

# YILDIZ TENKİK ÜNİVERSİTESİ

## YILDIZ ROVER

### ÖDEV 3

#### LACHIN TALAEI

**Github: <https://github.com/LachinTala/Yildizroverhw3>**

#### 1. Giriş

Klasik görüntü işleme yöntemleri renk, şekil veya kenar tabanlı özellikler kullanarak nesne tespiti yapabilmektedir. Ancak bu yöntemler, farklı ışık koşulları, açı değişimleri, kısmi örtülme ve karmaşık arka plan gibi durumlarda yetersiz kalmaktadır. Trafik işareti tespiti gibi kritik uygulamalarda bu tür sınırlılıklar büyük hata oranlarına yol açabilmektedir. Bu noktada yapay zekâ tabanlı çözümler devreye girmektedir. Derin öğrenme yöntemleri, veriden otomatik olarak ayırt edici özellikleri öğrenebilmekte ve böylece nesneleri daha doğru ve güvenilir şekilde tespit edebilmektedir. Bu çalışmada, derin öğrenme tabanlı modeller kullanılarak STOP işaretlerinin tespiti gerçekleştirilmiştir.

#### 2. Arka Plan

##### 2a) Yapay zekâ tabanlı çözümlerin avantajları

Nesne tespitinde yapay zekâ tabanlı yöntemlerin en önemli avantajı, özellik mühendisliğine gerek bırakmadan doğrudan veriden öğrenme yapabilmeleridir. Böylece farklı koşullarda, farklı kamera açıları veya ışıklandırma altında dahi başarılı sonuçlar elde edilebilmektedir. Ayrıca derin öğrenme modelleri, uçtan uca optimizasyon sayesinde hata oranlarını azaltmakta ve süreci hızlandırmaktadır. Özellikle YOLO ailesi gibi gerçek zamanlı çalışan modeller, otonom araçlar veya sürücü destek sistemleri gibi zaman kritikli uygulamalarda büyük avantaj sağlamaktadır.

##### 2b) Derin öğrenme ve öğrenme süreci

Derin öğrenme, makine öğrenmesinin bir alt dalı olup çok katmanlı yapay sinir ağlarıyla temsil öğrenimini hedefler. Özellikle evrişimli sinir ağları (CNN), görüntü işleme görevlerinde öne çıkar. Öğrenme süreci ileri yayılım, kayıp fonksiyonunun hesaplanması, geri yayılım ve ağırlıkların güncellenmesi adımlarından oluşur. Bu döngü, belirli sayıda epoch boyunca devam eder. Klasik makine öğrenmesi yöntemlerine kıyasla derin öğrenmenin en büyük avantajı, manuel özellik çıkarımına ihtiyaç duymadan veriden

hiyerarşik özellikler öğrenmesidir. Ayrıca büyük veri setlerinde daha yüksek doğruluk sağlar; küçük veri setlerinde ise transfer öğrenme sayesinde güçlü performans gösterir.

## **2c) Pre-trained modeller ve karşılaştırma**

Pre-trained modeller, büyük veri setlerinde önceden eğitilmiş ve yeniden kullanılabilen sinir ağı modelleridir. Bu sayede küçük veri setleriyle hızlı ve etkili bir şekilde yeniden eğitilebilirler. Örneğin Faster R-CNN, iki aşamalı yapısıyla yüksek doğruluk sağlarken hız açısından dezavantajlıdır. SSD, tek aşamalı yapısıyla daha hızlıdır ancak küçük nesnelerde performansı düşebilir. YOLO ailesi ise tek aşamada hem sınıflandırma hem de kutu regresyonu yaparak hem yüksek hız hem de yüksek doğruluk sunar. Özellikle YOLOv8 gibi son sürümler, gerçek zamanlı uygulamalar için en uygun seçeneklerden biridir. Bu nedenle bu çalışmada YOLO tabanlı yaklaşım tercih edilmiştir.

## **3. Yöntem ve Deneysel Çalışmalar**

### **3a) Veri seti**

Çalışmada kullanılan veri seti toplam 652 STOP işareti görselinden oluşmaktadır. Bu veriler eğitim, doğrulama ve test olmak üzere üç alt kümeye ayrılmıştır: 456 eğitim, 131 doğrulama ve 65 test görseli. Bu ayırım, modelin hem öğrenme sürecinde yeterli veriye ulaşmasını hem de bağımsız veriler üzerinde değerlendirilebilmesini sağlamıştır. Görseller farklı açılardan, uzaklıklardan, ışık koşullarından ve arka planlardan seçilmiştir. Test setinde ayrıca vandalize edilmiş, kısmi örtülmüş veya farklı yazı tipleriyle değiştirilmiş işaretler de yer almaktadır. Bu çeşitlilik, modelin yalnızca ideal koşullarda değil, gerçek dünyada karşılaşabileceği zorluklar altında da test edilmesini mümkün kılmıştır.

### **3b) Model parametreleri**

Modelin eğitiminde epoch, batch size, image size ve optimizer gibi parametreler önemli rol oynamıştır. Epoch, tüm eğitim verisinin modelden geçme sayısını ifade eder ve aşırıya kaçılması durumunda overfitting riski doğurabilir. Batch size, her adımda işlenen görüntü sayısını belirler; GPU belleğine göre ayarlanmıştır. Image size, modelin aldığı giriş çözünürlüğünü belirler; burada 640 piksel seçilerek ayrıntı ve hız arasında denge kurulmuştur. Optimizasyon sürecinde kullanılan algoritma (örneğin Adam veya SGD), ağırlıkların güncelleyerek kayıp fonksiyonunu minimize etmiştir. Bu parametreler uygun seçildiğinde, model hem doğruluk hem de genelleme açısından tatmin edici sonuçlar vermiştir.

### **3c) Eğitim ve test süreci**

Model, Roboflow ortamında eğitilmiş ve test edilmiştir. Test aşamasında confidence threshold %50 olarak belirlenmiştir; yani yalnızca güven oranı %50'nin üzerinde olan

tahminler kabul edilmiştir. Bu süreçte model, standart STOP işaretlerini yüksek doğrulukla tespit etmiştir. Ancak yapılan denemelerden birinde sahte bir tabela, sekizgen şekli ve kırmızı rengi nedeniyle yüksek güvenle “fake stop sign” sınıfı olarak etiketlenmiştir. Bu durum, modelin şekil ve renk benzerliklerinden güçlü şekilde etkilenebildiğini, fakat bazı sahte durumlarda yanılabilirliğini göstermektedir. Dolayısıyla bu tür yanlış pozitifleri azaltmak için eğitim verisine daha çeşitli örnekler eklemek faydalı olacaktır.

### **3d) Sonuçların değerlendirilmesi**

Eğitim sonunda modelin doğrulama seti üzerindeki başarısı oldukça yüksektir. Ortalama mAP@50 değeri %98 olarak ölçülmüş, “stop sign” sınıfında %99, “fake stop sign” sınıfında %100 ve “vandalized stop sign” sınıfında %95 doğruluk elde edilmiştir. Precision %93.9, recall ise %92.6 olarak gerçekleşmiştir. Bu değerler, modelin hem yanlış pozitifleri hem de yanlış negatifleri düşük seviyede tuttuğunu göstermektedir. Eğitim grafikleri incelendiğinde, mAP değerinin ilk 50 epoch içerisinde hızla yükseldiği ve sonrasında %0.9 seviyelerinde sabitlendiği görülmektedir. Box loss, class loss ve object loss değerleri zamanla düşerek düşük seviyelerde dengelenmiştir. Bu eğilim, modelin kararlı bir şekilde öğrendiğini ve overfitting yaşamadığını göstermektedir. Sonuç olarak, geliştirilen derin öğrenme tabanlı model, STOP işareti tespiti görevinde yüksek doğruluk ve güvenilirlik sergilemiştir.

### **4. Sonuç**

Bu çalışmada, derin öğrenme tabanlı bir model kullanılarak STOP işareti tespiti gerçekleştirilmiştir. Kullanılan YOLO tabanlı yaklaşım, hem hız hem de doğruluk açısından tatmin edici sonuçlar vermiştir. Eğitim sürecinde elde edilen yüksek mAP, precision ve recall değerleri, modelin gerçek dünyada güvenilir şekilde kullanılabileceğini göstermektedir. Bununla birlikte, sahte veya tahrif edilmiş işaretlerde görülen bazı yanlış pozitifler, gelecekte verinin daha da çeşitlendirilmesi gerektiğine işaret etmektedir. Genel olarak, bu çalışma derin öğrenmenin nesne tespitinde klasik yöntemlere kıyasla ne denli güçlü bir çözüm sunduğunu açıkça ortaya koymuştur.

Stop Sign Detection 1 1

Visualize

Download Weights

Deploy Model

Model URL:  
stop-sign-detection-1-7ajga/1

Updated On:  
9/10/25, 2:23 PM

Checkpoint:  
-  
Model Type:  
Roboflow 3.0 Object Detection (Accurate)

Dataset Version:  
stopsign1

Metrics

Valid Set

mAP@50  
97.8%

Precision

93.9%

External

Recall  
92.6%

Preview Model

Samples from Test Set



View Test Set

Upload Image or Video File

Drop file here or

Select File

Image URL

Paste a link...

Try With Webcam

Try On My Machine



1 object detected

Confidence Threshold: 50%

0% 100%

Overlap Threshold: 50

0% 100%

Opacity Threshold: 75

0% 100%

Label Display Mode:

Draw Confidence

```
{
  "predictions": [
    {
      "x": 200,
      "y": 167,
      "width": 298,
      "height": 306,
      "confidence": 0.971,
      "class": "fake stop sign",
      "class_id": 0,
      "detection_id": "290ac6ba-d9aa-4"
    }
  ]
}
```

Copy

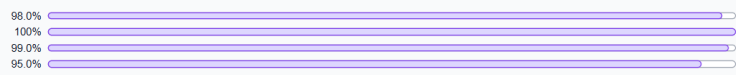
652 Total Images

Train 456 Valid 131 Test 65



Validation Set Test Set

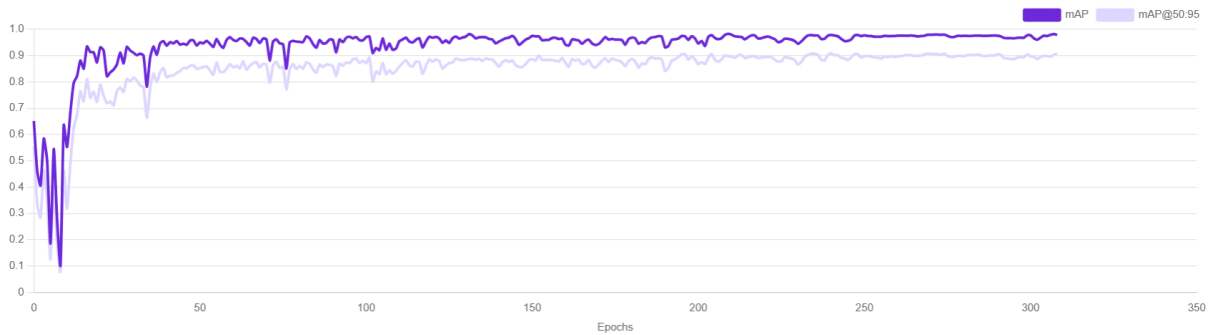
all  
fake stop sign  
stop sign  
vandalized stop sign



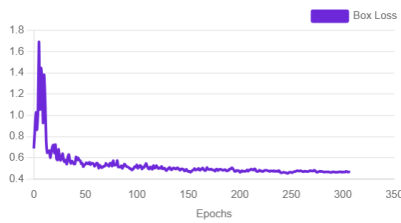
### Training Graphs

Advanced Graphs

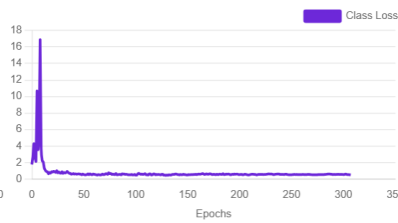
#### Model Performance



#### Box Loss



#### Class Loss



#### Object Loss

