**1. Kullanılan Kütüphanelerin Dahil Edilmesi:**

ekran görüntüsü, metin, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu kısımda, görüntü işleme için OpenCV (cv2), komut satırı argümanlarını işlemek için argparse ve diziler ve matrisler üzerinde işlem yapmak için numpy kütüphaneleri dahil edilir.

**2. Komut Satırı Argümanlarının İşlenmesi:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu kısımda argparse kullanılarak komut satırından dört adet argüman alınır: görüntü dosyasının yolu (--image), YOLOv3 ağına ait yapılandırma dosyasının yolu (--config), ağırlıklar dosyasının yolu (--weights) ve sınıf etiketlerinin bulunduğu dosyanın yolu (--classes).

**3. YOLO Modelinden Çıktı Katmanlarının İsimlerinin Alınması:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu işlev, YOLOv3 ağının çıkış katmanlarının isimlerini almak için kullanılır.

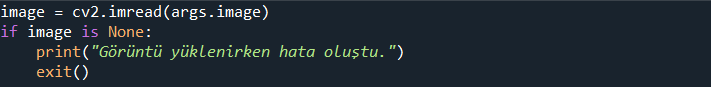
**4. Algılanan Nesneleri Çizmek İçin İşlev:**

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu işlev, algılanan nesnelerin çizilmesi için kullanılır. Etiketin, algılanan sınıfın adını dikdörtgenin sol üst köşesine ve dikdörtgenin dışına yazılmasını sağlar.

**5. Görüntünün Yüklenmesi:**



Belirtilen görüntü dosyası yüklenir. Eğer görüntü yüklenemezse, bir hata mesajı yazdırılır ve program sonlandırılır.

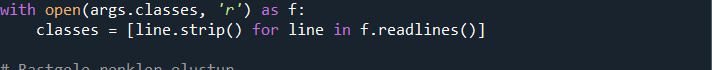
**6. Görüntü Özelliklerinin Alınması:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu satırlarda, görüntünün genişliği (Width) ve yüksekliği (Height) alınır. Ayrıca, görüntünün boyutlarını ölçeklemek için kullanılacak bir ölçek faktörü (scale) belirlenir.

**7. Sınıf İsimlerinin Yüklenmesi:**



Bu kısımda, sınıf isimleri dosyadan (args. Classes) okunur ve bir liste olarak classes değişkenine atanır.

**8. Rastgele Renklerin Oluşturulması:**



Her bir sınıf için rastgele bir renk oluşturulur ve bu renkler COLORS dizisinde saklanır.

**9. YOLOv3 Ağının Yüklenmesi:**



Bu satırda, YOLOv3 modeli cv2.dnn.readNet() fonksiyonu kullanılarak ağırlıklar ve yapılandırma dosyalarından (args.weights ve args.config) yüklenir.

**10. Görüntüden Blob Oluşturulması:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görüntü, YOLOv3 modeline uygun bir giriş blob'una dönüştürülür. Bu işlem, ağırlıkların işlenmesi için görüntünün ön işleme adımını oluşturur.

**11. Ağın Çıkışlarının Alınması:**



Ağa giriş olarak blob verilir ve çıkışlar alınır.

**12. Nesne Tespitinin Gerçekleştirilmesi:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Algılanan nesnelerin koordinatları, güven skorları ve sınıf kimlikleri class\_ids, confidences ve boxes listelerine eklenir.

**13. Non-Maximum Suppression (NMS) Uygulanması:**



Algoritmik olarak, NMS, aynı nesneyi birden fazla kez algılayan kutular arasından en uygun olanını seçer.

**14. Algılanan Nesnelerin Çizilmesi ve Sayılması:**

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Algılanan nesnelerin çizilmesi ve sayılması işlemleri gerçekleştirilir.

**15. Sonuçların Gösterilmesi ve Kaydedilmesi:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Elde edilen sonuçlar ekranda gösterilir ve görüntü kaydedilir. Bu kısımlar, algılama işlemi tamamlandıktan sonra sonuçları göstermek ve işlemi sonlandırmak için kullanılır.