

Yapay Görme Uygulama Projesi Raporu

Giriş

Bu rapor, Visea_-_Yapay_Gorme_Test_Projesi_v3 projesi üzerinde yapılan çalışmaları detaylı bir şekilde açıklamaktadır. Projenin amacı, segmentasyon modelleri kullanıldıktan sonra segmente edilmiş alanların sınıflandırmasının yapılması ve daha sonra nesnelerin sınıfları ile birlikte bounding box'larının input image üzerinde gösterilmesini sağlamaktır.

Proje Aşamaları

Proje akış şeması, aşağıdaki adımları içermektedir:

1: Veri seti üzerinde işlemler

YOLOv8 modelinde segmentasyon yapılabilmesi için veri seti üzerinde çeşitli düzenlemeler yapılmaktadır. 'RGB_to_GrayScale_AND_mask_to_YoloTxt.ipynb' kod dosyası kullanılarak ilk önce Ground Truth maskeleri, sadece belirlenmiş sınıflar olan insan ve arabalar için Resim[1]'de de görüldüğü üzere rgb değerleri kullanılarak seçilmekte ve Siyah-beyaz maske formu şeklinde kaydedilmektedir. Sadece insan ve arabaların siyah-beyaz maske forma dönüştürmesinin amacı ise Yolov8'in maske olarak txt dosyasındaki poligon bilgileri işlemesinden dolayı maskelerin txt formuna dönüştürülürken çevredekilerin poligon dosyasında yer almasının istenmemesidir.

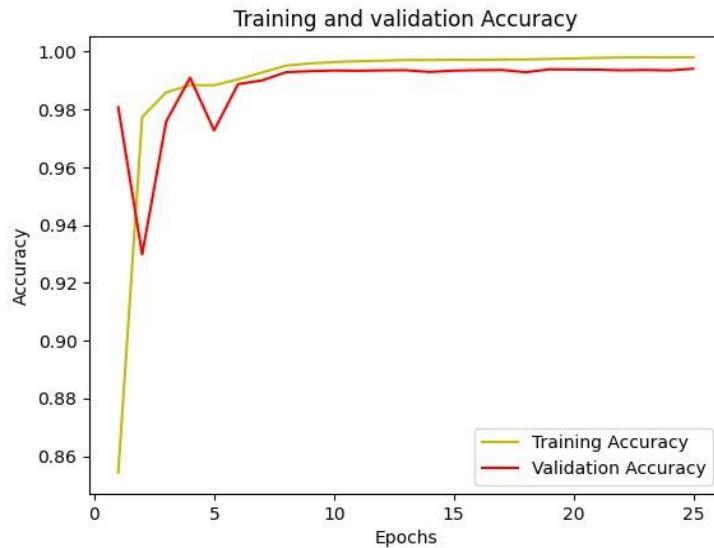
```
for i in range(height):
    for j in range(width):
        pixel = rgb_mask[i, j, :]
        if np.array_equal(pixel, [0, 0, 142]): # Car
            single_channel_mask[i, j] = 255 # Grayscale value for car
        elif np.array_equal(pixel, [255, 0, 0]): # Rider
            single_channel_mask[i, j] = 255 # Grayscale value for rider
        elif np.array_equal(pixel, [220, 20, 60]): # Human
            single_channel_mask[i, j] = 255 # Grayscale value for human
```

Resim [1]

İnsan ve arabaların yer aldığı siyah-beyaz formdaki maskeler, yine 'RGB_to_GrayScale_AND_mask_to_YoloTxt.ipynb' kod dosyası kullanılarak Yolov8'de kullanılmak üzere uygun txt poligon formatına dönüştürülmüştür.

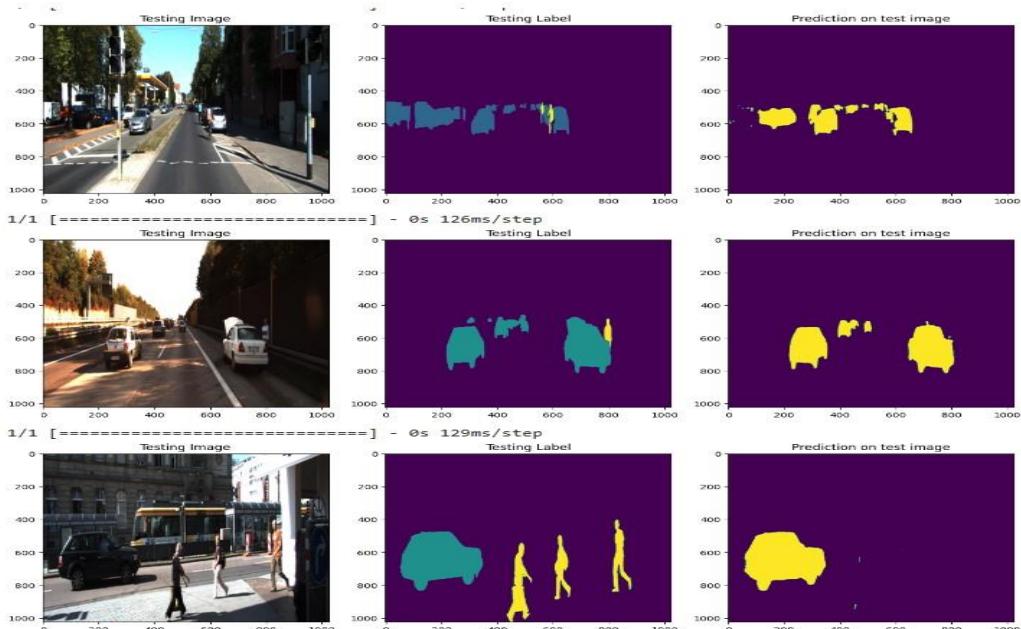
2: Segmentasyon Modeli Geliştirme

İlk segmentasyon çalışmaları yapılırken unet üzerinde resnext101 modeli kullanılarak eğitimler gerçekleştirılmıştır. Adam, SGD VE RMSprop optimizerlarının model üzerindeki başarıları kıyaslanmıştır ve en başarılı optimizer Adam olarak belirlenmiştir. Resim[2]’de unet-resnext101 modelinin train ve val sonuçları gösterilmektedir.



Resim[2]

Resim[3]’te ise resnext101 modelinin test image’ler üzerinde predict yapılması işleminin sonuçları gösterilmektedir.



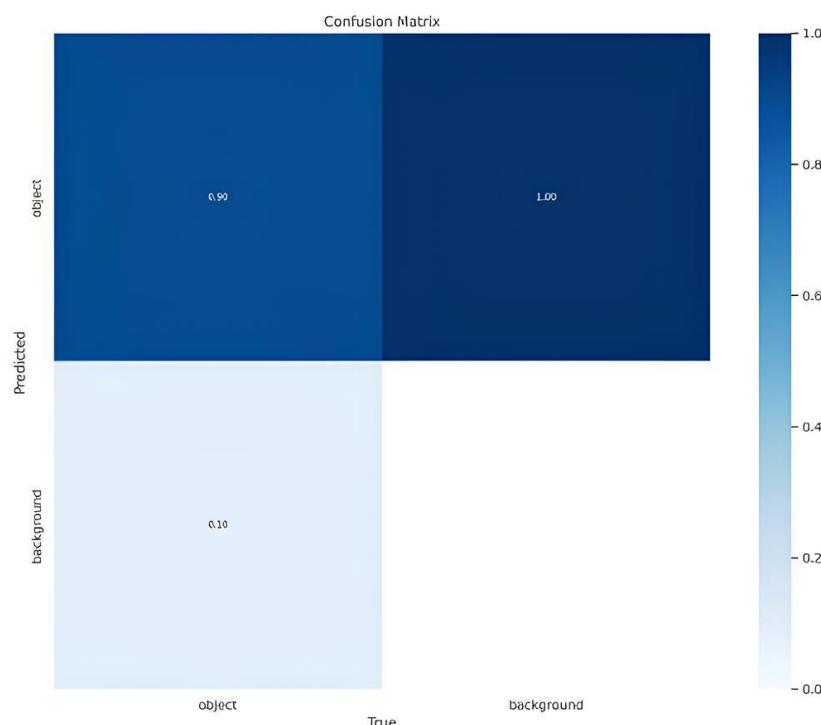
Resim[3]

Fakat unet-resnext101 modelinin eğitilmiş ağırlıkları olan *.hdf5 dosyası 405 mb gibi çok ağır ve fazla parametre içermesinden dolayı, video dosyalar üzerinde çok yavaş ve gecikmeli çalışabileceği öngörülülmüş bu nedenle YOLOv8'de segmentasyon aşamasına devam edilmesi kararlaştırılmıştır.

YOLOv8'de segmentasyon aşamasında Resim[4]'te de görüldüğü üzere yolov8n-seg, yolov8n-seg, yolov8m-seg, yolov8l-seg ve yolov8x-seg modellerinin eğitimleri ayrı ayrı gerçekleştirilmiştir ve en başarılı model Resim[5]'te konfüzyon matrisi verildiği üzere yolov8n-seg olmuştur.

Model	size (pixels)	mAP ^{box} 50-95	mAP ^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Resim[4]



Resim[5]

Eğitimi gerçekleştirilmiş olan YOLOv8 segmentasyon modelinin örnek predict sonucuna Resim[6]’da yer verilmiştir.



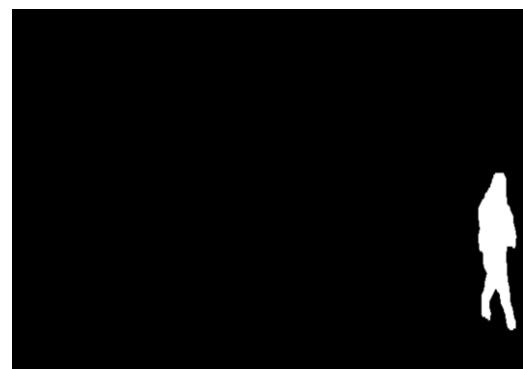
Resim[6]

3: Bounding Box Tespiti

Segmentasyon sonuçları üzerinde blob'ların tespiti ve bounding box'larının oluşturulması aşamasıdır. !yolo task=segment mode=predict . . . save_crop=True modülünde, YOLO ile predict etme aşamasında save_crop True seçilerek tespit edilen her nesnenin kaydedilmesi sağlanmıştır. Örnekler, Resim[7] ve Resim[8] de olmak üzere gösterilmektedir.



Resim[7]



Resim[8]

4: Sınıflandırma Modeli Geliştirme

Kaydedilen her nesne araba ve insan üzere iki sınıfaya ayrılmıştır. Daha sonrasında train-val-test split işlemleri yapılarak hazırlanan veri seti, sınıflandırma için hazır duruma getirilmiştir. Sınıflandırma aşamasında YOLOv8'in yolov8n-cls modeli kullanılmıştır. Image size 224 ve epoch sayısı 50 olarak belirlenmiştir.

Sınıflandırma işleminde sonuç olarak Tablo[1]’deki değerler elde edilmiştir.

epoch	train/loss	metrics/accuracy	metrics/accuracy_to_p5	val/loss	lr/pg0	lr/pg1
50	0.03939	0.92683	21	0.10191	0.1	0.017

```

names_dict = {0: 'Car', 1: 'Person'}
probs = results[0].probs.data.tolist()
print(names_dict)
print(probs)
print(names_dict[np.argmax(probs)])

```

Resim[9]

```

{0: 'Car', 1: 'Person'}
[0.9254406690597534, 0.07455933094024658]
Car

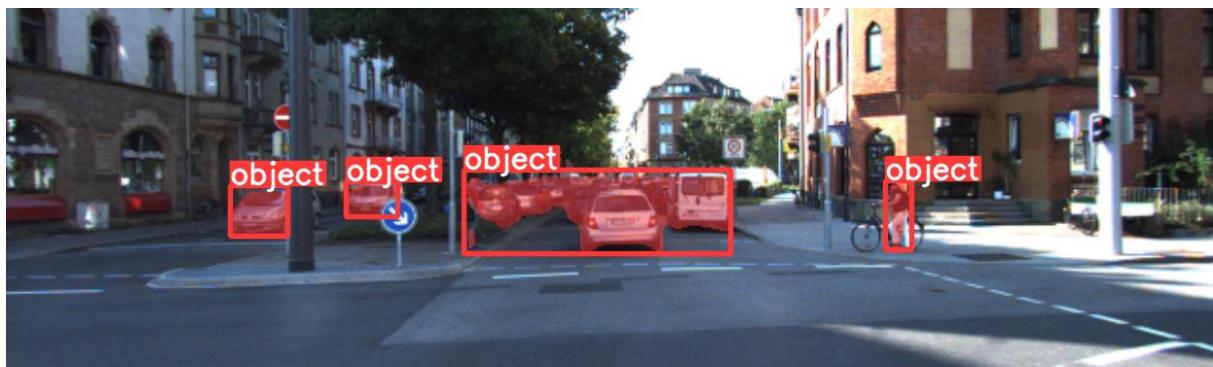
```

Resim[10]

Resim[9]’da görüldüğü üzere, classification modelini predict yapma işlemi gerçekleştirilmiştir. Numpy kütüphanesinin argmax fonksiyonu kullanarak 1.00'a en yakın probs seçilerek predict işleminin sonucu Resim[10]’da görüldüğü üzere ekrana bastırılmıştır. ./runs/classify/train/weights/train/classification_last_model.pt adres yolunda yer alan eğitilmiş model kaydedilerek segmentasyondan sonra sınıflandırma işleminin de yapılması için `Visea_Yapay_Görme_Test_Projesi_v3.ipynb` notebook’un da kullanılmıştır.

5: Obje Tespiti Uygulaması Geliştirme

Son aşama olan objelerin bounding box’larıyla birlikte sınıflarının da yazılması işlemiyle birlikte proje tamamlanmıştır. Bu aşamayla, projenin amacına ulaşarak yapay görme uygulamasının tamamlanması sağlanmıştır. Projelin son aşamasında tamamlanması gereken tek eksik işlem, Resim[11]’de de görüldüğü üzere sınıflandırma aşamasında tespit edilen sınıfların gerekli bounding box’ların üzerine yazılmasıdır.



Resim[11]

Sonuç

Projelin son aşamasındaki, tespit edilen sınıfların bounding box’ların yazılması haricinde her bir adım, başarıyla tamamlanarak yapay görme uygulamasının oluşturulması sağlanmıştır. Projeyi oluşturan kodlarda yapılan işlemler, adım adım yorum satırlarıyla açıklanmakta ve projelin anlaşılabilirliği kolaylaştırılmaktadır.