

ARAÇ GÖRÜNTÜLERİNDEN PLAKA ALGILAMA

HASAN FİDAN

PROJENİN GENEL TANIMI

- Önceden eğitilmiş olan model Wpod-Net ile OpenCV, Görüntü İşleme teknikleri kullanarak 10 farklı ülkeden (Almanya, Vietnam, Japonya, Tayland, Suudi, Rusya, Kore, ABD, Hindistan, Çin) araç görüntülerinin Plakalarını algılamak ve çıkarmak.

Kullanılan Kütüphaneler, Araçlar

- Keras
- Tensorflow
- Numpy
- Matplotlib
- OpenCV

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from local_utils import detect_lp
from os.path import splitext, basename
from keras.models import model_from_json
import glob
```

- cv2: Görüntü işleme tekniklerini gerçekleştirmek için kullanacağımız OpenCV olarak da bilinen kütüphane.
- numpy: Çok boyutlu dizileri ve matris işlemlerini destekleyen bir kütüphane.
- matplotlib: Verilerimizi görselleştirmek ve çizimler yapmak için kullanılan bir kütüphane.
- local_utils: Wpod-Net'den gelen verileri işlemek için kullanılacak bazı fonksiyonları içerir.
- os.path / glob: Python için işletim sistemi arayüz paketi/kütüphanesi. Dizinlerle ve dosya sistemleriyle çalışmak için kullanacağız.
- keras.models: Model mimarisini JSON formatında yüklemek için bu kütüphaneden model_from_json paketini kullanacağız.

```
def load_model(path):  
    try:  
        path = splitext(path)[0]  
        with open('%s.json' % path, 'r') as json_file:  
            model_json = json_file.read()  
        model = model_from_json(model_json, custom_objects={})  
        model.load_weights('%s.h5' % path)  
        print("Loading model successfully...")  
        return model  
    except Exception as e:  
        print(e)  
  
wpod_net_path = "wpod-net.json"  
wpod_net = load_model(wpod_net_path)
```

Burada önceden eğitilmiş olan modelimizi yüklüyoruz

```
def preprocess_image(image_path,resize=False):  
    img = cv2.imread(image_path)  
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
    img = img / 255  
    if resize:  
        img = cv2.resize(img, (224,224))  
    return img
```

Burada preprocess_image adında, plaka görüntülerini okuması ve işlemesi için bir fonksiyon oluşturuyoruz. 2. satırda görüntüyü okuyor , 3.satırda RGB formatına dönüştürüyor,4.satırda veriyi 0-1 aralığında normalize ediyor

```
image_paths = glob.glob("Plate_examples/*.jpg")  
print("%i içerik bulundu..."%(len(image_paths)))
```

```
fig = plt.figure(figsize=(12,8))  
cols = 5  
rows = 4  
fig_list = []  
for i in range(cols*rows):  
    fig_list.append(fig.add_subplot(rows,cols,i+1))  
    title = splitext(basename(image_paths[i]))[0]  
    fig_list[-1].set_title(title)  
    img = preprocess_image(image_paths[i],True)  
    plt.axis(False)  
    plt.imshow(img)  
  
plt.tight_layout()  
plt.show()
```

Burada araçlardan oluşan veri setimizi görselleştiriyoruz. Bu veri seti, plakalarıyla birlikte 10 farklı ülkeden (Almanya, Vietnam, Japonya, Tayland, Suudi Arabistan, Rusya, Kore, ABD, Hindistan, Çin) elde edilen 20 araç görüntüsünü içeriyor. Üstte gördüğümüz kod bloğu 5 sütun ve 4 satırdan oluşan bir şekilde araçların fotoğraflarını göstermemizi sağlıyor



```
def get_plate(image_path, Dmax=608, Dmin=300):  
    vehicle = preprocess_image(image_path)  
    ratio = float(max(vehicle.shape[:2])) / min(vehicle.shape[:2])  
    side = int(ratio * Dmin)  
    bound_dim = min(side, Dmax)  
    _, LpImg, _, cor = detect_lp(wpod_net, vehicle, bound_dim, lp_threshold=0.5)  
    return LpImg, cor  
  
test_image = image_paths[1]  
LpImg, cor = get_plate(test_image)  
print("Plaka bulundu : ", splitext(basename(test_image))[0])  
  
plt.figure(figsize=(12,5))  
plt.subplot(1,2,1)  
plt.axis(False)  
plt.imshow(preprocess_image(test_image))  
plt.subplot(1,2,2)  
plt.axis(False)  
plt.imshow(LpImg[0])
```

Burada görüntüyü işleyen ,modelimize gönderen ve plaka görüntüsünü (LpImg) ve koordinatlarını (cor) döndüren get_plate adında bir işlev yazıyoruz. Plaka bulunamazsa, "No License plate is founded!" şeklinde bir hata dönüyor bu hatayı aldığımızda daha yüksek Dmin değeri vererek sorunu çözüyoruz. Bunun sebebi Wpod-Net' in görüntünün fazla bulanık olduğunda plakayı okumaması.




```

fig = plt.figure(figsize=(12,6))
cols = 5
rows = 4
fig_list = []

for i in range(cols*rows):
    fig_list.append(fig.add_subplot(rows,cols,i+1))
    title = splitext(basename(image_paths[i]))[0]
    fig_list[-1].set_title(title)
    LpImg,_ = get_plate(image_paths[i])
    plt.axis(False)
    plt.imshow(LpImg[0])

plt.tight_layout()
plt.show()

```

Son olarak üstteki kod ile elimizdeki bütün araçların görüntülerine `get_plate` işlemi uygulayarak hepsinin plakalarına erişebiliyoruz.

