T.C.

Fırat Üniversitesi

Yazılım Mühendisliği Bölümü

|  |
| --- |
| Görüntü İşleme Tabanlı Hedef İmha Aracı |
| Hasan Gökhan Mercan |
| 26 Haziran - 2025 |

T.C.

Fırat Üniversitesi

Yazılım Mühendisliği Bölümü

|  |  |
| --- | --- |
| Başlığı: | Görüntü İşleme Tabanlı Hedef İmha Aracı |
| Yazarı: | Hasan Gökhan MERCAN |
| Proje Danışmanı: | Engin AVCI |
| Teslim Tarihi: | 26.06.2025 |

Beyan

Fırat Üniversitesi Yazılım Mühendisliği Bölümü bitirme projesi yazım kurallarına uygun olarak hazırladığım “Bitirme Projesi Başlığı Her Kelimenin İlk Harfi Büyük Olarak Buraya Yazılmalıdır” Başlıklı proje dokümanımın içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

26.05.2025

Hasan Gökhan MERCAN

**Önsöz**

İlk paragrafta bitirme projesinin konusunun önemi, zorlukları, sınırları ve isteklendirme (motivasyon) faktörleri hakkında bilgi verilmelidir. Bu bilgiler kesinlikle bitirme projesinin yazarı tarafından yazılmalı herhangi bir alıntı yapılmamalıdır. Bu açıklamalar için gerek duyuluyorsa 2-3 kısa paragraf oluşturulabilir. Bu bilgiyi izleyen yeni bir paragrafta, önce bitirme projesi çalışmalarına doğrudan katkı saylayan ve sonra dolaylı katkısı olan kişi, kurum ve kuruluşlara yaptıkları katkılar açıkça ifade edilmek şartıyla teşekkür edilmelidir.

Bitirme projesi, bir proje desteği veya etik kurul izni ile hazırlanmış ise, Önsöz metninin son paragrafında, bitirme projesi çalışmalarına protokol numaralı proje ile maddi destek sağlayan ve/veya yazılı izin (olur) veren kurum/kuruluşlara protokol numaraları ve/veya ilgili yazı tarihi belirtilerek aşağıdaki gibi teşekkür edilir. Böyle bir durum yok ise bu paragraf silinir.

Hasan Gökhan MERCAN

Elazığ, 2025

İçindekiler

Sayfa

[Önsöz iv](#_Toc44596524)

[İçindekiler v](#_Toc44596525)

[Özet vi](#_Toc44596526)

[Şekiller Listesi vii](#_Toc44596527)

[Tablolar Listesi viii](#_Toc44596528)

[Ekler Liste ix](#_Toc44596529)

[Simgeler ve Kısaltmalar x](#_Toc44596530)

[1. Giriş 1](#_Toc44596531)

1. **[Materyal ve Metot](#_Toc44596532)** […………………………………………………………………….2  
   2.1. Sistem Tasarımı……………………………………………………………………..2  
   2.2. Yazılım Mimarisi ve Görüntü İşleme Süreci …………………………………..3  
   2.2.1. ESP32-CAM Görüntü Yayını …………………………………………………...3  
   2.2.2. Orange Pi 3 LTS Görüntü İşleme ………………………………………………4  
   2.2.3. Arduino Mega Motorların Kontrolü ……………………………………….5  
   2.3. Görüntü İşleme Yöntemleri ………………………………………………...….12  
   2.3.1. HSV Tabanlı Görüntü İşleme Algoritması …………………………..…….10  
   2.4. Performans Metrikleri ……………………………………...………………….12  
   2.5. Deneysel Çıktılar ………………………………………………………….……13](#_Toc44596532)
2. **[Deneysel Kurulumlar](#_Toc44596532)** [………………………………………………………………14](#_Toc44596532)
3. **[Bulgular ve Tartışma](#_Toc44596532)** [………………………………………………………………15](#_Toc44596532)
4. **[Sonuçlar](#_Toc44596532)** [………………………………………………………………………………...15](#_Toc44596532)**[Kaynaklar](#_Toc44596532)** […………………………………………………………………………….15](#_Toc44596532)**[Ekler](#_Toc44596532)** [………………………………………………………………………………...…15](#_Toc44596532)

Özet

Dummy

Görüntü İşleme Tabanlı Hedef İmha Aracı

Hasan Gökhan MERCAN

Fırat Üniversitesi

Yazılım Mühendisliği Bölümü

Bu çalışma, görüntü işleme tekniklerine dayalı, kırmızı bir topu takip eden otonom bir araç sisteminin geliştirilmesini ele almaktadır. Sistem, Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN) ile gerçek zamanlı görüntü analizi ve sınıflandırma yetkinliklerini hedeflemiş, ancak donanım kısıtlamaları nedeniyle Hue-Saturation-Value (HSV) renk uzayına dayalı algoritmalar kullanılmıştır. Görüntü toplama için ESP32-CAM modülü, görüntü işleme için Orange Pi 3 LTS ve motor kontrolü için Arduino Mega ile L298N sürücüsü kullanılmıştır. Araç, dört adet 12V DC motorla tahrik edilmiş; 6.000 mAh ve 12.000 mAh bataryalarla desteklenmiştir.

Test sonuçları, sistemin kontrollü ortamlarda kırmızı topu başarıyla takip ettiğini, ancak dinamik aydınlatma ve hızlı hareket senaryolarında performansın düştüğünü göstermiştir. Gelecekte, daha güçlü donanımlar ve optimize edilmiş CNN modelleriyle performansın artırılması hedeflenmektedir. Bu çalışma, düşük maliyetli gömülü sistemlerde görüntü işleme uygulamalarına katkı sağlamaktadır.

**Anahtar Kelimeler:** CNN, HSV Renk Kodları.

Şekiller Listesi

Sayfa

[**Şekil 2.1.** Ana bölüm başlığı oluşturmadan önce yeni bir sayfada yeni bir bölüm başlatma 3](#_Toc44596539)

[**Şekil 2.2.** Stil galerisinden “Başlık 1” seçilerek ana bölüm başlığı için otomatik numara alma ve başlık yazımı yapılandırma işlemi 3](#_Toc44596540)

[**Şekil 3.1.** Bir şeklin öğeleri; nesne alanı, etiket ve başlık 4](#_Toc44596541)

[**Şekil 3.2.** Otomatik şekil etiketi eklemek için Word programında “Başvuru” kullanımı 5](#_Toc44596542)

[**Şekil 3.3.** İki satırdan oluşan bu şekil başlık yazısı 10 punto yazı büyüklüğü ve tek aralık ayarı ile yazılmıştır. Ayrıca, Girinti-Özel-Asılı = 1,7 cm olarak ayarlanmıştır. 5](#_Toc44596543)

[**Şekil 5.1.** Bulguların sunulmasında kullanılabilir bir şekil için örnek ölçeklendirme 8](#_Toc44596544)

Tablolar Listesi

Sayfa

[**Tablo 3.1.** Örnek bir tablo yapısı 6](#_Toc44596459)

[**Tablo 5.1.** Numunelerin hesaplanan bant aralıkları (Örnektir) 9](#_Toc44596460)

[**Tablo 5.2.** Güneş pili numunelerine ait ölçülen ve hesaplanan elektriksel parametreler (Örnektir) 9](#_Toc44596461)

[**Tablo 5.3.** Numara sistemiyle kaynak gösterimleri içeren bir tablo örneği 9](#_Toc44596462)

[**Tablo 5.4.** Soyadı-Yıl sistemiyle kaynak gösterimleri içeren bir tablo örneği 9](#_Toc44596463)

Ekler Listesi

Sayfa

[Ek- 1: Mendeley Desktop Programının Bilgisayara Kurulumu 13](#_Toc44596466)

[Ek- 2: Mendeley Programında Kaynak Gösterme Yönteminin Seçilmesi 14](#_Toc44596467)

Simgeler ve Kısaltmalar

**Simgeler**

: Fotoakım

VOC : Açık devre gerilimi

**Kısaltmalar**

CIGS : Bakır İndiyum Galyum Diselenit (Cu(InGe))

DSSCs : Boya duyarlı güneş pilleri

1. Giriş

Otonom araç teknolojileri, yapay zeka, görüntü işleme ve robotik sistemlerin entegrasyonuyla modern mühendislik ve bilgisayar bilimlerinin yenilikçi alanlarından biridir. Görüntü işleme, otonom araçların nesne takibi ve navigasyon gibi kritik görevlerinde merkezi bir rol oynar. Bu bağlamda, Evrişimsel Sinir Ağları (CNN), görüntü verilerinden anlamlı özellikler çıkararak gerçek zamanlı sınıflandırma görevlerinde yüksek doğruluk sağlar. CNN’ler, evrişim, havuzlama ve tam bağlantılı katmanlarıyla, görüntülerdeki uzaysal ilişkileri koruyarak karmaşık desenleri algılar [1]. Evrişim katmanları yerel özellikleri (örneğin, renk ve şekil) tespit ederken, havuzlama katmanları hesaplama verimliliğini artırır; tam bağlantılı katmanlar ise özellikleri sınıflandırmada kullanır [2].

CNN’lerin başarısı, büyük veri setleri ve yüksek hesaplama gücüne dayanır. TensorFlow ve PyTorch gibi çerçeveler, GPU’ların gelişimiyle karmaşık CNN modellerinin tasarlanmasını sağlamıştır [3]. Ancak, bu algoritmaların yüksek hesaplama gereksinimleri, düşük maliyetli gömülü sistemlerde (örneğin, Orange Pi veya Arduino) uygulanabilirliklerini sınırlar. Literatürde, otonom araçlar genellikle yüksek performanslı işlem birimleri (GPU, TPU) ile geliştirilir; ancak bu çözümler maliyet ve enerji tüketimi açısından pratik değildir [4]. Buna karşılık, düşük maliyetli gömülü sistemler ekonomik bir alternatif sunar, ancak CNN tabanlı algoritmaları gerçek zamanlı desteklemede yetersiz kalır.

Bu çalışmada, kırmızı bir topu takip eden, görüntü işleme tabanlı otonom bir araç sistemi geliştirilmiştir. Sistem, ekranı üçe bölerek (sağ, sol, orta) topun konumuna göre hareket eder ve top yakın olduğunda durur. CNN tabanlı bir model TensorFlow ile eğitilmeye çalışılmış, ancak Orange Pi 3 LTS’nin sınırlı işlem kapasitesi nedeniyle gerçek zamanlı çalıştırılamamış; bunun yerine HSV tabanlı renk sınıflandırma algoritmaları kullanılmıştır. Çalışma, donanım kısıtlamalarının CNN tabanlı sistemler üzerindeki etkilerini analiz eder ve düşük maliyetli platformlarda otonom araç uygulamaları için pratik bir örnek sunar.

1. Materyal ve Mototlar
   1. Sistem TAsarımı

Bu çalışmada, kırmızı bir topu takip eden otonom bir araç sistemi geliştirilmiştir. Sistem, görüntü toplama, işleme, motor kontrolü ve güç yönetimi için donanım ve yazılım bileşenlerini entegre eder. Görüntü ekranı yatay olarak üçe bölünerek (sağ, sol, orta) topun konumu tespit edilmiş; topun yakınlığı, kontur alanına göre belirlenerek araç hareketleri (sağa dön, sola dön, ileri git, dur) kontrol edilmiştir. Arduino Mega, Orange Pi 3 LTS ile seri iletişim yoluyla motor kontrol komutlarını alır. Şekil 2.1, sistemin donanım mimarisini gösterir.

* + 1. ESP32-CAM Modülü

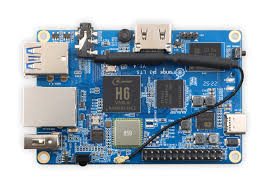
*Tanım:* ESP32-CAM, Espressif Systems tarafından geliştirilen, OV2640 kamera sensörlü, Wi-Fi ve Bluetooth destekli düşük maliyetli bir mikrodenetleyici modülüdür.  
*Amaç:* Gerçek zamanlı görüntü toplama için kullanılır. 640x480 piksel RGB görüntüler yakalayarak Orange Pi 3 LTS’ye HTTP protokolü üzerinden akış sağlar. Kompakt yapısı, gömülü sistemlerde görüntü işleme için uygundur.

****

**Şekil 2.2:** ESP32-CAM modülü.

* + 1. Orange Pi 3 LTS

*Tanım:* Orange Pi 3 LTS, Allwinner H6 dört çekirdekli Cortex-A53 işlemciye sahip, 2 GB RAM’li, Linux tabanlı bir tek kart bilgisayardır.  
*Amaç:* Ana işlem birimi olarak, ESP32-CAM’den gelen görüntü akışını işler, HSV tabanlı algoritmayı çalıştırır ve kırmızı topun konumunu (sağ, sol, orta) tespit eder. Seri iletişim yoluyla Arduino Mega’ya motor kontrol komutları gönderir. Sınırlı işlem kapasitesi nedeniyle CNN modeli gerçek zamanlı çalıştırılamamış, HSV algoritması tercih edilmiştir.

****

**Şekil 2.3:** Orange Pi 3 LTS’nin sistemdeki bağlantı düzeni ve işlem birimi yerleşimi.

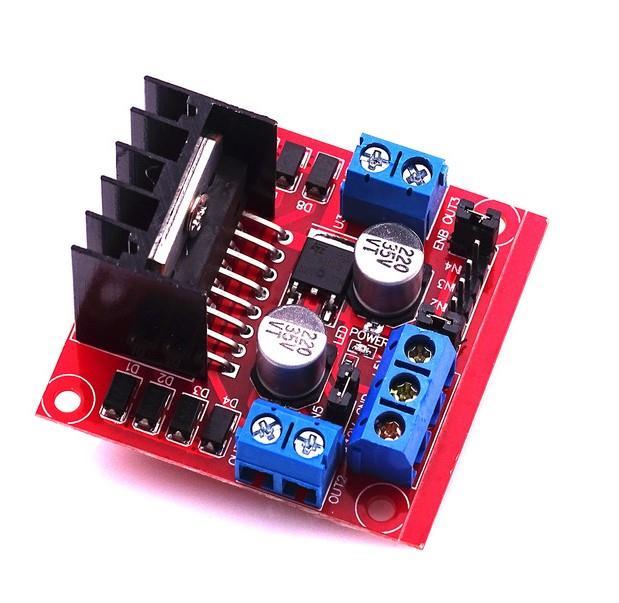
* + 1. Arduino Mega 2560

*Tanım:* Arduino Mega 2560, ATmega2560 tabanlı, çok sayıda giriş/çıkış pini sunan bir mikrodenetleyici kartıdır.  
*Amaç:* Motor kontrol birimi olarak, Orange Pi 3 LTS’den seri iletişimle gelen komutları (örneğin, ‘0’ için dur, ‘1’ için ileri, ‘3’ için sağa dön, ‘4’ için sola dön) alır ve L298N sürücüsüne sinyaller göndererek DC motorları kontrol eder.

**  
Şekil 2.4:** Arduino Mega 2560.

* + 1. L298N Çift Kanallı DC Motor Sürücüsü

*Tanım:* L298N, iki DC motoru veya bir step motoru kontrol eden, H-köprüsü tabanlı bir motor sürücü modülüdür.  
*Amaç:* Dört adet 12V DC motorun hızını ve yönünü kontrol eder. Arduino Mega’dan gelen PWM sinyalleriyle aracın hareketini (sağa, sola, ileri) sağlar.

**  
Şekil 2.5:** L298N motor sürücüsü.

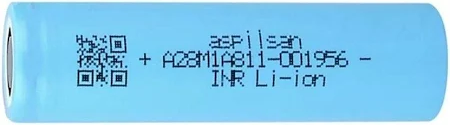
* + 1. 12V DC Motorlar (4 Adet)

*Tanım:* 12V DC motorlar, elektrik enerjisini mekanik harekete dönüştüren yüksek torklu cihazlardır.  
*Amaç:* Aracın tahrik sistemini oluşturur. L298N sürücüsüyle kontrol edilerek diferansiyel tahrikle yön kontrolü sağlar.

**  
Şekil 2.6:** 12V DC motor.

* + 1. 6.000 mAh Batarya (İşlem Birimi İçin)

Tanım: 6.000 mAh kapasiteli lityum-iyon batarya, taşınabilir cihazlar için güç kaynağıdır.  
Amaç: Orange Pi 3 LTS ve ESP32-CAM’in güç ihtiyacını karşılar. Gerilim regülatörü ile 5.2 V çıkış sağlar.

**  
Şekil 2.7:** 6.000 mAh bataryanın bağlantı şeması ve işlem birimi güç entegrasyonu.

* + 1. 12.000 mAh Batarya (Motorlar İçin)

*Tanım:* 12.000 mAh kapasiteli lityum-iyon batarya, yüksek akım gerektiren uygulamalar için tasarlanmıştır.  
*Amaç:* DC motorlar ve L298N sürücüsünün güç ihtiyacını karşılar. Gerilim regülatörü ile güç dalgalanmaları önlenir.



**Şekil 2.812.000 mAh Batarya (Motorlar İçin)**

* + 1. Gerilim Regülatörü

*Tanım:* Gerilim regülatörü (LM2596 DC-DC buck converter), giriş gerilimini sabit bir çıkışa dönüştürür.  
*Amaç:* Bataryalardan gelen 7.4-8 V gerilimi 5.2 V’a düşürerek Orange Pi 3 LTS, ESP32-CAM ve motorlar için stabil güç sağlar.

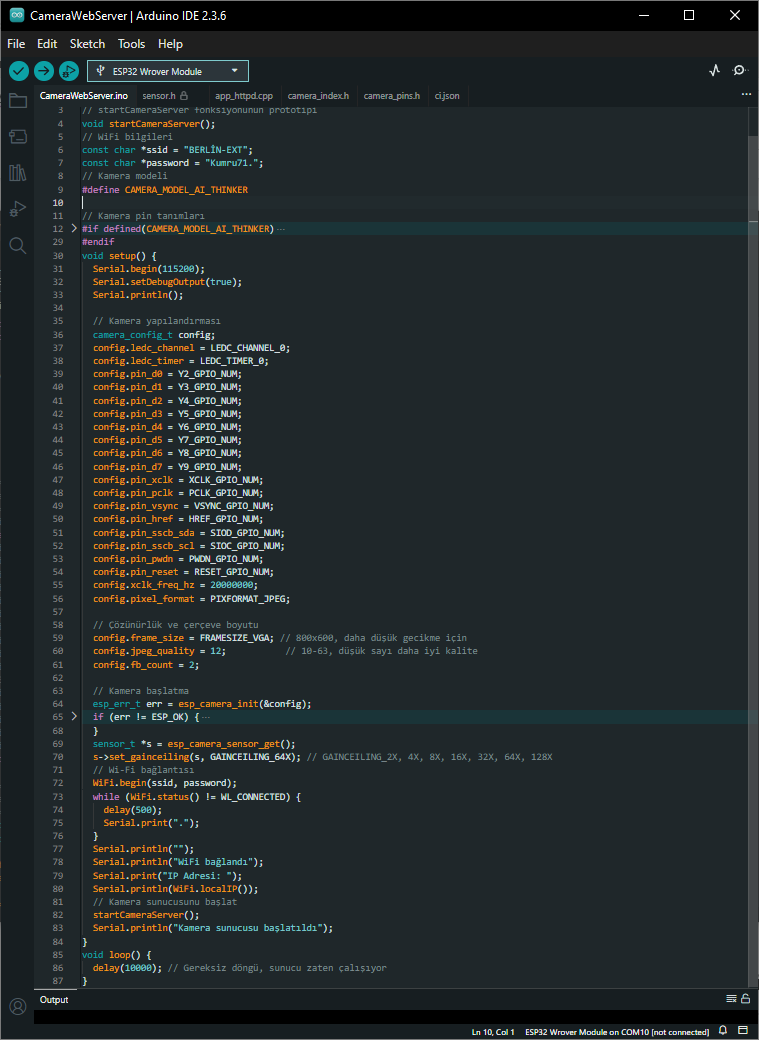
**  
Şekil 2.9:** Gerilim regülatörü.

**2.2. Yazılım Mimarisi ve Görüntü İşleme Süreci**

Sistemin yazılım mimarisi, ESP32-CAM, Orange Pi 3 LTS ve Arduino Mega arasında koordineli bir iş akışı sağlar. Aşağıda, her bir bileşenin yazılım işlevleri açıklanmıştır:

**2.2.1. ESP32-CAM Görüntü Yayını**

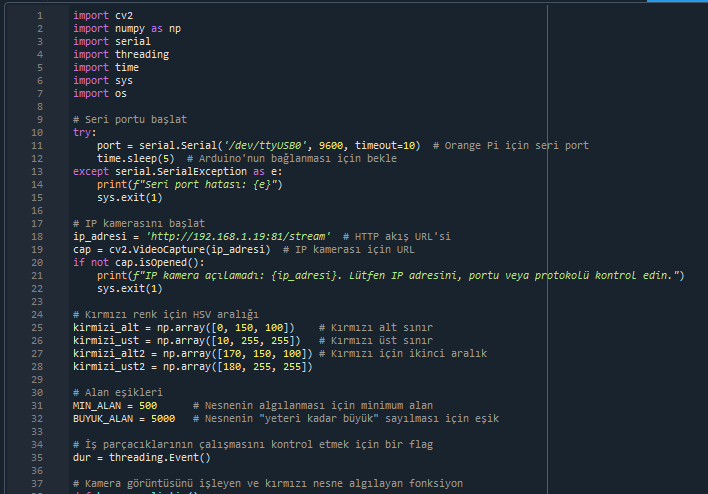
ESP32-CAM, OV2640 kamera sensörüyle 640x480 piksel çözünürlüğünde JPEG formatında görüntü yakalar ve Wi-Fi üzerinden HTTP protokolüyle Orange Pi 3 LTS’ye akış sağlar. setup() fonksiyonu, kamera yapılandırmasını (çözünürlük, JPEG kalitesi, kare hızı) başlatır ve Wi-Fi bağlantısını kurar. startCameraServer() fonksiyonu, bir web sunucusu oluşturarak görüntü akışını IP adresi üzerinden yayınlar. Bu, Orange Pi 3 LTS’nin gerçek zamanlı görüntü verilerine erişmesini sağlar. Görüntü kalitesini optimize etmek için set\_gainceiling() gibi yöntemler kullanılarak sensör ayarları yapılmıştır.

****

**Şekil 2.10:** ESP32-CAM’in yayın kodu.

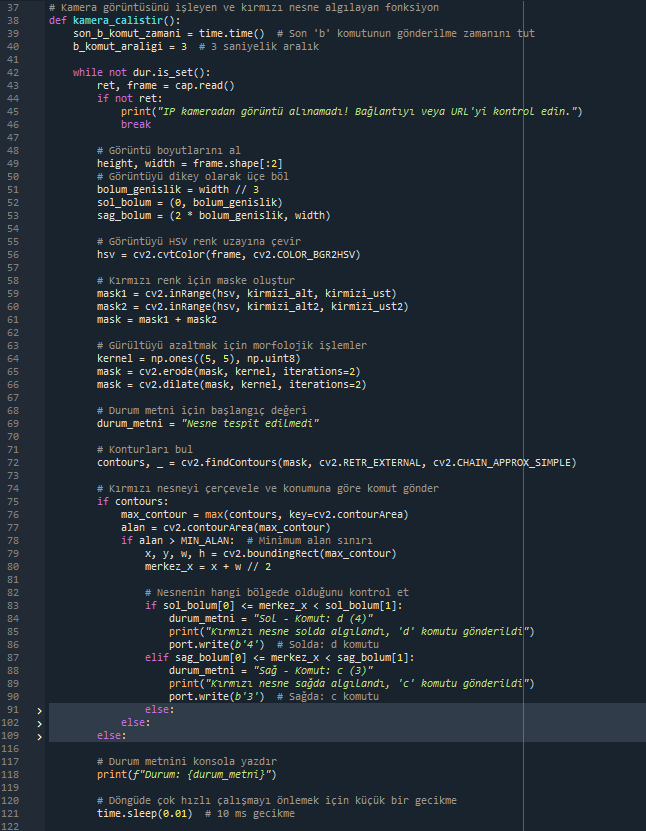
**2.2.2. Orange Pi 3 LTS Görüntü İşleme**

Orange Pi 3 LTS, Python tabanlı bir yazılım ile ESP32-CAM’den gelen görüntü akışını işler. OpenCV kütüphanesi kullanılarak geliştirilen kamera\_calistir() fonksiyonu, görüntüleri RGB’den HSV renk uzayına dönüştürür ve kırmızı topu tespit eder.

****

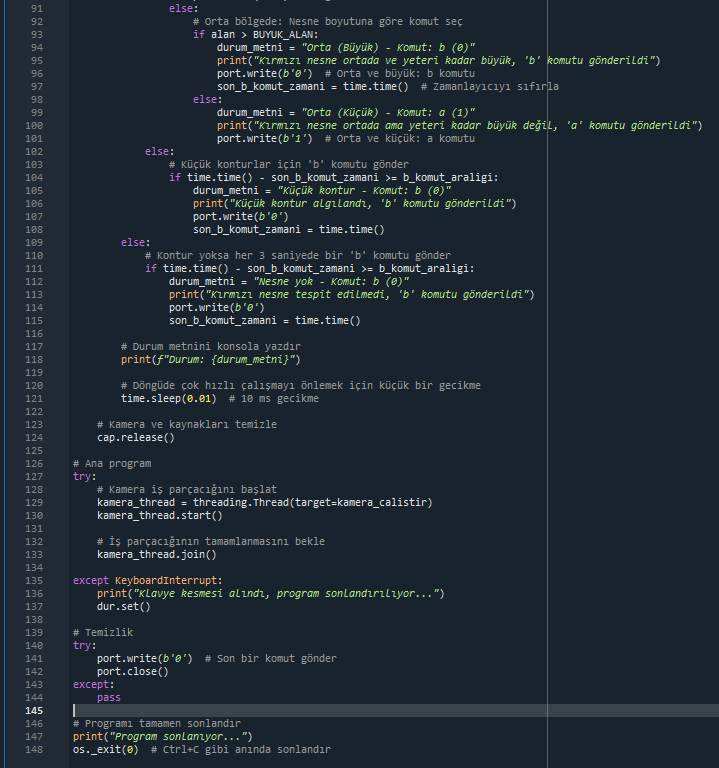
**Şekil 2.11.1:** Orange Pi 3 LTS’nin görüntü işleme kodu.

Görüntü yatay olarak üçe bölünerek (sağ, sol, orta) topun konumu belirlenir. cv2.inRange() yöntemiyle kırmızı renk için iki HSV eşik aralığı tanımlanır, cv2.erode() ve cv2.dilate() ile gürültü azaltılır, cv2.findContours() ile en büyük kontur seçilerek topun merkezi hesaplanır.

****

**Şekil 2.11.2:** Orange Pi 3 LTS’nin görüntü işleme kodu.

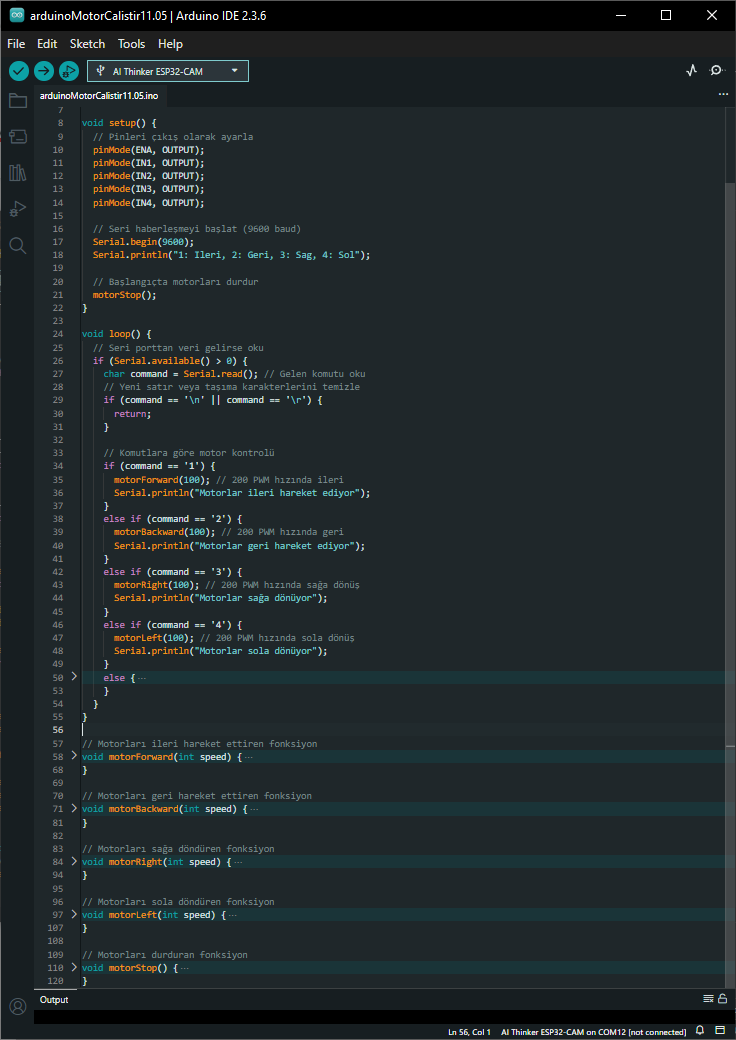
Topun konumu ve kontur alanı (minimum 500 piksel, büyük eşik 5000 piksel) temel alınarak hareket komutları (0 için dur, 1 için ileri, 3 için sağa dön, 4 için sola dön) belirlenir. Bu komutlar, /dev/ttyUSB0 portu üzerinden 9600 baud hızıyla seri iletişimle Arduino Mega’ya gönderilir.

****

**Şekil 2.11.3**: Orange Pi 3 LTS’nin görüntü işleme kodu.

**2.2.3. Arduino Mega Motorların Kontrolü**

Arduino Mega, Orange Pi 3 LTS’den seri iletişimle gelen komutları alır ve L298N motor sürücüsüne PWM sinyalleri göndererek dört adet 12V DC motoru kontrol eder. setup() fonksiyonu, seri iletişimi başlatır ve motor pinlerini yapılandırır. Gelen komutlara göre (0 için dur, 1 için ileri, 3 için sağa dön, 4 için sola dön), motorların yönü ve hızı ayarlanır. Diferansiyel tahrik sistemiyle, araç kırmızı topa yaklaşmak için uygun hareketleri gerçekleştirir.

****

**Şekil 2.12:** Arduino Mega’nın motor kontrol kodu.

**2.3. Görüntü İşleme Yöntemleri**

**2.3.2. HSV Tabanlı Görüntü İşleme Algoritması**

CNN modelinin gerçek zamanlı çalıştırılamaması nedeniyle, kırmızı topun tespiti için OpenCV kütüphanesiyle HSV tabanlı bir algoritma geliştirilmiştir. Algoritmanın temel bileşenleri şunlardır:

* *Renk Algılama:*Görüntü, RGB’den HSV renk uzayına dönüştürülerek kırmızı renk için iki eşik aralığı ([0, 150, 100] - [10, 255, 255] ve [170, 150, 100] - [180, 255, 255]) tanımlanmıştır.
* *Gürültü Azaltma:*5x5 çekirdek ile erozyon ve genişletme işlemleri uygulanarak gürültü azaltılmıştır.
* *Kontur Algılama:*En büyük kontur (minimum 500 piksel) seçilerek topun konumu ve sınırlayıcı kutusu ([x, y, w, h]) hesaplanmıştır.
* *Ekran Bölünmesi ve Hareket Kontrolü*: Görüntü üçe bölünerek (sağ, sol, orta) topun merkez noktasına göre hareket belirlenmiş; kontur alanına göre yakınlık tespit edilerek araç durdurulmuştur. Komutlar, 9600 baud hızında seri iletişimle Arduino Mega’ya gönderilmiştir.

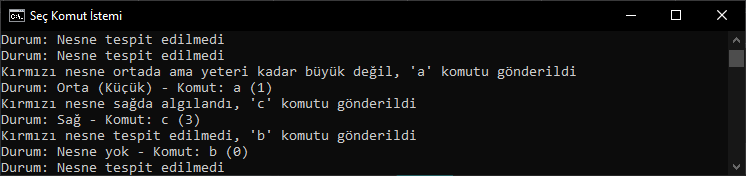
**2.4. Performans Metrikleri**

Sistemin performansı, doğruluk, kesinlik, duyarlılık ve F1-skoru metrikleriyle değerlendirilmiştir. Doğruluk, topun konum ve mesafesinin doğru tespitini ölçer; kesinlik ve duyarlılık, sınıflandırma başarısını değerlendirir; F1-skoru ise bunların harmonik ortalamasını sunar. Metrikler, Denklem 1–Denklem 3’te tanımlanmıştır.

**2.5. Deneysel Çıktılar**

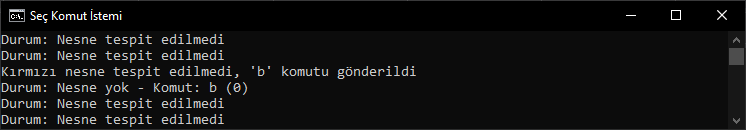
Sistemin çalışması sırasında Orange Pi 3 LTS terminalinde elde edilen loglar, kırmızı nesnenin tespit edilip edilmediği ve buna bağlı olarak gönderilen motor kontrol komutlarını yansıtır. Aşağıdaki görseller, deneysel süreçlerin gerçek zamanlı çıktılarını göstermektedir:

* *Şekil 2.13:*Terminalde "Nesne tespit edilmedi" mesajı ve "b" komutu gönderildiği durumun logu. Kırmızı nesne algılanmadığında sistemin varsayılan olarak dur komutunu ilettiği görülmektedir."Nesne yok - Komut: b (0)" mesajı ve tekrarlanan "Nesne tespit edilmedi" logları. Sistem, nesne bulunmadığında her 3 saniyede bir dur komutunu tekrarlar."Orta (Küçük) - Komut: a (1)" mesajı ve "a" komutunun gönderildiği durum. Kırmızı nesne orta bölgede tespit edilmiş, ancak kontur alanı büyük eşik değerini aşmadığı için ileri hareket komutu iletilmiştir. "Sağ - Komut: c (3)" mesajı ve "c" komutunun gönderildiği durum. Kırmızı nesne sağ bölgede algılanmış ve sağa dönme komutu iletilmiştir.

****

**Şekil 2.13:** Arduino Mega’nın motor kontrol kodu.

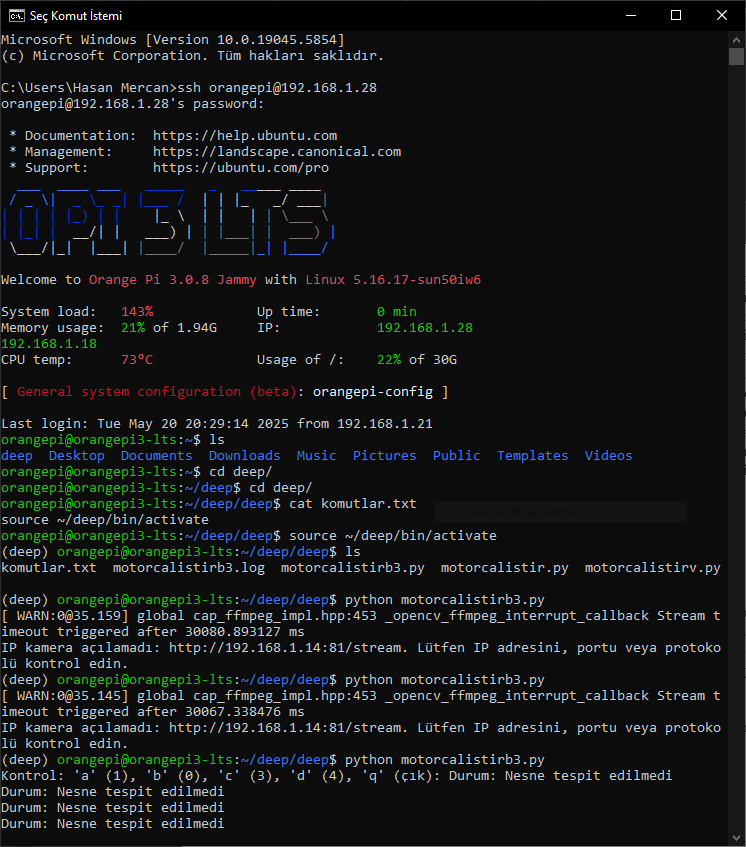
* *Şekil 2.14:*Terminalde "Nesne yok - Komut: b (0)" mesajı ve bağlantı kesintisi (Connection reset) uyarısı. Sistem, bağlantı sorunlarıyla karşılaştığında dur komutunu korumuştur ayrıca sistem hareket kontrolünü dengelemek için her 3 saniyede bir b yani dur komutu göndermektedir.

****

**Şekil 2.14:** Arduino Mega’nın motor kontrol kodu.

* *Şekil 2.15:*Terminalde Orange Pi 3 LTS’nin başlangıç ekranı ve "IP kamera açılamadı" hatası ile birlikte "Nesne tespit edilmedi" logları. IP adresi veya protokol hatası nedeniyle görüntü akışının kesildiği bir durum gösterilmektedir.

Bu loglar, sistemin kırmızı nesneyi algılama, konum belirleme ve motor kontrol komutlarını iletme süreçlerini gerçek zamanlı olarak belgelemektedir.

****

**Şekil 2.1:** Arduino Mega’nın motor kontrol kodu.

**3. Deneysel Kurulumlar**

Deneyler, kontrollü bir iç mekan test alanında gerçekleştirilmiştir. ESP32-CAM ile görüntü toplama, Orange Pi 3 LTS ile görüntü işleme ve Arduino Mega ile motor kontrolü yapılmıştır. Test senaryoları, kırmızı topun farklı konumlarda (sağ, sol, orta) ve mesafelerde (uzak, yakın) takibini içerir. Araç, topun konumuna göre hareket etmiş ve yakınlıkta durmuştur. Deneyler, Python ve OpenCV kullanılarak gerçekleştirilmiş; CNN modeli denenmiş, ancak donanım kısıtlamaları nedeniyle HSV algoritması tercih edilmiştir. Tablo 1, sistemin hiperparametrelerini gösterir.

**4. Tartışma ve Sonuçlar**

Bu çalışma, düşük maliyetli gömülü sistemlerde kırmızı bir topu takip eden otonom bir araç sistemi geliştirmiştir. CNN modeli, Orange Pi 3 LTS’nin sınırlı işlem kapasitesi nedeniyle gerçek zamanlı çalıştırılamamış; bunun yerine HSV tabanlı algoritma kullanılmıştır. Testler, sistemin kontrollü ortamlarda topun konumunu ve mesafesini doğru tespit ettiğini, ancak dinamik aydınlatma ve hızlı hareket senaryolarında performansın düştüğünü göstermiştir. Gelecekte, NVIDIA Jetson gibi güçlü donanımlar, veri artırma ve MobileNet gibi hafif CNN modelleriyle performans artırılabilir. Bu çalışma, otonom araçlarda görüntü işleme tekniklerine teorik ve pratik katkı sunar.

**Kaynaklar**

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.  
[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.  
[3] Abadi, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467.  
[4] Liu, W., et al. (2016). SSD: Single shot multibox detector. European Conference on Computer Vision, 21-37.