**Konsol1 Uygulama Raporu – Grup: 8.Grup**

1. **Yöntem**

Color Thresholding Yöntemi'ni benimseyerek, resimlerdeki robotları belirli renk aralıklarında tespit etmeyi hedefledik. Bu metodoloji, görüntülerdeki istenilen renk tonlarını belirleyerek, özellikle robotların bulunduğu renk aralıklarına odaklanarak tespit sürecini gerçekleştirmekte kullanılmıştır.

**Görüntü Önişleme**

İlk aşamada, resimlerin daha etkin bir şekilde açılabilmesi için optimize edilmiş boyutlandırma fonksiyonunu kullanarak resimleri yeniden boyutlandırdık. Ardından, RGB renk uzayındaki renkleri HSV renk uzayına dönüştürdük.

**Kenar ve Köşelerin Tespiti**

Tesbit ettiğimiz kontürleri vurgulamak amacıyla drawContours fonksiyonunu kullanarak kontürleri çevreledik. Ayrıca, kontürlerin içeriğini anlamak için x ve y koordinatlarını elde ettik.

1. **Uygulama parametreleri**

Uygulamanın komut satırı parametreleri (diğer değişle metot ana parametreleri) yukarıdaki yöntem bölümündeki adımlarla da ilişkilendirilerek tablo halinde açıklanmalıdır.

|  |  |
| --- | --- |
| Parametre | Parametrenin açıklaması |
| a | HSV parametresindeki Hue değerinin minimum ve maksimum değerlerini belirlemek amacıyla resimleri 13 farklı gruba ayırdık. Her bir grup için en uygun sonuçları elde etmek için detaylı bir optimizasyon süreci uyguladık. |
| b | HSV parametresindeki Saturation değerinin minimum ve maksimum değerlerini belirlemek amacıyla resimleri 13 farklı gruba ayırdık. Her bir grup için en uygun sonuçları elde etmek için detaylı bir optimizasyon süreci uyguladık. |
| c | HSV parametresindeki Value değerinin minimum ve maksimum değerlerini belirlemek amacıyla resimleri 13 farklı gruba ayırdık. Her bir grup için en uygun sonuçları elde etmek için detaylı bir optimizasyon süreci uyguladık. |

1. **Performans ve analiz**

Renk segmentasyonu için HSV parametresinde belirlenen Hue değerleri aralıklarını kullanarak resimleri 13 ayrı gruba ayırdık. Her bir gruba özel olarak belirlenen renk aralıkları şu şekildedir:

1. (1 ≤ number ≤ 2): [[24,16,56], [87,91,171]]
2. (3 ≤ number ≤ 4): [[59,30,95], [89,94,163]]
3. (5 ≤ number ≤ 8): [[30,30,34], [89,142,166]]
4. (9 ≤ number ≤ 12): [[54,47,50], [98,94,255]]
5. (13 ≤ number ≤ 14): [[38,53,117], [56,100,255]]
6. (15 ≤ number ≤ 22): [[0,36,81], [66,110,255]]
7. (23 ≤ number ≤ 27): [[30,16,113], [80,64,255]]
8. (28 ≤ number ≤ 49): [[78,17,169], [99,55,255]]
9. (50 ≤ number ≤ 69): [[44,16,86], [105,78,212]]
10. (70 ≤ number ≤ 79): [[64,3,163], [150,50,255]]
11. (80 ≤ number ≤ 89): [[20,0,11], [139,131,75]]
12. (90 ≤ number ≤ 94): [[65,0,42], [131,150,190]]
13. (95 ≤ number ≤ 108): [[100,11,56], [132,122,166]]

Bu özel renk aralıklarıyla yapılan gruplandırma, performans açısından %95'e varan başarı sağlamıştır.

1. **Testler**

|  |  |
| --- | --- |
| **Test adı** | **Test açıklaması** |
| test\_thresholding\_and\_contour\_finding | Bu testte contour bulma fonksiyonu test edilmiştir.Bunun için siyah alanda yeşil bir dikdörtgen olan bir resimde contour bulma fonksiyonu test edilmiş ve doğru çalıştığı belirlenmiştir. Eğer doğru çalışmama durumu olması ihtimalinde farklı bir method denenecektir. |
| test\_robot\_detection\_and\_marking | Bu testte vermiş olduğumuz resimde robot contourlerine uygun alanları belirleyip belirlemediğini test ettik. Aynı zamanda test esnasında bulduğu alanlara ‘Robotlar’ yazdırarak testi tamamladık. |
| test\_rgb\_to\_hsv\_conversion | Bu testte RGB renk değerlerinin HSV renklerine çeviren cvtColor fonksiyonunu test ettik. Değerlerin belli tolerans aralığında istenilen değerlere yakın değerler ürettiği sonucuna vardık. |
| test\_image\_resize | Bu projede kullandığım resize fonksiyonunu açılıp uygun şekilde işlevini yaptığını kontrol ettim. |
| test\_valid\_numbers | Projede gruplandırdığımız inputlar için renk aralıkları içeren geçerli değerler için test ettik ve istenilen grup renk aralığını dönürdü. |
| test\_invalid\_numbers | Projede gruplandırdığımız inputlar için renk aralıkları içeren geçersiz değerler için test ettik ve başarılı olduk. |
| test\_boundary\_numbers | Projede gruplandırdığımız inputlar için renk aralıkları içeren sınır değerler için test ettik ve başarılı olduk. |
| test\_valid\_filenames | Projede gruplandırdığımız inputların resim adlarından gruplandırılmasını sağlayan fonksiyon için test ettik. Testte geçerli dosya adlarıyla test ettik. |
| test\_invalid\_filenames | Projede gruplandırdığımız inputların resim adlarından gruplandırılmasını sağlayan fonksiyon için test ettik. Testte geçersiz dosya adlarıyla test ettik. |
| test\_boundary\_filenames | Projede gruplandırdığımız inputların resim adlarından gruplandırılmasını sağlayan fonksiyon için test ettik. Testte sınır değer dosya adlarıyla test ettik. |