

Görüntü Segmentasyonu

Watershed algoritması: Bir görüntüyü farklı bölgelerine ayırmak için kullanılan morfolojik bir yaklaşımdır. Görüntüyü coğrafi bir manzara gibi ele alır ve her bir nesneyi ayrı bir watershed olarak tanımlar. Algoritma, görüntüyü bir topografik haritaya benzer bir yapıya dönüştürür. Görüntüdeki her bir piksel, bir yüksekliğe karşılık gelir. Daha sonra, en düşük noktalardan başlayarak suyun akış yönü belirlenir. Suyun birleştiği noktalar, havza sınırlarını tanımlar. Bu, özellikle nesnelerin birbirine çok yakın olduğu veya gürültülü görüntülerde nesne segmentasyonu için oldukça etkili bir yöntemdir. Eğer segmentlemek istenen nesneler birbirine yakın ve keskin sınırlarla ayrılmışsa Watershed iyi bir tercih olabilir. Görüntüde kenarların belirgin olduğu ve morfolojik işlemlerin kullanılabildiği durumlarda daha etkilidir. Bu model aşağıdaki şekilde uygulanır:

- Görüntü ve Kenar Haritası Oluşturma: Öncelikle, görüntü gri tonlamaya çevrilir ve kenar haritası oluşturulur.
- Marker Belirleme: Her bir nesne ve arka plan için markerlar (işaretleyiciler) tanımlanır.
- Watershed Uygulama: “cv2.watershed()” fonksiyonu kullanılarak segmentasyon gerçekleştirilir.

GrabCut Algoritması: GrabCut, interaktif bir görüntü segmentasyon algoritmasıdır. Kullanıcı, segmentasyon yapılacak nesnenin kaba bir sınırını belirler ve algoritma bu sınırları kullanarak nesneyi arka plandan ayırır. Bu işlem, enerji minimizasyonu ve graf tabanlı yöntemlerle gerçekleştirilir. Kullanıcının segmentasyonu başlatmak için bir dikdörtgen veya mask belirleyebildiği durumlar için uygundur. Nesnenin arka plandan belirgin bir şekilde ayrıldığı durumlarda ve insan vb. canlıları görüntülerden ayırmak için daha etkilidir. Bu algoritma aşağıdaki şekilde uygulanır:

- Başlangıç Maskesi ve Dikdörtgen Tanımlama: Segmentasyon için başlangıç maskesi ve başlangıç dikdörtgen alanı belirlenir.
- GrabCut Uygulama: “cv2.grabCut()” fonksiyonu ile segmentasyon gerçekleştirilir.
- Sonuç Maskesi: Sonuç maskesi ile nesne ve arka plan ayrıştırılır.

K-means Kümeleme: Görüntüdeki farklı renk bölgelerini ayırmak için uygundur. Renk veya yoğunluk tabanlı basit segmentasyon gerektiren durumlarda kullanılabilir. Pikselleri belirli sayıda kümeye ayırarak segmentasyon yapar. Her küme, benzer renk veya yoğunluk özelliklerine sahip piksellerden oluşur. Bu yöntem, renk tabanlı segmentasyon için oldukça etkilidir. Bu model aşağıdaki şekilde uygulanır:

- Görüntü Verilerinin Hazırlanması: Görüntü verilerini bir vektör formatına dönüştürülür.
- K-means Uygulama: “cv2.kmeans()” fonksiyonu ile kümeleri oluşturulur.
- Sonuçları Görselleştirme: Her kümeyi farklı renklerle temsil edilir.

K-means Kümelemenin Giriş Parametreleri:

1. örnekler: np.float32 veri türünde olmalıdır ve her özelliğin tek bir sütunda yer alması gerekmektedir.
2. küme_sayısı (K): Sonuçta istenen küme sayısıdır.
3. kriterler: İterasyonun sonlandırma kriteridir. Bu kriter karşılandığında algoritma iterasyonu durur. Aslında bu, 3 parametreden oluşan bir demet olmalıdır. Bunlar (tip, max_iter, epsilon):
 - a. tip: Sonlandırma kriterinin türüdür. Aşağıdaki 3 başlık altında açıklanır:
 - i. “cv.TERM_CRITERIA_EPS”: Belirtilen doğruluk, epsilon, elde edilirse algoritmanın iterasyonunu durdurur.
 - ii. “cv.TERM_CRITERIA_MAX_ITER”: Belirtilen maksimum iterasyon sayısına ulaşıldığında algoritmayı durdurur.
 - iii. “cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER”: Yukarıdaki koşullardan herhangi biri karşılandığında iterasyonu durdurur.
 - b. max_iter: Maksimum iterasyon sayısını belirten bir tam sayıdır.
 - c. epsilon: Gereken doğruluk seviyesidir.
4. attempts: Algoritmanın farklı başlangıç etiketlemeleri kullanılarak kaç kez çalıştırılacağını belirten bir bayraktır. Algoritma, en iyi sıklığı sağlayan etiketleri döndürür. Bu sıklık bir çıktı olarak döndürülür.
5. flags: Başlangıç merkezlerinin nasıl alındığını belirlemek için kullanılan bayraktır. Genellikle iki bayrak kullanılır: “cv.KMEANS_PP_CENTERS” ve “cv.KMEANS_RANDOM_CENTERS”.

K-means Kümelemenin Çıkış Parametreleri:

1. compactness: Her noktanın kendi merkezine olan kareler toplamı uzaklığıdır.
2. labels: Her bir elemanın '0', '1' şeklinde işaretlendiği etiket dizisidir.
3. centers: Kümelerin merkezlerinin dizisidir.