarımda robot kolların yerleştirildiği tabana tespit edilecek nesnelere yerleştirilmiştir. Yerleştirilen nesnelere renk olarak kırmızı, yeşil ve mavi, şekil olarak kare, yuvarlak ve üçgen,

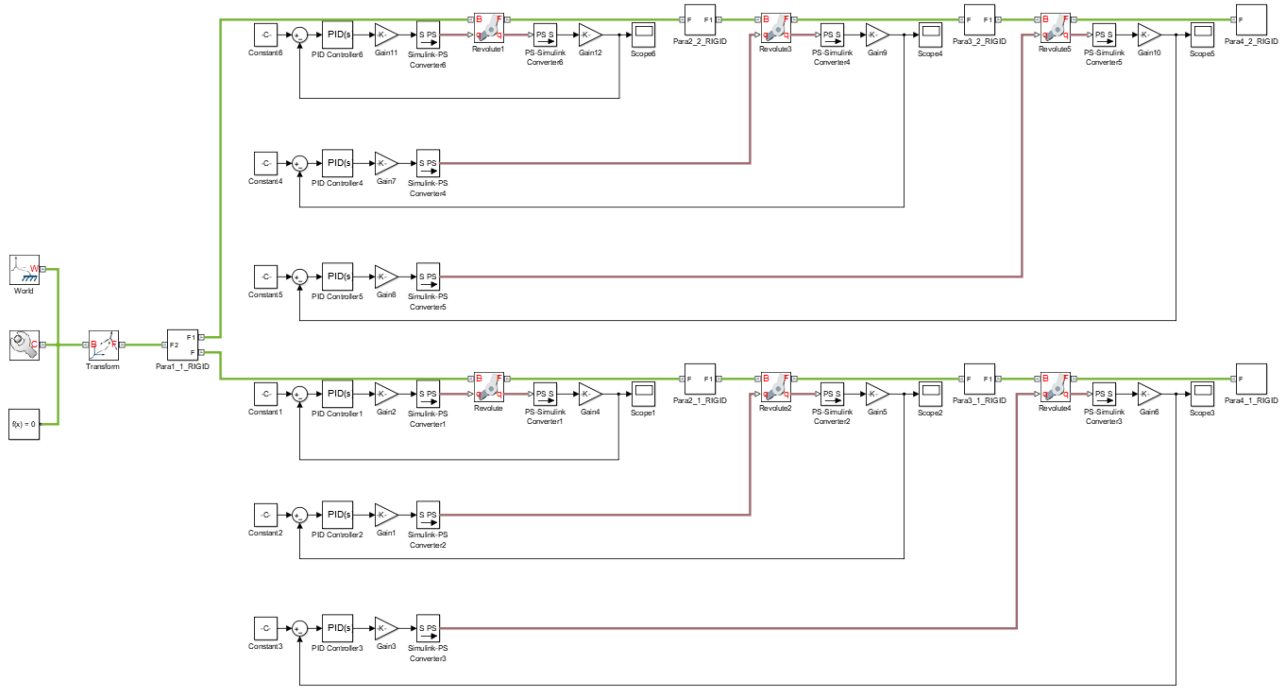
boyut olarak küçük ve büyük olmak üzere ik farklı boyut referans alınmıştır.



Şekil 2.4: Solidworks ile Oluşturulan Robot Kol Tasarımı

## 2.2.2. Matlab Simmechanics

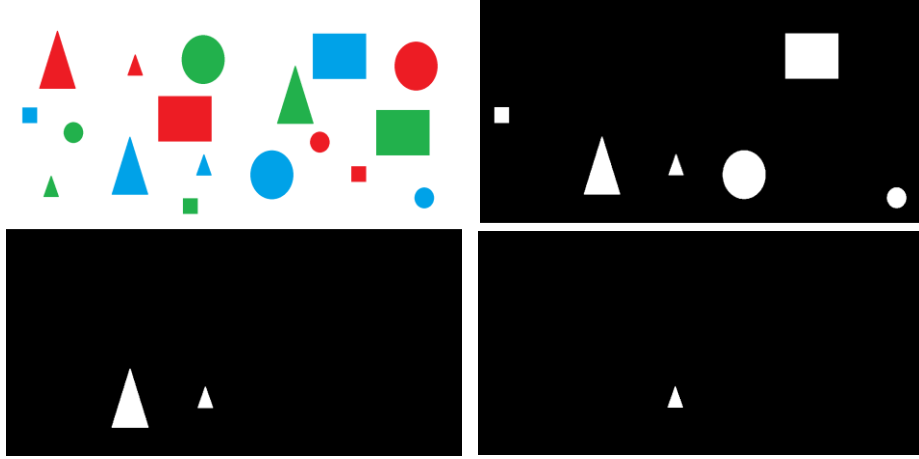
Simülasyon ortamı için SolidWorks ile tasarım oluşturulduktan sonra Matlab'a aktarılmıştır. Matlab GUI ile oluşturulan arayüzden gelen açı bilgileri PID kontrol uygulanarak robot kol eklemlerine gönderilmiştir. PID kontrol için katsayılar deneme-yanılma yöntemiyle bulunmuştur.



Şekil 2.5: Matlab Simmechanics ile Oluşturulan Program

### 3. Tartışma

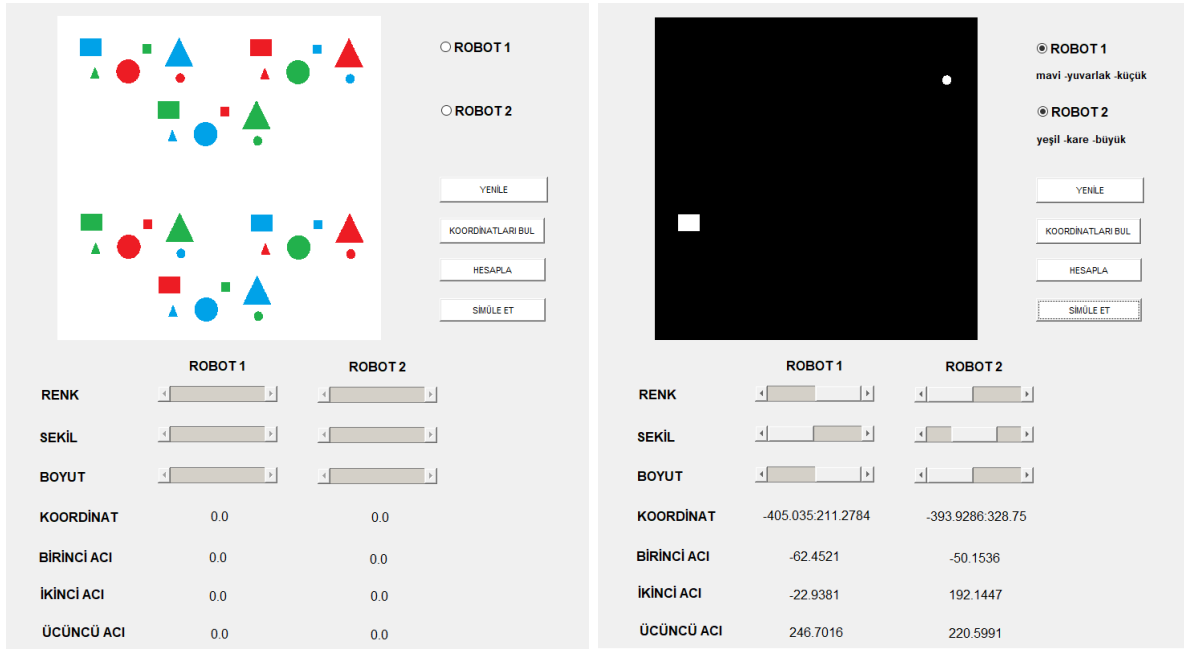
Görüntü işleme teknikleri kullanılarak kullanıcıdan girdi olarak alınan renk, şekil ve boyut bilgileri doğrultusunda nesnenin bulunması ve koordinatlarının tespit edilme işlemi gerçekleştirilmiştir. Şekil 3.1’de yer alan ilk görüntü üzerinden nesne tespit işlemi gerçekleştirilmiştir. Referans olarak alınan renk, şekil ve boyut üzerinden işlem yapılmıştır. Şekil 3.1’deki görüntü üzerinden sadece renk üzerine bir nesne tespit işlemi yapıldığında Şekil 3.1’de yer alan ikinci görüntü elde edilmektedir. Burada referans olarak alınan renk mavidir.



Şekil 3.1: Matlab ile Renk, Şekil ve Boyut Tespit İşlemi

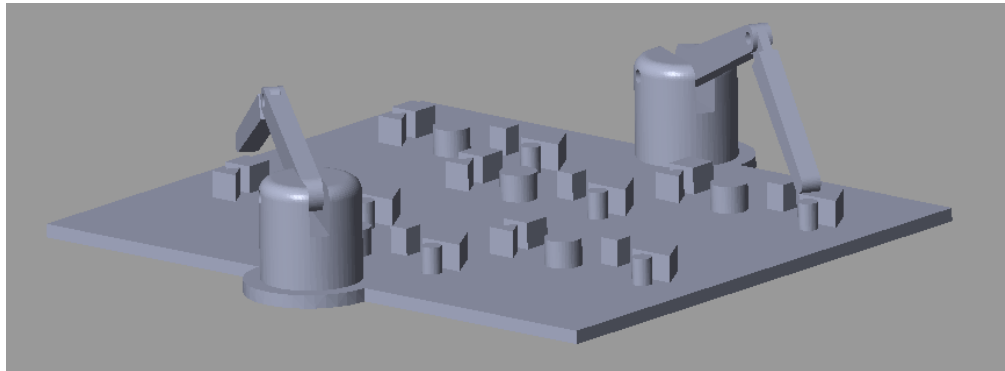
Referans olarak renk alınarak nesne tespit işlemi yapılmıştır. Renk tespiti sonucunda elde edilen görüntü üzerinde şekil tespiti yapıldığında Şekil 3.2’te yer alan görüntü elde edilmektedir. Referans olarak üçgen şekli alınmıştır. Renk ve şekil tespiti üzerinden nesne tespit işlemi yapıldıktan sonra referans olarak boyut alınarak nesne tespit işlemi gerçekleştirilmiştir. Referans olarak küçük boyutlu nesne alınarak elde edilen görüntü Şekil 3.1’de yer almaktadır.

Matlab GUI ile oluşturulan arayüz ile nesne tespit işlemini gerçekleştirilmiştir. Gerçekleştirilen uygulama Şekil 3.2’de gösterilmektedir. Matlab GUI ile oluşturulan arayüzde öncelikle robotlara görev ataması işlemi gerçekleştirilmiştir. Buton ile robotların kullanılıp kullanılmayacağına belirlenerek hangi renk, hangi şekil ve hangi boyuttaki nesnenin tespit edileceğine karar verilmektedir. Ardından arayüzde yer alan “Koordinatları Bul” butonu ile istenilen nesnelerin koordinatları tespit edilmektedir. Elde edilen koordinat değerleri piksel cinsinden olduğundan dolayı simülasyon ortamı için oluşturulan tasarım ölçüleri kullanılarak piksel – metre dönüşümü yapılmıştır. Elde edilen koordinat değerlerine göre robot kollara gönderilecek açılarının bulunması için kinematik analiz yapılmıştır. Arayüzde yer alan “Hesapla” butonu ile kinematik analiz yapılarak elde edilen denklemler kullanılarak açılar elde edilmektedir. Bütün bu işlemler gerçekleştirildikten sonra arayüzdeki “Simüle Et” butonu ile robot kollarının simülasyonu gerçekleştirilmektedir.



Şekil 3.2: Matlab GUI ile Oluşturulan Arayüz ve Nesne Tespit İşlemi

Yapılan çalışmada Matlab GUI ile oluşturulan arayüzde kullanıcı tarafından Robot1 ve Robot2 aktif hale getirilerek görev atamaları gerçekleştirilmiştir. Robot1 için kendi çalışma alanında bulunan renk olarak mavi, şekil olarak yuvarlak ve boyut olarak küçük olan nesneyi, Robot2 için ise renk olarak yeşil, şekil olarak kare ve boyut olarak büyük olan nesneyi tespit etmesi istenmiştir. Renk, şekil ve boyut seçme işlemleri slider ile gerçekleştirilmektedir. Robot1 için istenilen nesnenin koordinatları (-405.035, 211.2784), Robot2 için ise (-393.9286, 328.75) olarak tespit edilmiştir. Ardından tespit edilen koordinatlar arayüze yazdırılarak bulunan koordinatlara göre kinematik analiz yapılarak robot kolların açıları tespit etmesi istenmiştir. Yapılan kinematik analiz sonucunda Robot1 için kola gönderilecek açılar (-62.4521, -22.9381, 246.7016), Robot2 için ise gönderilecek açılar (-50.1536, 192.1447, 220.5991) olarak tespit edilmiştir. Tespit edilen açılar "Simüle Et" butonu ile simülasyon ortamına gönderilerek simülasyon gerçekleştirilmiştir.



Şekil 3.3: Robot Kolların Simülasyon Görüntüsü

**Tablo 3.1:** Robot1 İçin Koordinat, Açılı ve Hata Değerleri

<b>NESNELER</b>	<b>KOORDİNATLAR</b>	<b>AÇILAR</b>	<b>AÇI HATA DEĞERLERİ</b>
Mavi - Kare - Büyük	(396.0714, 142.75)	(70.1801, -27.8529, 259, 2859)	(3.69e-8, 1.46e-8, 1.363e-8)
Yeşil - Kare - Büyük	(156.785, 278.25)	(29.4, -37.0725, 288.4048)	(2.461e-8, 3.103e-8, 2.414e-7)
Yeşil - Üçgen - Küçük	(383.5581, 202.0932)	(62.2157, -26.1883, 255.0903)	(8.612e-7, 3.625e-7, 3.531e-6)
Kırmızı-Yuvarlak - Büyük	(281.0572, 195.2684)	(55.2098, -35.4607, 282.4121)	(8.427e-7, 5.412e-7, 4.31e-6)
Yeşil - Kare- Küçük	(222.1429, 146.5)	(56.5957, -39.6706, 301.5805)	(1.352e-6, 9.478e-7, 7.206e-6)
Kırmızı - Kare - Küçük	(-17.1429, 282)	(-3.4787, -39.059, 297.6223)	(5.871e-9, 6.592e-8, 5.024e-7)
Mavi -Üçgen - Büyük	(123.7902, 163.8921)	(37.0643, -39.9912, 315.6046)	(1.415e-7, 1.527e-7, 1.205e-6)
Yeşil - Üçgen - Büyük	(-115.4955, 299.3921)	(-21.095, -36.9743, 288.0147)	(1.111e-7, 1.948e-7, 1.517e-6)
Kırmızı Yuvarlak - Küçük	(119.965, 209.7784)	(29.7637, -40.2118, 307.3323)	(8.485e-7, 1.146e-6, 8.761e-6)
Mavi - Yuvarlak - Küçük	(-405.035, 211.2784)	(-62.4521, -22.9381, 246.7016)	(1.509e-7, 5.542e-8, 5.961e-7)
Kırmızı - Kare - Büyük	(-128.9286, 144.25)	(-41.7899, -39.5763, 318.2642)	(4.063e-8, 3.848e-8, 3.094e-7)
Mavi - Üçgen - Küçük	(144.2723, 337.5932)	(23.1397, -33.3932, 275.5781)	(2.38e-7, 3.434e-7, 2.834e-6)
Mavi - Yuvarlak - Büyük	(41.7714, 330.7684)	(7.1976, -36.1168, 284.7583)	(5.344e-9, 2.682e-8, 2.114e-7)
Mavi - Kare - Küçük	(-302.8571, 148)	(-63.9562, -35.8478, 283.7833)	(3.596e-6, 2.01e-6, 1.591e-6)
Kırmızı - Üçgen - Küçük	(-141.4419, 203.5932)	(-34.7888, -40.1195, 305.8774)	(4.221e-8, 4.868e-8, 3.711e-7)
Yeşil - Yuvarlak - Küçük	(-119.3207, 345.2784)	(-19.064, -33.5548, 276.0884)	(3.392e-8, 5.989e-8, 4.92e-7)
Yeşil - Yuvarlak - Büyük	(-243.9426, 196.7684)	(-51.1098, -37.4475, 289.933)	(3.109e-6, 2.278e-6, 1.764e-5)
Kırmızı - Üçgen - Büyük	(-401.2098, 165.3921)	(-67.5969, -26.1335, 254.9458)	(1.348e-9, 5.214e-10, 5.088e-8)

**Tablo 3.2:** Robot2 İçin Koordinat, Açılı ve Hata Değerleri

<b>NESNELER</b>	<b>KOORDİNATLAR</b>	<b>AÇILAR</b>	<b>AÇI HATA DEĞERLERİ</b>
Mavi - Kare - Büyük	(131.0714, 327.5)	(21.8273, 214.6452, 279.6296)	(2.346e-5, 2.087e-7, 2.718e-7)
Yeşil - Kare - Büyük	(-393.9286, 328.75)	(-50.1536, 192.1447, 220.591)	(6.194e-5, 1.134e-8, 1.302e-8)
Yeşil - Üçgen - Küçük	(-381.4152, 269.4068)	(-54.7651, 201.3612, 242.7446)	(9.361e-5, 2.787e-6, 3.36e-6)
Kırmızı-Yuvarlak - Büyük	(-278.9143, 276.2316)	(-45.2769, 210.9432, 268.1918)	(5.281e-5, 3.22e-6, 4.093e-6)
Yeşil - Kare- Küçük	(19.2857, 189.5)	(5.8111, 219.4398, 318.9275)	(1.354e-5, 3.703e-7, 5.383e-7)
Kırmızı - Kare - Küçük	(-220, 325)	(-34.095, 210.9527, 268.2193)	(3.397e-5, 5.04e-6, 6.408e-6)
Mavi -Üçgen - Büyük	(117.6384, 172.1079)	(34.3532, 220.0674, 314.9126)	(1.553e-5, 1.159e-6, 1.659e-6)
Yeşil - Üçgen - Büyük	(-121.6473, 307.6079)	(-21.5769, 216.3026, 285.4435)	(2.699e-9, 8.26e-7, 1.09e-6)
Kırmızı Yuvarlak - Küçük	(407.1778, 260.2216)	(57.4179, 198.568, 235.8773)	(1.578e-5, 4.798e-7, 5.699e-7)
Mavi - Yuvarlak - Küçük	(-117.8222, 261.7216)	(-24.2364, 218.8602, 296.5231)	(3.859e-5, 6.239e-6, 8.453e-6)
Kırmızı - Kare - Büyük	(-154.6429, 193.25)	(-38.6676, 220.1263, 305.9699)	(2.67e-6, 1.842e-7, 2.561e-7)
Mavi - Üçgen - Küçük	(-142.1295, 133.9068)	(-46.7062, 219.652, 317.8634)	(1.526e-5, 2.259e-6, 3.269e-6)
Mavi - Yuvarlak - Büyük	(-39.6286, 140.7316)	(-15.7267, 215.4217, 328.5674)	(5.395e-6, 1.599e-7, 2.439e-7)
Mavi - Kare - Küçük	(305, 323.5)	(43.314, 204.702, 251.2107)	(0.1672e-4, 1.148e-5, 1.408e-5)
Kırmızı - Üçgen - Küçük	(143.5848, 267.9068)	(28.1891, 218.0002, 292.3227)	(2.129e-5, 2.645e-7, 3.547e-7)
Yeşil - Yuvarlak - Küçük	(121.4636, 126.2216)	(43.8995, 218.5106, 322.297)	(0.1489e-4, 3.887e-7, 5.734e-7)
Yeşil - Yuvarlak - Büyük	(246.0857, 274.7316)	(41.8518, 213.2401, 275.0978)	(0.9031e-4, 1.297e-5, 1.673e-5)
Kırmızı - Üçgen - Büyük	(403.3527, 306.1079)	(52.8048, 193.7984, 224.4761)	(1.625e-6, 3.866e-9, 4.478e-9)

Tabloya baktığımızda robot kollar için yapılan nesne tespit işleminde bütün nesnelere tespit edilmiş ve koordinatları bulunmuştur. Bunun yanı sıra kinematik analiz sonucu robot kol eklemlerine gönderilecek tüm açılarda bulunmuştur. Eklemlere bulunan açılıları gönderirken PID kontrol uygulanmıştır. PID kontrol sonucu oluşan hata değerlerine bakıldığında çok düşük olduğu



görülmektedir. Hesaplanan açı değerlerine çok yakın bir açı gönderildiği hata değerlerinin çok düşük olmasıyla birlikte görülmektedir. En yüksek hata değeri 0.1489e-4 olurken en düşük hata değeri 5.214e-10 olduğu gözlemlenmiştir.

#### 4. Sonuçlar

Bir otomasyon hattında görüntü işleme temelli parça seçim ve ayrıştırma işlemlerinde çoklu robot kollarında görev paylaşımı üzerine yapılan çalışmanın ön hazırlık çalışmasında Matlab ile kırmızı, yeşil ve mavi olmak üzere farklı renklerde, kare, üçgen ve yuvarlak olmak üzere farklı şekillerde ve küçük ve büyük olmak üzere farklı boyutlarda nesnelere tespiti başarıyla gerçekleştirilmiştir. Farklı özelliklere sahip nesnelere kullanıcı girdisi ile istenilen nesnelere tespit edilmesi ve koordinat bilgilerinin bulunması başarıyla gerçekleştirilmiştir. Koordinat bilgileri kullanılarak kinematik analiz sonucu açılar elde edilip simülasyon ortamında robot kol eklemlerine başarıyla gönderilmiştir. Simülasyon ortamında ise PID kontrol uygulanıp robot kolların hareketleri tespit edilen konuma başarıyla sağlanmıştır. PID kontrol sonucu robot kol eklemlerinin gitmesi gereken açılarda hata değerlerinin çok düşük olduğu gözlemlenmiştir. Özetlersek robot kollar arasında görev paylaşımı gerçekleştirilmiş ve bunun sonucunda istenilen nesnenin konumuna robot kollar başarıyla hareket etmiştir.

#### Referanslar

- [1] Basile, F., Caccavale, F., Chiacchio, P., Coppola, J. and Curatella, C., 2012. Taskoriented motion planning for multi-arm robotic systems, *Robotics and Computer-Integrated Manufacturing*, 28(5), 569-582.
- [2] SARIEL, Sanem. *Çoklu-robotların Ortak Çalışması ve Koordinasyonu İçin Tümlşik Bir Planlama, Görev Atama ve Yürütme Mimarisi*. 2007. PhD Thesis. Fen Bilimleri Enstitüsü.
- [3] DÖNMEZ, Emrah; KOCAMAZ, Adnan Fatih. Çoklu Hedeflerin Çoklu Robotlara Paylaştırılması İçin Bir Yük Dengeleme Sistemi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 8.2: 533-548.
- [4] ÇAYIRPUNAR, Ömer. *Çoklu robot sistemlerinde robotlar arası haberleşme ve iş birliği kullanılarak arama verimliliğinin artırılması*. 2009. Master's Thesis. TOBB Ekonomi ve Teknoloji Üniversitesi-Fen Bilimleri Enstitüsü-Bilgisayar Mühendisliği Bölümü-Bilgisayar Mühendisliği Anabilim Dalı\.
- [5] CUŞKUN, Y., DUMAN, F., BASIK, H., GÜN, F., KAPLAN, K., & ERTUNÇ, H. M. (2016). Görüntü İşleme Tabanlı 4 Eksenli Çok Amaçlı Robot Mekanizması. In *Conference Paper, Mechatronics Engineering Department, Kocaeli University. Kocaeli, Turkey*.

- [6] THENMOZHI, K.; REDDY, U. Srinivasulu. Image processing techniques for insect shape detection in field crops. In: *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2017. p. 699-704.
- [7] WALI-UR-RAHMAN, Md, et al. Robotic Arm with Proximity and Color Detection. In: *2018 IEEE 7th International Conference on Power and Energy (PECon)*. IEEE, 2018. p. 322-326.
- [8] KAYMAK, Ahmet Murat Murat; ÖRNEK, Mustafa Nevzat Nevzat; KAHRAMANLI, Humar. GÖRÜNTÜ İŞLEME TEKNOLOJİLERİNİN ELMA BAHÇELERİNE YÖNELİK KULLANIM ÖRNEĞİ. *Uluborlu Mesleki Bilimler Dergisi*, 2.1: 17-26.
- [9] ALITAPPEH, Reza Javanmard; JEDDISARAVI, Kossar. Multi-robot exploration in task allocation problem. *Applied Intelligence*, 2021, 1-23.
- [10] DOS REIS, Wallace Pereira Neves; LOPES, Gustavo Leite; BASTOS, Guilherme Sousa. An arrovian analysis on the multi-robot task allocation problem: Analyzing a behavior-based architecture. *Robotics and Autonomous Systems*, 2021, 103839.
- [11] WOOSLEY, Bradley, et al. Multi-robot information driven path planning under communication constraints. *Autonomous Robots*, 2020, 44.5: 721-737.
- [12] MARCOTTE, Ryan J., et al. Optimizing multi-robot communication under bandwidth constraints. *Autonomous Robots*, 2020, 44.1: 43-55.
- [13] ALSAYEGH, Murtadha, et al. Lightweight Multi-robot Communication Protocols for Information Synchronization. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020. p. 11831-11837.
- [14] SERRA-GÓMEZ, Alvaro, et al. With whom to communicate: learning efficient communication for multi-robot collision avoidance. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020. p. 11770-11776.
- [15] FAN, Tingxiang, et al. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 2020, 39.7: 856-892.
- [16] PATEL, Jayam, et al. Direct and Indirect Communication in Multi-Human Multi-Robot Interaction. *arXiv preprint arXiv:2102.00672*, 2021.
- [17] THABIT, Sahib; MOHADES, Ali. Multi-robot path planning based on multi-objective particle swarm optimization. *IEEE Access*, 2018, 7: 2138-2147.