**Kod Analizi Raporu**

**Giriş**

Bu kod, YOLOv8 modelini kullanarak video akışındaki araçları tanımlamak ve saymak amacıyla tasarlanmıştır. Araç türlerini algılar ve belirli bir hiza çizgisini geçen araçları sayar.

import cv2

from ultralytics import YOLO

import pandas as pd

# YOLOv8 modelini yükle

model = YOLO('yolov8s.pt')

# Araç sınıflarını tanımla

vehicle\_classes = ['car', 'bus', 'truck', 'motorcycle']

# Sayım çizgisinin Y koordinatı (hiza)

counting\_line\_y = 460

line= 490

# Araç sayımı

vehicle\_count = {

    'car': 0,

    'bus': 0,

    'truck': 0,

    'motorcycle': 0

}

# Araçların izini tutmak için bir yapı

vehicle\_tracker = {}

# Video işleme fonksiyonu

def process\_video(video\_path):

    global vehicle\_count, vehicle\_tracker

    # VideoCapture nesnesi oluştur

    cap = cv2.VideoCapture(video\_path)

    # Video yazıcı (output) ayarları

    fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

    out = cv2.VideoWriter('output.avi', fourcc, 20.0, (int(cap.get(3)), int(cap.get(4))))

    vehicle\_id = 0  # Araç ID'si izlemek için

    while cap.isOpened():

        ret, frame = cap.read()

        if not ret:

            break

        # YOLO modelini kullanarak tahmin yap

        results = model(frame)

        # Sonuçları al

        results\_list = results[0].boxes.xyxy.cpu().numpy()

        classes = results[0].boxes.cls.cpu().numpy()

        confidences = results[0].boxes.conf.cpu().numpy()

        # Mevcut çerçevede görülen araçların ID'lerini takip etmek için geçici bir set

        current\_frame\_vehicles = set()

        # Tahmin edilen araçları gruplandır ve ekranda göster

        for i, (box, cls, conf) in enumerate(zip(results\_list, classes, confidences)):

            if model.names[int(cls)] in vehicle\_classes:

                vehicle\_type = model.names[int(cls)]

                xmin, ymin, xmax, ymax = map(int, box)

                confidence = conf

                # Yeni bir araçsa ID ata ve izleyiciye ekle

                if (xmin, ymin, xmax, ymax) not in vehicle\_tracker:

                    vehicle\_tracker[(xmin, ymin, xmax, ymax)] = {'id': vehicle\_id, 'counted': False}

                    vehicle\_id += 1

                vehicle\_info = vehicle\_tracker[(xmin, ymin, xmax, ymax)]

                current\_frame\_vehicles.add(vehicle\_info['id'])

                # Araç hiza çizgisini geçti mi kontrol et

                if ymin <= counting\_line\_y <= ymax and  vehicle\_info['counted']==False:

                    # Kutunun tam orta noktasını hesapla

                    y\_center = (ymin + ymax) // 2

                    # Eğer kutunun orta noktası, sayım çizgisine değdiyse

                    if y\_center >= counting\_line\_y and y\_center<=line  and y\_center< line:

                        vehicle\_count[vehicle\_type] += 1

                        vehicle\_info['counted'] = True

                # Sınıf adı ve güven ile sınır kutusunu çiz

                label = f"{vehicle\_type} {confidence:.2f}"

                cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)

                cv2.putText(frame, label, (xmin, ymin - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

        # Araçları izleme listesinden güncelle

        vehicle\_tracker = {box: info for box, info in vehicle\_tracker.items() if info['id'] in current\_frame\_vehicles}

        # Hiza çizgisini çiz

        #cv2.line(frame, (0, counting\_line\_y), (frame.shape[1], counting\_line\_y), (0, 0, 255), 2)

        #cv2.line(frame, (0, line), (frame.shape[1], counting\_line\_y), (0, 0, 255), 2)

        cv2.line(frame,(0,450),(1280,450),(0,0,255),3)

        cv2.line(frame,(0,470),(1280,470),(0,0,255),3)

        # Araç sayımını sağ üst köşeye yazdır

        text\_y = 30

        for vehicle\_type, count in vehicle\_count.items():

            text = f"{vehicle\_type}: {count}"

            cv2.putText(frame, text, (frame.shape[1] - 200, text\_y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (255, 255, 255), 2)

            text\_y += 30

        # Sonuçları video dosyasına yaz

        out.write(frame)

        # Sonuçları ekranda göster

        cv2.imshow('frame', frame)

        if cv2.waitKey(40) & 0xFF == ord('q'):

            break

    # Excel dosyasına kaydet

    save\_to\_excel(vehicle\_count)

    # Kaynakları serbest bırak

    cap.release()

    out.release()

    cv2.destroyAllWindows()

def save\_to\_excel(vehicle\_count):

    # DataFrame oluştur

    df = pd.DataFrame(list(vehicle\_count.items()), columns=['Vehicle Type', 'Count'])

    # Excel dosyasına yaz

    df.to\_excel('vehicle\_count.xlsx', index=False)

# Test işlemi

video\_path = 'cars.mp4'  # Buraya test edeceğiniz videonun yolunu ekleyin

process\_video(video\_path)

import cv2

from ultralytics import YOLO

import pandas as pd

**cv2**: OpenCV kütüphanesi, görüntü işleme ve video analizi için kullanılır.

**YOLO**: Ultralytics tarafından sağlanan YOLOv8 modeli, nesne tanıma için kullanılır.

**pandas**: Verileri işlemek ve Excel dosyasına kaydetmek için kullanılır.

**YOLOv8 Modelini Yükleme**

model = YOLO('yolov8s.pt')

YOLOv8 model dosyası 'yolov8s.pt' kullanılarak yüklenir.

**Araç Sınıflarını Tanımlama**

vehicle\_classes = ['car', 'bus', 'truck', 'motorcycle']

Algılanacak araç türleri belirlenir.

**Sayım Çizgisi**

counting\_line\_y = 460

line= 490

Sayım çizgisinin Y koordinatları belirlenir. Bu çizgiler araçların geçişini tespit etmek için kullanılır.

**Araç Sayımı ve İzleyici**

vehicle\_count = { 'car': 0,

'bus': 0,

'truck': 0,

'motorcycle': 0

}

vehicle\_tracker = {}

Araç türlerine göre sayım yapılacak sözlük oluşturulur.

Araçların izini tutmak için bir yapı tanımlanır.

**Video İşleme Fonksiyonu**

def process\_video(video\_path):

global vehicle\_count, vehicle\_tracker

Video dosyasını işlemek için bir fonksiyon tanımlanır ve global değişkenler kullanılır.

**Video Okuma ve Yazma**

cap = cv2.VideoCapture(video\_path)

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

out = cv2.VideoWriter'output.avi', fourcc, 20.0, (int(cap.get(3)), int(cap.get(4))))

VideoCapture nesnesi, video dosyasını okumak için kullanılır.

Video yazıcı (output) ayarları yapılır.

**Video Çerçeve İşleme Döngüsü**

while cap.isOpened():

ret, frame = cap.read()

if not ret:

break

results = model(frame)

results\_list = results[0].boxes.xyxy.cpu().numpy()

classes = results[0].boxes.cls.cpu().numpy()

confidences = results[0].boxes.conf.cpu().numpy()

current\_frame\_vehicles = set()

Video akışı çerçeve çerçeve okunur.

YOLO modeli her bir çerçeve için tahminler yapar ve sonuçları alır.

**Araç Tespiti ve Sayma**

for i, (box, cls, conf) in enumerate(zip(results\_list, classes, confidences)):

if model.names[int(cls)] in vehicle\_classes:

vehicle\_type = model.names[int(cls)]

xmin, ymin, xmax, ymax = map(int, box)

confidence = conf

if (xmin, ymin, xmax, ymax) not in vehicle\_tracker:

vehicle\_tracker[(xmin, ymin, xmax, ymax)] = {'id': vehicle\_id, 'counted': False}

vehicle\_id += 1

vehicle\_info = vehicle\_tracker[(xmin, ymin, xmax, ymax)]

current\_frame\_vehicles.add(vehicle\_info['id'])

if ymin <= counting\_line\_y <= ymax and vehicle\_info['counted']==False:

y\_center = (ymin + ymax) // 2

if y\_center >= counting\_line\_y and y\_center<=line and y\_center< line:

vehicle\_count[vehicle\_type] += 1

vehicle\_info['counted'] = True

label = f"{vehicle\_type} {confidence:.2f}"

cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)

cv2.putText(frame, label, (xmin, ymin - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

Tespit edilen araçların bilgileri ve konumları elde edilir.

Araçların geçiş çizgisini geçip geçmediği kontrol edilir ve sayım yapılır.

Araçların etrafına dikdörtgenler çizilir ve sınıf adı ile güven skoru eklenir.

Sayım Çizgisini Çizme ve Sonuçları Gösterme

cv2.line(frame,(0,450),(1280,450),(0,0,255),3)

cv2.line(frame,(0,470),(1280,470),(0,0,255),3)

text\_y = 30

for vehicle\_type, count in vehicle\_count.items():

text = f"{vehicle\_type}: {count}"

cv2.putText(frame, text, (frame.shape[1] - 200, text\_y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (255, 255, 255), 2)

text\_y += 30

out.write(frame)

cv2.imshow('frame', frame)

if cv2.waitKey(40) & 0xFF == ord('q'):

break

Sayım çizgisi çerçeveye çizilir.

Araç sayımları ekranda görüntülenir ve video dosyasına yazılır.

Kullanıcı 'q' tuşuna basarak döngüyü sonlandırabilir.

**Excel'e Kaydetme Fonksiyonu**

def save\_to\_excel(vehicle\_count):

df = pd.DataFrame(list(vehicle\_count.items()), columns=['Vehicle Type', 'Count'])

df.to\_excel('vehicle\_count.xlsx', index=False)

Araç sayımları bir pandas DataFrame'e dönüştürülür ve Excel dosyasına yazılır.

**Sonuç**

Bu kod, YOLOv8 modeli kullanarak bir video akışındaki araçları tanımlamak ve saymak için etkili bir yöntem sağlar. Araç türlerini tespit eder, sayım çizgisini geçen araçları sayar ve sonuçları bir Excel dosyasına kaydeder. Ayrıca, işlenmiş video dosyasını oluşturur ve ekranda gösterir.