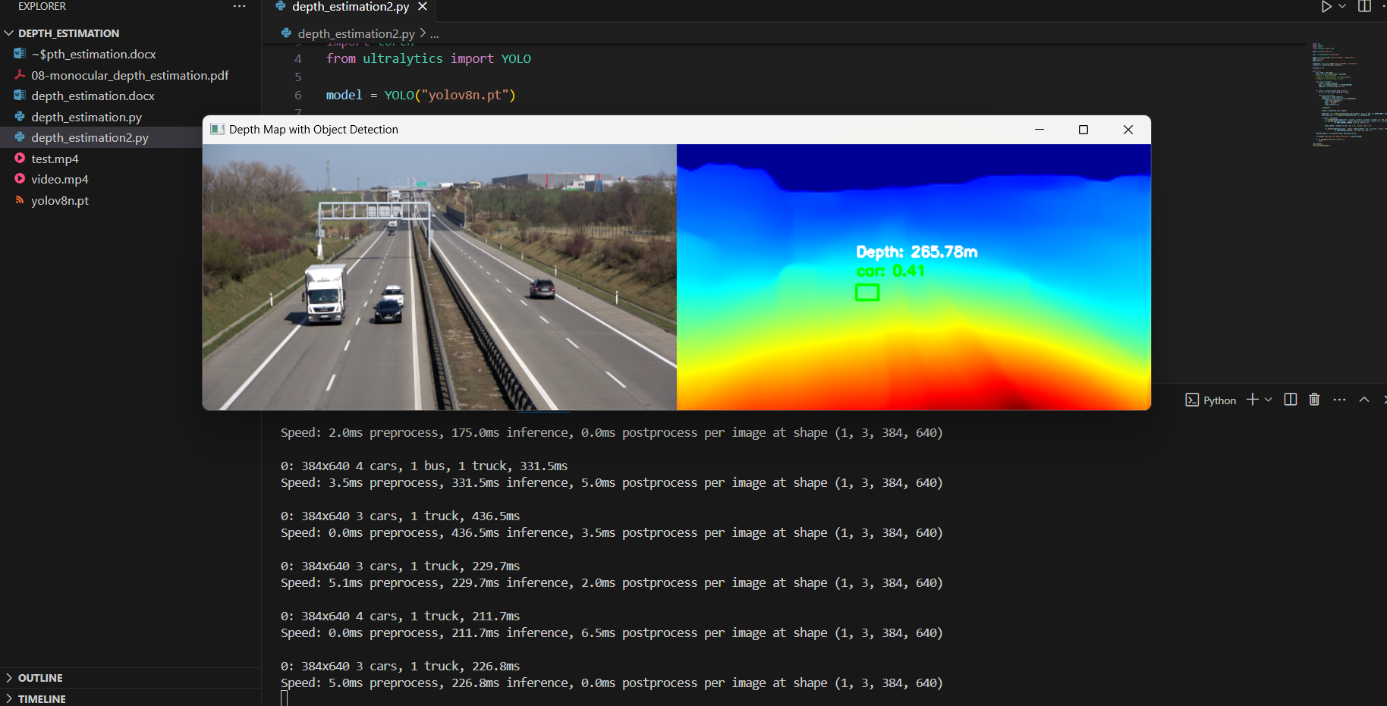
**Depth Estimation Nedir?**

Depth estimation(derinlik algısı), 2 boyutlu görüntüleri 3 boyutlu algılayarak nesnelerin merkez noktasına uzaklıklarını tahmin etmek için kullanılır.

**Depth Estimation Türleri**

Stereo yani çoklu ve mono yani tekli olmak üzere temelde 2 tane depth estimation türü mevcuttur.

İki türü üzerinde de durulacağı gibi öncelikle monoküler ile bazı denemeler yapılmıştır.



**Şekil-1**

Şekil-1 de torch kütüphanesinin içinde bulunan MıDaS monocular depth estimation modeli değerlendirme modunda kullanılmış yolov8 nesne tespiti ile entegre kullanılmıştır. Ardından tespit edilen nesnenin orta noktasının derinliği (Deneme aşamasında olduğu için değerler çok doğru olmayabilir) ölçülmüştür.

**Yolo Nedir?**

Önce Yolo denilen şeyin ne olduğu açıklanmak istenmektedir. Yolo aslında bir mimari VGG16, VGG19, ResNet(Residual Neural Network) gibi. Sayılan herhangi bir mimari ile model eğitilirken Cifar, Mnist vb tensorflow-keras’ın içine gömülü bazı verisetleri kullanılıp bu verisetlerinden transfer learning(aktarımlı öğrenme) yapılarak model eğitilir. Ama yolo bunlardan farklı olarak kendi verisetini de içinde bulundurur ve son zamanlarda da ultralytics denilen bir kütüphanenin altında toplanmıştır.

**Neden Yolov8?**

Bu konuda düşüncemiz diğerlerinden oldukça farklıdır. Neden mi?

Çünkü herhangi bir nesnenin tespit edilirken confidince(kısacası güven skoru) yüzde 90 dan 95’e çıkarmak büyük başarı olarak görülür. Ama bu modellerin genelde yüzde 30 ve üstü insan dediği nesne büyük olasılıkla insandır. O yüzden yolonun herhangi bir modeli kullanılabilir ve asla bir farkı olmayacaktır bu anlamda. Yüzde 5 lik bir başarı için modelin ağırlığı artıyor ve fps düşüyor sonuç olarak iyi olmayan bir sonuç ortaya çıkıyor.

TensorRT ve onnx denen iki şey var modelin hafifletip daha hızlı çalışmasını sağlıyor. Tabi model hafiflediği için bir şeyler kaybediliyor. Ama bu işlem optimize bir şekilde yapılırsa modelin kullanılırken verdiği fps gerek cpu gerek gpu gerekse de tpu üzerinde artış sağlayacaktır. Bu iki yöntem yolo modelleri üzerinde uygulanırken oldukça uyumlu çalışmaktadır o yüzden tercih edilebilir.

**Yolov8+Depth Estimation**

Şekil-1 de derinlik algısı yapılan görüntü üzerindeki her noktanın derinliği bize lazım değil. Otonom sürüş yapan herhangi bir vasıta sadece istediği yani yolo gibi bir modelle tespit ettiği nesnelerin derinliğini bilmek ister. Biz de yolo ile tespit edilen nesnelerin derinliğini vermek için torch kütüphanesine gömülü MıDaS monocular depth estimation modelini yolo ile entegre kullanıyoruz. Tabi şu an deneme amaçlı her cismin orta noktası verilmiştir.

**Segmentation**

Nesnelerin piksel tabanlı olarak bölütlenmesini ihtiva eden bir yöntemdir. Depth estimation ile entegre etmek için piksel tabanlı olduğu için uyumluluk oranı yüksek görülmektedir.

Segmentasyon ile ilgili detaylı yazımızı[1] medium hesabımızda yayınladık. İncelemek isterseniz diye linkini de bırakıyoruz.

**Point Cloud**

Nokta bulutu olarak adlandırılan yöntemdir. Nasıl object detection yaparken dikdörtgenler, segmentasyon yaparken piksel tabanlı bölmeler, point cloudda ise nokta bulutları sınıflandırılır. Bu yöntemin de depth estimation ile entegresi uyumlu görülmektedir.

**Biz Ne Yapmak İstiyoruz?**

Özellikle bu konunun hakkında çok fazla kaynak olmadığından ve olanların da genelde yabancı kaynak olup son birkaç yılda yayınladığından bu konuda gerek yerel gerekse küresel çapta okunup değer görmesi adına bir makale yazılması istenmektedir.

**Kaynakça**

[1] <https://medium.com/@atasyunusemre9603/bilgisayarl%C4%B1-g%C3%B6r%C3%BCde-i%CC%87leri-b%C3%B6l%C3%BCtleme-y%C3%B6ntemleri-75d6cb3c28c5>