



T.C.

SAMSUN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ (YL) TEZLİ PROGRAMI

**BÖLGESEL TABANLI YÖNTEMLER KULLANILARAK**

**NESNE TESPİTİ YAPILMASI**

**(YOLO V5 KULLANILMIŞTIR)**

DERİN ÖĞRENME DERSİ FİNAL SUNUMU

BAHAR DÖNEMİ

Hazırlayan

Ferat EFİL

210708010

Öğretim Üyesi

Doç. Dr. Muammer TÜRKOĞLU

Samsun

Haziran , 2022

## İçindekiler

01. Dataseti Oluşturulması .....	3
02.Google Colaboratory Kullanılarak Proje Oluşturulması – Sistemin Eğitilmesi – Test Edilmesi .....	4
03.Model Değerlendirilmesi .....	6

## 01. Dataseti Oluşturulması

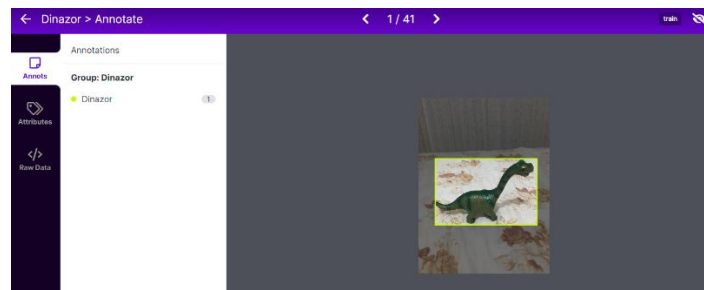
Bu çalışmada bölgesel tabanlı yöntemler kullanılarak nesne tespiti yapabilmek için YOLO V5 kullanılmıştır.

YOLO V5 kullanılarak tespit edilmek istenen nesne oyuncak bir dinazordur. Veri seti oluşturmak için oyuncak dinazorun değişik açılardan, uzaktan ve yakından resimleri çekilmiştir.



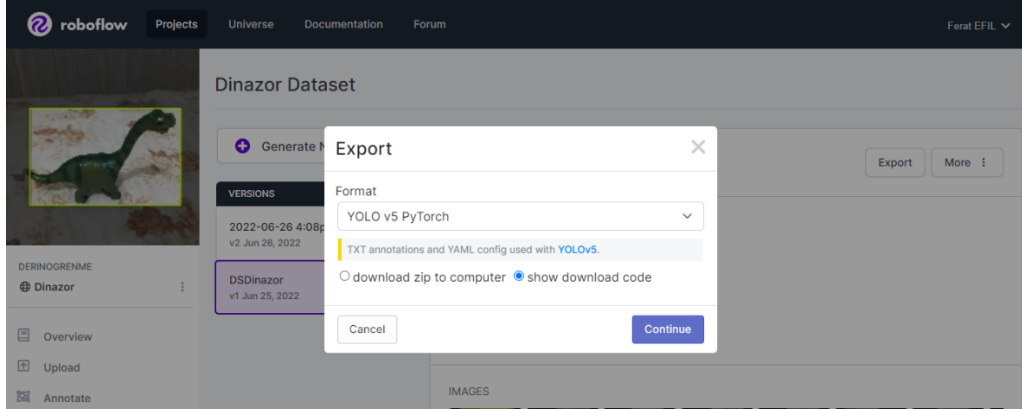
Daha sonra çekilen resimleri etiketlemek için ücretsiz olarak kullanılabilen <https://roboflow.com/> sitesi üzerinden etiketleme işlemi yapılmıştır.

Oluşturulan Derin Öğrenme çalışma alanı içerisine “Dinazor Dataset” adında bir veri seti oluşturulmuştur. Oluşturulan bu veri setine telefon aracılığıyla çekilen 42 adet ham resim Upload edilmiştir. Dataset alanına yüklenen bütün resimler tek tek seçilerek işaretlenmiş ve “Dinazor” olarak etiketlenmiştir.



Etiketlenen resimlere Preprocessing (Yeniden boyutlandırma vb), Augmentation (Resmin bir kısmına odaklanma, ters ya da yan çevirme vb işlemler) gibi işlemler yapılarak verilerin artırılması işlemi yapılmış ve toplamda elimizde 113 adet etiketli veri olmuştur.

Etiketlenen verilere Export işlemi yapılarak veri setinin dışarı alınması sağlanmıştır.



Projemizde kullanabileceğimiz veri setimiz hazır. Veri setimize ulaşmak için aşağıdaki linklerden birini kullanabilirsiniz :

Sadece Etiketlenmiş Verileri İndirmek İçin :

[https://drive.google.com/drive/folders/1ZokytKX\\_VjxivF-WkpZ0nTl\\_Ro5YWtJJ?usp=sharing](https://drive.google.com/drive/folders/1ZokytKX_VjxivF-WkpZ0nTl_Ro5YWtJJ?usp=sharing)

Sadece Ham Resimleri İndirmek İçin :

[https://drive.google.com/drive/folders/1ENcFmIhz\\_6eBeqx0ZeH\\_jLOQiZaRbLpO?usp=sharing](https://drive.google.com/drive/folders/1ENcFmIhz_6eBeqx0ZeH_jLOQiZaRbLpO?usp=sharing)

Sadece Sistemin Hiç Görmediği Test Verilerini İndirmek İçin :

<https://drive.google.com/drive/folders/1ZvTU0xNa2oEwP5zBdevfkhH8cpzzGh6U?usp=sharing>

Tamamı İçin (Ham Veri, Etiketlenmiş Veri ve Sistemin Hiç Görmediği Test Verileri) :

<https://drive.google.com/drive/folders/1xnZz4z2ktO9eTBda64D7pqzyPw0MZoCg?usp=sharing>

## 02.Google Colaboratory Kullanılarak Proje Oluşturulması – Sistemin Eğitilmesi – Test Edilmesi

Sistemin oluşturulması, eğitilmesi ve test edilmesi için Google Colaboratory uygulaması kullanılmıştır. Uygulamanın kodlarına erişmek için Github ve Colab hesabını ziyaret edebilirsiniz:

<https://github.com/ferate/DeepLearning>

<https://colab.research.google.com/gist/ferate/ad22a0b9e2f3dc180d4caa5ca7218c63/dinazorubul.ipynb>

Öncelikle YOLO V5 kütüphanesi kullanacağımız için projemize dahil ediyoruz :

```
!git clone https://github.com/ultralytics/yolov5 # repository'i klonluyoruz
%cd yolov5
!pip install -qr requirements.txt # gerekli kütüphaneleri install ediyoruz
!pip install -q roboflow # roboflow u install ediyoruz

import torch
import os
from IPython.display import Image, clear_output # resimleri görüntülemek için

print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")
```

Cloning into 'yolov5'...

remote: Enumerating objects: 12281, done.

remote: Total 12281 (delta 0), reused 0 (delta 0), pack-reused 12281

Receiving objects: 100% (12281/12281), 12.08 MiB | 32.47 MiB/s, done.

Resolving deltas: 100% (8476/8476), done.

/content/yolov5

596 kB	29.3 MB/s
145 kB	37.1 MB/s
178 kB	58.4 MB/s

Veri setimizi roboflow üzerinden ekleyeceğimiz için gerekli kütüphaneleri ekliyoruz :

```
[ ] from roboflow import Roboflow
rf = Roboflow(model_format="yolov5", notebook="ultralytics")
```

Roboflowda oluşturduğumuz veri setini indiriyoruz sisteme dahil ediyoruz :

```
[ ] #roboflowdan aldığımız api kodlarını kopyalıyoruz çalıştırınca indirecek
rf = Roboflow(api_key="whj1IsvctXGUUhBxMu9f")
project = rf.workspace("derinogrenme").project("dinazor")
dataset = project.version(1).download("yolov5")

loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in /content/datasets/Dinazor-1 to yolov5pytorch: 100% [1885975 / 1885975] bytes
Extracting Dataset Version Zip to /content/datasets/Dinazor-1 in yolov5pytorch:: 100% [238/238] [00:00<00:00, 2172.51it/s]
```

YOLO V5 Kütüphanesi kullanarak sistemimizi eğitiyoruz :

```
# 416 resmi 16 lık batchler halinde 100 epochda eğitecek saatler sürebilir lütfen sabırlı olun ve aralıklarla ekranı
!python train.py --img 416 --batch 16 --epochs 100 --data {dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Bu işlem bilgisayarımızın donanımına ve veri setimizin büyüklüğüne göre biraz zaman alabilir.

Sistemin tespit ettiği en iyi modele göre eğitim için kullanmadığımız verilerimizi test edebiliriz :

```
#test resimlerinizi modele sokuyoruz ve sonuçlar en sonda belirtilen dosya konumuna kaydediliyor
#her farklı resim veya video eklediğinizde en sondaki exp1 exp2 diye artar. runs/detect/exp
!python detect.py --weights /content/yolov5/runs/train/exp/weights/best.pt --img 416 --conf 0.70 --source {dataset.location}/valid/images
```

Sistemin hiç görmediği resimler üzerinden test işlemi yapıldığında tahminleri şu şekilde görebiliriz :

