

# El Sayıları Tanıma Sistemi (YOLOv11n Tabanlı)

## Bu Proje Nedir

Bu proje, kamera üzerinden alınan görüntülerdeki el işaretlerini gerçek zamanlı olarak algılayarak kaç parmak gösterildiğini tahmin eder. YOLOv11 nesne algılama modeli ve PyTorch kullanılarak geliştirilmiştir. Sistem GPU destekli çalışabilir ve OpenCV ile görselleştirme sağlar. Bu proje kendini geliştirmek isteyen her öğrenci tarafından kullanılabilir.

## Nasıl Çalışıyor

- [ Görüntü toplama ]
- [ Veri seti hazırlama ]
- [ Veri setinde eğitim için gerekli işlemlerin yapılması ]
- [ Modeli görmek için gerekli kodlar ]
- [ Kodları çalıştırma ]
- [ Kamera ile görüntü alma ]
- [ Görüntü üzerinden tespit yapma ]
- [ YOLOv11 tahmini ]
- [ Sayı algılama ]
- [ Görselleştirme ve FPS ]
- [ Tahmin edilen sonucu görme ]

## Kullanılan Teknolojiler

Python : Programlama dili  
OpenCV : Görüntü işleme  
PyTorch : Derin öğrenme çatısı  
Ultralytics YOLO : YOLOv11 modeli için Python paketi  
Roboflow : Projede büyük önemi olan veri seti hazırlamak  
Pycharm : Kodları yazdığım ve test ettiğim platform  
Kamera : Laptopda bulunan webcam

## ◆ Neden bu proje ◆

Geliştirmeye çok açık olması  
Arkadaşıma tavsiye ettiğim ilk yolo projem

Veri toplama konusundaki kolaylığı  
Kendimi bu proje ile geliştirmek

Projenin github linki ( örnek çalışma videosu githubda mevcuttur videoyu görmek için github reposunu indirmanız gerekmekte)



Github için tarat

## Projeden örnek kod ( test aşaması )

```
from ultralytics import YOLO
import cv2
import time
import torch
import os

assert os.path.exists("elSayilariYOLOv11nv2.pt"), "Model dosyası bulunamadı!"
model = YOLO("elSayilariYOLOv11nv2.pt")
device = "cuda" if torch.cuda.is_available() else "cpu"
model.to(device)
print(f"Model is using device: {device}")

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 360)

prev_time = time.time()
predict_every_n_frames = 2
frame_count = 0
fps_list = []
last_result = None

try:
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        current_time = time.time()
        fps = 1 / (current_time - prev_time)
        prev_time = current_time
        fps_list.append(fps)
        if len(fps_list) > 30:
            fps_list.pop(0)
        avg_fps = sum(fps_list) / len(fps_list)

        if frame_count % predict_every_n_frames == 0:
            with torch.no_grad():
                results = model.predict(source=frame, imgsz=640, conf=0.5, stream=True)
                last_result = next(results)

        frame_count += 1
        annotated_frame = last_result.plot() if last_result else frame

        cv2.putText(annotated_frame, f"FPS: {int(avg_fps)}", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow("YOLOv11 ile El Sayıları", annotated_frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
finally:
    cap.release()
    cv2.destroyAllWindows()
```

**Proje sahibi :** Ömer Çınar Yaman öğrenci no: 2405902004/Yapay zeka  
**Proje çalışanı :** operatörlüğü