

23 NİSAN 2024



SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ GÖRÜNTÜ İŞLEME VİZE ÖDEVİ

HAFİZE CEMİLE ÖĞÜT
B210109058

GÖRÜNTÜ İŞLEME VİZE ÖDEVİ



Vize ödev kapsamında yeşil zemin üzerinde farklı renkteki topların içeriğinde kırmızı top tespiti yapılmak istenmiştir. Top tespit edildikten sonra hızı ve topların durma anı istenmiştir.

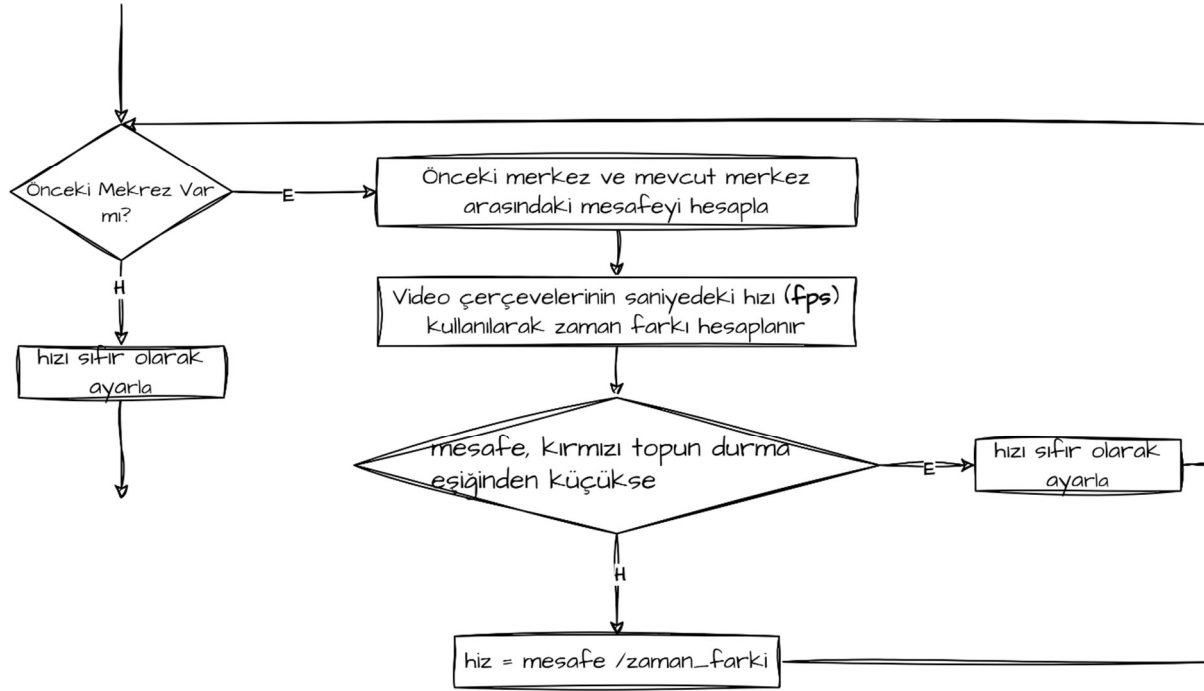
Proje kapsamında videolar veri seti dosyasını içinde bulunmaktadır. Video yolu değişkenine gönderilmek üzere videonun bulunduğu dosya yolu gönderilmektedir. Farklı görüntü işleme teknikleri kullanılarak deneyimlerim sonucunda algoritmamda kullandığım görüntü işleme teknikleri aşağıda verilmiştir.

1. **HSV Renk Uzayı Dönüşümü:** Görüntüler HSV (Ton, Doygunluk, Parlaklık) renk uzayına dönüştürülür. HSV, renkleri temsil etmek için daha uygun olan bir renk uzayıdır.
2. **Renk Maskeleme:** Belirli renk aralıklarını tanımlayarak, görüntüdeki istenmeyen renkleri filtrelemek için renk maskeleme işlemi yapılır. Bu kodda, kırmızı ve beyaz renkleri için ayrı ayrı maskeler oluşturulur.
3. **Maske Yumuşatma:** Elde edilen renk maskeleri yumuşatılır. Bu, maskenin içindeki küçük gürültüleri azaltmaya ve daha homojen bir görünüm elde etmeye yardımcı olur. Median bulanıklaştırma (median blur) kullanılarak yumuşatma sağlanır.
4. **Eşikleme:** Görüntülerdeki belirli bir eşik değeri kullanılarak piksellerin siyah veya beyaz olarak sınıflandırılmasına dayalı olarak ikili bir görüntü elde edilir. Bu kodda, kırmızı için bir eşikleme işlemi uygulanır.
5. **Morfolojik İşlemler:** Morfolojik işlemler, görüntüler üzerinde şekil tabanlı operasyonlar uygular. Bu işlemler genellikle gürültüyü azaltma, nesne tespiti ve nesneler arasındaki boşluğu doldurma gibi amaçlarla kullanılır. Bu kodda, morfolojik açma işlemi ile gürültü azaltma ve nesnelerin daha iyi tanınması sağlanır.
6. **Kontur Bulma:** Maske üzerindeki nesnelerin konturları bulunur. Bu konturlar, nesnelerin sınırlarını belirlemek için kullanılır. Bu kodda, kırmızı ve beyaz topların konturları bulunur.

top_hizi_hesapla(self, merkez_kirmizi):

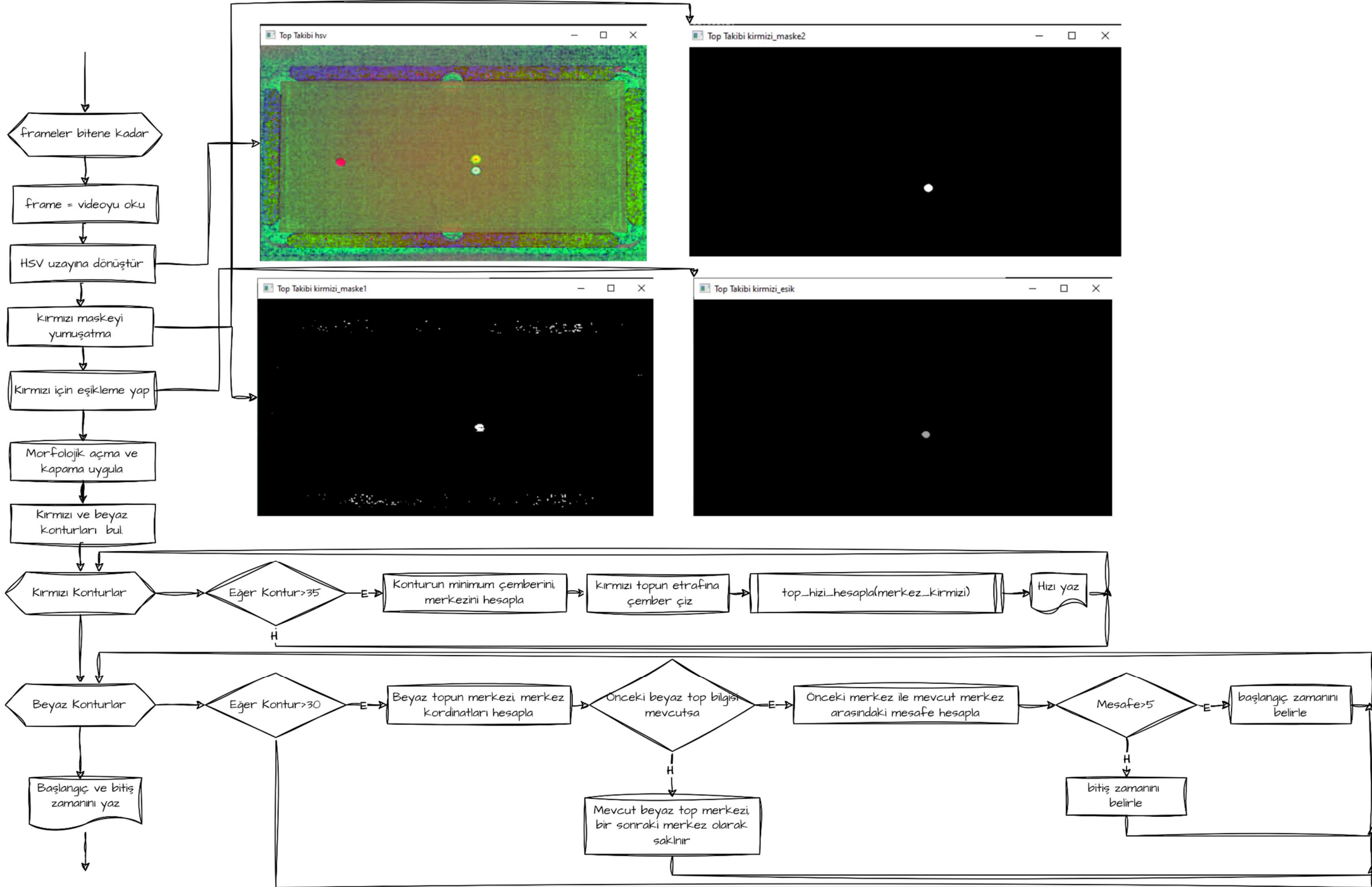
1. Önceki merkez bilgisi mevcut ise:
 - 1.1. Önceki merkez ve mevcut merkez arasındaki mesafe hesaplanır.
 - 1.2. Video çerçevelerinin saniyedeki hızı (**fps**) kullanılarak zaman farkı hesaplanır.
 - 1.3. Eğer mesafe, kırmızı topun durma eşiğinden (**kirmizi_durma_esigi**) küçükse, hız sıfır olarak ayarlanır.
 - 1.4. Aksi halde, hız mesafe ile zaman farkının bölünmesiyle hesaplanır.
 - 1.5. Hesaplanan hız değeri döndürülür.
2. Önceki merkez bilgisi mevcut değilse, hız olarak sıfır döndürülür.

top_hizi_hesapla(self, merkez_kirmizi):

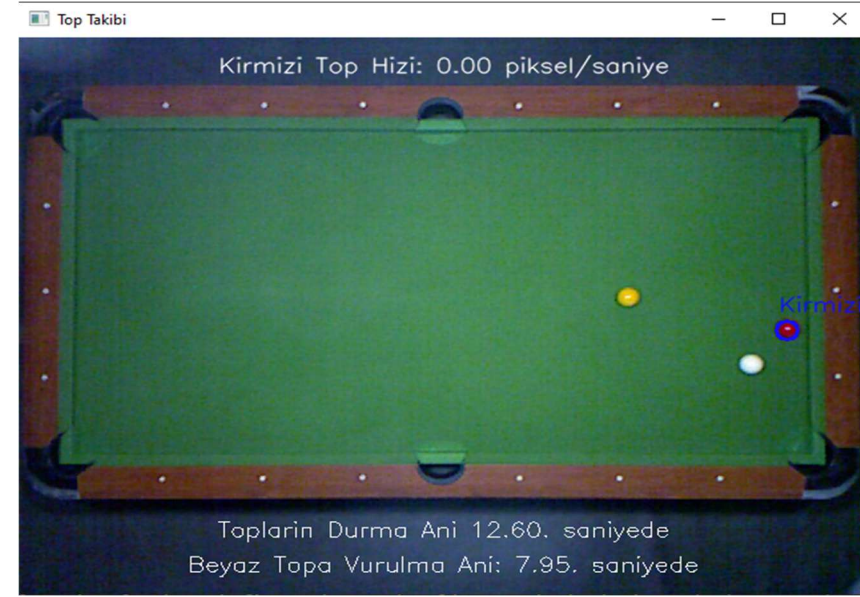


run(self):

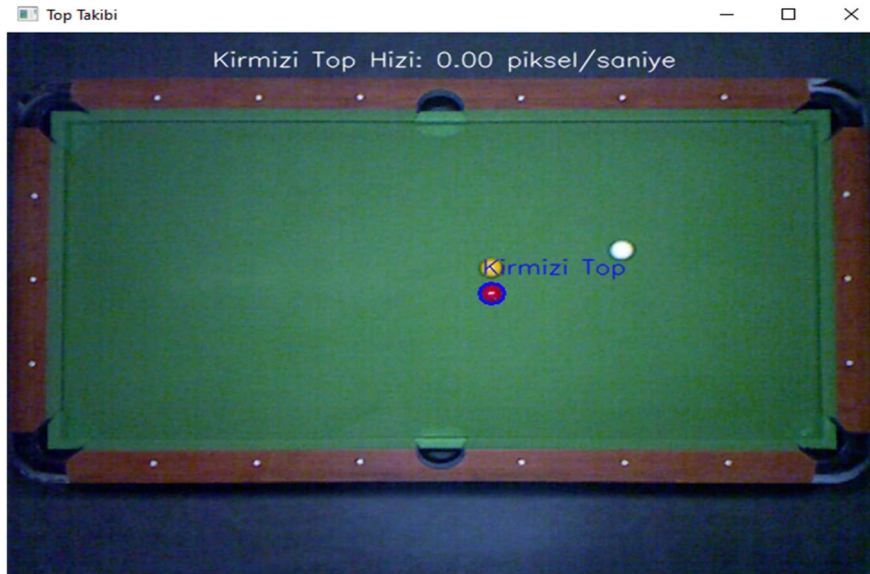
1. Sonsuz bir döngü oluşturarak video çerçevelerini okur.
2. Her çerçeve için HSV renk uzayına dönüştürme işlemi yapılır.
3. Kırmızı ve beyaz renkler için maske oluşturulur.
4. Kırmızı ve beyaz için morfolojik açma işlemi yapılır.
5. Kırmızı ve beyaz konturlar bulunur.
6. Kırmızı topun konturları izlenir ve hızı hesaplanır.
7. Hız bilgisi çerçeveye yazdırılır.
8. Beyaz topun vuruş anı ve durma anı belirlenir.
9. Vuruş ve durma anları ekrana yazdırılır.
10. Çerçeve gösterilir.
11. Çıkış tuşu (q) ile döngü sonlandırılır.
12. Sonuçlar yazdırılır.
13. Video kaynağı serbest bırakılır ve pencereler kapatılır.



Ekran Çıktıları:



Şekil1- Video1 için kırmızı top tespiti ve hız hesaplaması yapılmıştır.



Şekil 2- Video 2 için kırmızı top tespiti ve hız hesaplaması yapılmıştır.