

Nesne Algılama

Şablon Eşleştirme: OpenCV kütüphanesini kullanarak şablon eşleştirme işlemi, bir görüntü içinde belirli bir şablonu veya alt görüntüyü bulmak için kullanılır. Bu, özellikle belirli bir nesneyi tespit etmek veya yerini belirlemek için yaygın olarak kullanılır. Şablon Eşleştirme yöntemi dışındaki diğer yöntemler sınıflandırıcı olarak tanımlanan Machine Learning algoritmalarıdır. Bu algoritmalar da öğretilmiş bir nesneyi tanımak için kullanılır. Şablon eşleştirme işleminde kullanılan fonksiyonlar:

- “cv2.matchTemplate” fonksiyonu kullanılarak, ana görüntüdeki her konum için şablonun eşleşme puanı hesaplanır.
- “cv2.minMaxLoc” fonksiyonu ile elde edilen sonuçlar analiz edilebilir.
- Eşleşmenin uygulanıp, bulunması sonucunda elde edilen eşleşmenin konumu ve boyutu, ana görüntü üzerinde bir dikdörtgen çizerek görselleştirilebilir. Bu, “cv2.rectangle()” fonksiyonu kullanılarak yapılır.

OpenCV kütüphanesinde yüz, göz, profil gibi belirli nesneleri hızlı ve etkili bir şekilde algılamak için kullanılan bir makine öğrenmesi algoritmasıdır.

Şablon eşleştirme yönteminde kaynak ile şablonu eşleştiren kullanılan farklı yöntemler bulunur. Bu yöntemler şunlardır:

- “cv2.TM_CCORR”: Korelasyon katsayısıdır.
- “cv2.TM_CCORR_NORMED”: Normalleştirilmiş korelasyon katsayısıdır.
- “cv2.TM_SQDIFF”: Korelasyondur.
- “cv2.TM_SQDIFF_NORMED”: Normalleştirilmiş korelasyondur.
- “cv2.TM_CCOEFF”: Korelasyon katsayısıdır.
- “cv2.TM_CCOEFF_NORMED”: Normalleştirilmiş korelasyon katsayısıdır.
- “cv2.TM_CCORR”: Korelasyondur.
- “cv2.TM_CCORR_NORMED”: Normalleştirilmiş korelasyondur.
- “cv2.TM_SQDIFF”: Kare farkları toplamıdır.
- “cv2.TM_SQDIFF_NORMED”: Normalleştirilmiş kare farkları toplamıdır.

Haar Cascades ile Yüz Algılama: OpenCV kütüphanesinde yüz, göz, profil gibi belirli nesneleri hızlı ve etkili bir şekilde algılamak için kullanılan bir makine öğrenmesi algoritmasıdır. Bu yöntem, önceden eğitilmiş sınıflandırıcılar sayesinde, bir görüntüdeki nesnelerin özelliklerini belirleyerek nesneyi tanımlamaya çalışır. Bunlar arasında “haarcascade_frontalface_default.xml” gibi yüz tespiti için kullanılan modeller de vardır.

Bu yüz algılama modelinde kullanılan fonksiyonlardan biri olan “detectMultiScale” fonksiyonu ile de görüntüdeki yüzler algılanır ve bu fonksiyon şu parametreleri alır:

- ‘scaleFactor’: Algoritmanın her görüntü ölçeğinde boyutunu ne kadar azaltacağını belirler.
- ‘minNeighbors’: Algılanan nesnelerin birleştirilmesi için gereken minimum komşu dikdörtgen sayısını ifade eder.
- ‘minSize’: Algılanacak nesnelerin minimum boyutunu belirtir.

Haar özellikleri, belirli bir görüntüdeki aydınlık ve karanlık bölgelerin farklılıklarını tespit eden basit dikdörtgensel desenlerdir. Bu özellikler, insan yüzü gibi nesneleri tanımak için kullanılır. Aşağıda yaygın kullanılan birkaç Haar özelliğine örnekler verilmiştir:

- Kenar Özellikleri: Bir dikdörtgenin bir tarafı diğerine göre daha aydınlık veya karanlık olduğunda oluşur.
- Çizgi Özellikleri: Bir dikdörtgenin ortası kenarlarına göre daha aydınlık veya karanlık olduğunda oluşur.
- Dörtgen Özellikleri: Karşıtlık gösteren dört bölgeye ayrılan dikdörtgenlerdir.

Sonuç olarak Haar Cascades ile yüz algılama, aşağıdaki adımlarla gerçekleştirilir:

1. Görüntünün Hazırlanması: Giriş görüntüsü gri tonlamalı bir formata dönüştürülür. Bu işlem, hesaplama yükünü azaltır ve daha doğru sonuçlar sağlar.
2. Özelliklerin Hesaplanması: Bütünlük görüntüsü kullanılarak Haar özellikleri hızlıca hesaplanır.
3. Cascade Sınıflandırıcı ile Tarama: Sınıflandırıcı, çeşitli ölçeklerde ve konumlarda görüntü üzerinde kaydırılarak yüzlerin varlığı kontrol edilir.
4. Sonuçların Filtrelenmesi: Tespit edilen yüzler, doğru sonuçlar elde etmek için ek post-processing adımları ile filtrelenir.
5. Algılanan Yüzlerin Çizilmesi ve Sonuçları Gösterme: Filtrelenen yüzlerin etrafına dikdörtgen çizilerek sonuçlar görselleştirilip, cv2.imshow komutuyla yüzler gösterilir.

HOG ile İnsan Algılama: İnsanları algılamak için kullanılan bir özellik tanımlayıcısıdır. Bir görüntünün yerel şekil ve doku bilgilerini temsil etmek için kullanılan güçlü bir özneliktir. Özellikle insan algılamada oldukça başarılı sonuçlar verir. HOG, nesnelerin genel şeklini, kenarlarını ve konturlarını yakalayıp, farklı pozlarda, aydınlatma koşullarında ve kısmi tıkanmalarda bile insanları doğru bir şekilde tespit edebilir.

HOG algoritmasının tanımlayıcı tekniği, bir görüntü algılama penceresinin veya ilgilenilen bölgenin lokalize bölümlerinde gradyan oryantasyonunun oluşumlarını sayar. HOG modelini kullanmak için “`cv2.HOGDescriptor()`” sınıfı kullanılır.

HOG, bir görüntünün yerel nesne görünümünü ve şeklini, gradyan yöntemlerini kullanarak temsil eder. HOG'nin nasıl çalıştığına dair adımlar aşağıda listelenmiştir:

1- Gradyan Hesaplama:

- Her piksel için gradyan yönü ve büyüklüğü hesaplanır. Bu genellikle Sobel filtresi gibi basit bir diferansiyel operatör kullanılarak yapılır.

2- Hücrelere Bölme:

- Görüntü, genellikle 8x8 piksel boyutlarında küçük hücrelere bölünür.
- Her hücredeki piksel için hesaplanan gradyan yönleri histogramda toplanır.

3- Yönlendirilmiş Gradyan Histogramı:

- Her hücre için bir yönlendirilmiş gradyan histogramı oluşturulur. Bu histogram, belirli sayıda yönlendirme kutusuna ayrılır. Her kutu, hücredeki gradyanların belirli bir yönde ne kadar yoğun olduğunu temsil eder.

4- Blok Normalizasyonu:

- Komşu hücre grupları üzerinde histogramlar birleştirilir ve normalize edilir. Bu adım, aydınlatma değişikliklerine karşı dayanıklılığı artırır ve genellikle ‘2x2’ hücreden oluşan bloklar kullanılır.

5- Özellik Vektörü Oluşturma:

- Görüntünün tüm blokları için oluşturulan ve normalize edilen histogramlar birleştirilerek bir özellik vektörü elde edilir.
- Bu vektör, görüntüdeki nesneleri tanımlamak için kullanılabilecek bir modelle beslenir.

HOG özellikleri, insanları algılamak için yaygın olarak bir SVM ile birleştirilir. İnsan Algılama ile HOG Kullanımı için OpenCV'de bu model önceden eğitilmiştir ve bu model, “cv2.HOGDescriptor_getDefaultPeopleDetector()” fonksiyonu ile kullanılabilir.

HOG Tanımlayıcı ve SVM yüklemek için, “cv2.HOGDescriptor()” ile bir HOG tanımlayıcısı oluşturulur. “setSVMDetector()” fonksiyonu, önceden eğitilmiş bir Lineer SVM modeli yükler. Bu modelde, insanların HOG özellikleriyle tanımlanmasına odaklanılmıştır.

Bir görselde çoklu ölçeklerde insanları algılamak için “detectMultiScale()” fonksiyonu kullanılır. Bu fonksiyon şu parametrelerle kontrol edilir:

- ‘winStride’: Algılama penceresinin her adımda ne kadar kaydırılacağını belirler.
- ‘padding’: Algılama sırasında pencerelerin etrafına eklenen boşluktur.
- ‘scale’: Algılama ölçeğini ayarlayarak farklı boyutlardaki insanları bulmayı sağlar.

Görseldeki insanlar algılandıktan sonra, algılanan her nesne(insan) için “cv2.rectangle()” fonksiyonu kullanılarak görüntü üzerinde yeşil bir dikdörtgen çizilerek algoritma tamamlanır.