BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

YOLOV8 İLE ZEYTİN AĞAÇLARINDAKİ ZARARLI VE ZARARSIZ BÖCEKLERİN TESPİTİ VE TAKİBİ

LİSANS BİTİRME ÇALIŞMASI

Wahib Hael Abdulwareth MOQBEL

20360859109

Bilgisayar Mühendisliği Bölümü

Danışman: Prof. Dr. Hayri Volkan AKGUN

TEZİN SAVUNULDUĞU AY YIL

Haziran 2024

BTÜ, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nün 20360859109 numaralı öğrencisi Wahib MOQBEL, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "YOLOV8 İLE ZEYTİN AĞAÇLARINDAKİ ZARARLI VE ZARARSIZ BÖCEKLERİN TESPİTİ VE TAKİBİ" başlıklı bitirme çalışmasını aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Danışmanı : **Prof. Dr. Hayri Volkan AGUN**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğr. Ergün GÜMÜŞ**
Bursa Teknik Üniversitesi

Arş. Gör. ESMA İBİŞ
Bursa Teknik Üniversitesi

Savunma Tarihi : 13 Haziran 2024

BM Bölüm Başkanı : Doç. Dr. Turgay Tugay BİLGİN

Bursa Teknik Üniversitesi 13/06/2024

İNTİHAL BEYANI

Bu bitirme çalışmasında görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, bitirme çalışması içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri bitirme çalışmasında kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: Wahib MOQBEL

İmzası : 

ÖNSÖZ

Bu bitirme çalışmasının hazırlanması sürecinde değerli katkılarını ve rehberliğini esirgemeyen danışman hocam Sayın Dr. Öğr. Hayri Volkan AGUN'a en içten teşekkürlerimi sunarım

Ayrıca, bu projede birlikte çalıştığım Teknofest takım arkadaşlarıma da teşekkür ederim. Takım çalışması ruhu ve iş birliği sayesinde, projeyi birkaç aşamasını başarıyla tamamlayabildik. Her birine ayrı ayrı teşekkür ederim.

Bu çalışmanın gerçekleşmesinde emeği geçen ve desteklerini esirgemeyen tüm kurum ve kuruluşlara da teşekkürlerimi sunarım. Destekleri olmadan bu projeyi gerçekleştirmek mümkün olmazdı.

Sevgili arkadaşım Vural Bayraklı'ya da ayrıca teşekkür etmek isterim. Kendisinin moral ve motivasyon desteği, projenin zorlu dönemlerinde bana yardımcı oldu. Her zaman yanımda olduğu ve desteklerini esirgemediği için kendisine minnettarım.

Son olarak, bu çalışmanın zeytin ağaçlarındaki zararlı ve zararsız böceklerin tespit ve takibi konusunda faydalı olmasını umuyor ve bu konuda çalışma yapacaklara ışık tutmasını diliyorum.

Haziran 2024 Wahib MOQBEL

İÇİNDEKİLER

Sayfa

KISALTMALAR

YOLO	**:** You Only Look Once
SOTA	**:** State Of The Art
IoU	**:** Intersection over Union
mAP	**:** Mean Average Precision
P	**:** Precision (Hassasiyet)
R	**:** Recall (Duyarlılık)
FPS	**:** Frames Per Second
F1 Score	**:** F1 Skoru
GFLOPs	: Giga Floating Point Operations per Second
CBS	:Convolution, Batch Normalization, SiLU Activation
SPPF	:Spatial Pyramid Pooling Function

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

Y.OLOV8 İLE ZEYTİN AĞAÇLARINDAKİ ZARARLI VE ZARARSIZ BÖCEKLERİN TESPİTİ VE TAKİBİ

ÖZET

Bu bitirme projesinde, YOLOV8 algoritması kullanılarak zeytin ağaçlarındaki zararlı ve zararsız böceklerin tespiti ve takibi amaçlanmıştır. Veri seti hazırlama aşamasında, Roboflow platformu kullanılarak toplamda 2600 adet resim toplanmış ve etiketlenmiştir. Pamuklu bit, uğur böceği ve arılar gibi böcek türlerini tespit edebilmek için hem internetten veri toplandı hem de Teknofest takımıyla birlikte çiftliğe gidilerek resim ve video çekimleri gerçekleştirildi. Çekilen videolar, Python OpenCV kütüphanesi ile karelere (frame) ayrılarak veri setine eklendi. Ayrıca, zeytin güvesi resimleri Roboflow'daki veri setlerinden projeye dahil edildi. Eğitim aşamasında YOLOV8n kullanılarak 50 epochs ayarlanıp model eğitimi deneme amacıyla yapıldı. Daha yüksek doğrulukla sonuç elde edebilmek için YOLOV8 sürümleri YOLOv8n, YOLOv8s ve YOLOv8m kullanılarak epoch sayısı 100'e çıkarılarak dorğululuğu artırıldi. Veri setine augmentasyon yapılarak resim sayısı 5906'ya yükseltilmesiyle dorğululuğun artmasını sağlandı.

Veri seti şu sınıflardan oluşmaktadır: pamuklu bit, pamuklu bit yuvası, zeytin güvesi, arı, arı topluluğu, uğur böceği ve uğur böceği topluluğu. Eğitim süreci Google Colab Premium platformu kullanılarak gerçekleştirilmiştir. Eğitimde, Ultralytics YOLOv8 kütüphanesi, görüntülerin işlenmesi ve gösterilmesi için Display ve Image kütüphaneleri, yapılandırma dosyalarının yüklenmesi için yaml kütüphanesi, dosya ve dizin işlemleri için os ve glob kütüphaneleri kullanılmıştır. Bu kütüphaneler yardımıyla, veri seti yüklenmiş, veriler işlenmiş ve modelin eğitimi optimize edilmiştir. Ayrıca, eğitim sürecinde öğrenme oranı ve batch size gibi hiperparametreler üzerinde ayarlamalar yapılmıştır. Bu çalışma, Teknofest'te kullanılacak drone için bir eğitim modülü oluşturmayı amaçlamaktadır. Bu modül, drone vasıtasıyla zeytin ağaçlarındaki zararlı (pamuklu bit, zeytin güvesi) ve zararsız (uğur böceği, arılar) böcekleri algılayarak, zararlı olanlara püskürtme sistemi ile ilaçlama yapılmasını amaçlamaktadır.

Sonuç olarak, zararlı ve zararsız böcekleri tanıyan etkili bir model geliştirilmiştir. YOLOv8 algoritması, zeytin ağaçlarındaki böcek tespitinde yüksek doğruluk oranı sağlamıştır. Veri artırma (augmentasyon) teknikleri kullanılarak, YOLOv8 sürümlerinden faydalanarak test edilidi. Modelin performansı artırılmıştır. Eğitim sürecinde, 150 epoch ile daha yüksek doğruluk oranına ulaşılmıştır. Drone ile gerçek zamanlı böcek tespiti ve ilaçlama yapılabilir hale getirilmiştir. Bu yenilikçi çözüm, tarımda böcek yönetimi süreçlerini otomatikleştirerek, zaman ve maliyet tasarrufu sağlayacaktır. Geliştirilen model, tarım sektöründe yenilikçi uygulamalara kapı aralayarak, üreticilere büyük katkılar sunmaktadır. Bu çalışma, zeytin üretiminin sürdürülebilirliği ve verimliliği açısından da büyük önem taşımaktadır.

Anahtar kelimeler: YOLOv8, zeytin ağaçları, zararlı böcekler, zararsız böcekler, drone, tarım teknolojisi.

SUMMARY

In this thesis project, the YOLOV8 algorithm is employed for the detection and monitoring of harmful and beneficial insects in olive trees. During the data preparation phase, a total of 2400 images were collected and labeled using the Roboflow platform. To detect insect species such as cotton aphids, ladybugs, and bees, data were gathered from the internet and field visits to farms were conducted in collaboration with the Teknofest team. The captured videos were processed into frames using the Python OpenCV library and added to the dataset. Additionally, images of the olive fruit fly were included in the project from Roboflow datasets. The initial training phase involved training the model with YOLOV8n for 50 epochs. To achieve higher accuracy, the number of epochs was increased to 100 using YOLOV8m, and data augmentation techniques were applied to increase the number of images to 5906.

The dataset consists of the following classes: cotton aphid, cotton aphid nest, olive fruit fly, bee, bee swarm, ladybug, and ladybug swarm. The training process was conducted using the Google Colab Premium platform. The training utilized the Ultralytics YOLOv8 library, Display and Image libraries for image processing and visualization, the yaml library for loading configuration files, and the os and glob libraries for file and directory operations. With the assistance of these libraries, the dataset was loaded, processed, and the model was optimized. Furthermore, adjustments were made to hyperparameters such as learning rate and batch size during the training process. This study aims to create an educational module for a drone to be used at Teknofest. This module aims to detect harmful (such as cotton aphids and olive fruit flies) and beneficial insects (such as ladybugs and bees) in olive trees using a drone, and to enable targeted spraying for pest control.

In conclusion, an effective model capable of identifying harmful and beneficial insects has been developed. The YOLOV8 algorithm has achieved high accuracy in detecting insects in olive trees. Data augmentation techniques have increased the number of images in the dataset to 5906, thereby enhancing the model's performance. With 100 epochs, a higher accuracy rate has been attained during the training process. Real-time

insect detection and spraying can be achieved with a drone. This innovative solution automates insect management processes in agriculture, resulting in time and cost savings. The developed model opens doors to innovative applications in the agricultural sector and contributes significantly to producers. This study is of great importance for the sustainability and efficiency of olive production.

Keywords: YOLOV8, insect detection, olive trees, drone technology, agricultural automation, data augmentation.

1. GİRİŞ

Tarım, Türkiye ekonomisinin önemli bir bileşeni olup, ülkenin çeşitli bölgelerinde geniş bir yelpazede tarım ürünleri yetiştirilmektedir. Bu ürünler arasında zeytin, özellikle Akdeniz, Ege ve Marmara bölgelerinde yaygın olarak yetiştirilmekte olup, Türkiye'nin en önemli tarım ürünlerinden biridir. Zeytin, sadece ekonomik değil, aynı zamanda kültürel ve sosyal öneme sahiptir. Ancak, zeytin üretiminde, zararlı böceklerin ve hastalıkların olumsuz etkileri sık sık karşılaşılan bir sorundur. Bunlar, zeytin ağaçlarının verimini azaltabilir ve ürün kalitesini düşürebilir. Bu nedenle, zararlı böceklerin erken tespiti ve izlenmesi, zeytin yetiştiriciliğinde önemli bir strateji olarak kabul edilir. Betirme tezim, zeytin ağaçlarında bulunan zararlı ve zararsız böcekleri tespit etmek ve takip etmek için bir teknoloji geliştirmektir Yöntem olarak, görüntü işleme tekniklerinden faydalanarak YOLOv8 algoritması kullanılarak bir modül geliştirilecek ve bu modül drone sistemine entegre edilecektir. Zeytin üreticilerine zararlı böceklerle etkili bir şekilde mücadele etmeleri için yeni bir yapay zekaya dayalı bir sistem geliştirilmeyi hedefler. Çalışmanın odak noktası, zeytin ağaçlarında bulunan böceklerin tespit edilmesi ve izlenmesinde kullanılacak olan YOLOV8 algoritması ve farklı sürümlerinin test edilmesidir. Bu çalışmada, özellikle YOLOV8s ve YOLOV8n sürümleri incelenecek ve karşılaştırılacaktır. Her bir sürüm, zeytin ağaçlarında bulunan zararlı ve zararsız böceklerin tespitindeki performansı açısından değerlendirilecektir. Bu değerlendirme, doğruluk, hassasiyet ve hız gibi metrikler üzerinden yapılacak ve elde edilen sonuçlar analiz edilerek farklı sürümlerin avantajları ve dezavantajları belirlenecektir.

1.1 Tezin Amaçları

Türkiye genelinde zeytin ağaçlarında bulunan zararlı ve zararsız böceklerin tespit edilmesi ve izlenmesi için modern görüntü işleme teknikleri ve yapay zeka algoritmalarını kullanarak bir sistem geliştirmektir. Bu ana amacı desteklemek için belirlenen spesifik amaçlar şunlardır:

1.1.1 Zararlı ve zararsız böceklerin tespiti:

Zeytin ağaçlarında bulunan zararlı ve zararsız böcek türlerini tespit etmek ve sınıflandırmak için YOLOv8 algoritmasının çeşitli sürümleriniden YOLOv8s ve YOLOv8n kullanılması amaçlanmıştır.

1.1.2 Algoritmaların performans değerlendirmesi:

YOLOv8s ve YOLOv8n algoritmalarının böcek tespitindeki performanslarını doğruluk, hassasiyet ve hız gibi metrikler üzerinden değerlendirmesi amaçlanmaktadır. Algoritmaların farklı sürümlerinin avantajlarını ve dezavantajlarını belirlemek ve bu sürümler arasındaki performans farklarını analiz edilmesini planlanmıştır.

Bu amaçlar doğrultusunda, çalışma hem zeytin üreticilerine pratik bir çözüm sunmayı hedeflerken, hem de akademik literatüre katkıda bulunmayı amaçlamaktadır.

1.1.3 YOLOv8 alogoritma versiyonların karşılaştırılması

YOLOv8 algoritmasının farklı versiyonlarının (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) performansını karşılaştırılmasını planlanmıştır. Karşılaştırmayı gerçekleştirmek için her bir versiyonu aynı test bir görüntü üzerinde değerlendirildi. Testlerde kullanılan parametreler şunlardı: (Model, katman Sayısı, 2 parametre Sayısı, gradyanlar ve GFLOPs (Milyarlarca Float Ops). Elde edilen sonuçlar, bu çalışmada detaylı olarak etikrtlenmiş görüntü ve tahmin sonucunun detaylarını aşağıda yer almaktadır.

1.2 Literatür Araştırması

Zeytin yetiştiriciliği, Türkiye ekonomisinde önemli bir yere sahiptir ve bu alanda yapılan araştırmalar, verimliliği artırmak ve ürün kalitesini korumak amacıyla çeşitli zararlı böceklerle mücadele yöntemlerine odaklanmıştır. Bu bağlamda, modern

görüntü işleme teknikleri ve yapay zeka algoritmalarının kullanımı giderek önem kazanmaktadır. Bu literatür araştırması, zeytin zararlılarının tespiti konusundaki mevcut çalışmaları, kullanılan yöntemleri ve bu alandaki teknolojik gelişmeleri incelemektedir.

1.2.1 Zeytin yetiştiriciliğinde zararlı böcekler

Zeytin ağaçlarında görülen zararlı böcekler, ürün verimliliğini ve kalitesini olumsuz yönde etkileyen önemli faktörlerdir. Örneğin, zeytin güvesi (Prays oleae) ve pamuklu bit (Euphyllura olivina) gibi zararlılar, zeytin tanelerine ve yapraklarına zarar vererek ciddi ekonomik kayıplara neden olabilir. Bu zararlıların etkin bir şekilde kontrol edilmesi, zeytin üreticileri için büyük önem taşımaktadır.

1.2.2 Geleneksel zararlı böcek mücadele yöntemleri

Geleneksel olarak, zeytin zararlılarıyla mücadelede kimyasal ilaçlar, biyolojik yöntemler ve kültürel önlemler kullanılmaktadır. Kimyasal ilaçlar, hızlı ve etkili çözümler sunarken çevresel ve sağlık açısından riskler taşımaktadır. Biyolojik yöntemler, doğal düşmanların kullanımıyla zararlı böcek popülasyonlarını kontrol etmeyi amaçlar, ancak bu yöntemlerin etkisi zaman alabilir ve her durumda yeterli olmayabilir. Kültürel önlemler ise tarımsal uygulamalar ve bakım yöntemleriyle zararlıların yayılmasını engellemeyi hedefler.

1.2.3 Görüntü işleme teknikleri ve yapay zeka algoritmalarının kullanımı

Görüntü işleme teknikleri ve yapay zeka algoritmaları, tarımda zararlı böceklerin tespiti ve izlenmesi konusunda yeni ve etkili çözümler sunmaktadır. Son yıllarda, özellikle YOLO (You Only Look Once) algoritması, nesne tespiti alanında popülerlik kazanmıştır. YOLO algoritması, gerçek zamanlı nesne tespiti yapabilme kapasitesiyle tarımsal uygulamalarda geniş bir kullanım alanı bulmuştur.

1.2.4 Zeytin zararlılarının tespiti ve izlenmesinde yapılan çalışmalar

Literatürde, zeytin zararlılarının tespiti ve izlenmesi konusunda çeşitli çalışmalar mevcuttur. Bu çalışmalar, genellikle görüntü işleme ve yapay zeka algoritmalarının entegrasyonunu içermekte ve farklı yöntemlerin etkinliğini değerlendirmektedir. Örneğin, bir çalışma, YOLO algoritmasının zeytin sineği tespiti üzerindeki başarısını inceleyerek yüksek doğruluk oranları elde etmiştir. Başka bir araştırma, drone tabanlı

görüntüleme sistemleri kullanarak zeytin bahçelerinde zararlı böcek izleme çalışmalarının etkinliğini ortaya koymuştur.

1.3 Hipotez

Bu çalışma, modern görüntü işleme teknikleri ve yapay zeka algoritmalarını kullanarak zeytin ağaçlarında bulunan zararlı ve zararsız böceklerin tespit edilmesi ve izlenmesine yönelik bir yöntem geliştirmeyi amaçlamaktadır. Bu bağlamda, çalışma kapsamında test edilecek hipotezler şunlardır:

1.3.1 Ana hipotez

1.3.1.1 YOLOv8 algoritmasının böcek zararlılarının tespitinde etkinliği:

- H1: YOLOv8 algoritması, zeytin ağaçlarında bulunan zararlı ve zararsız böceklerin tespitinde yüksek doğruluk, hassasiyet ve hız performansı gösterir.

1.3.2 Alt hipotezler

1.3.2.1 Farklı YOLOv8 sürümlerinin performans farklılıkları:

- H2a: YOLOv8x sürümü, zeytin ağaçlarındaki böceklerin tespitinde en yüksek doğruluk oranını sağlar.
- H2b: YOLOv8m sürümü, zeytin ağaçlarındaki böceklerin tespitinde dengeli bir performans (hız ve doğruluk) sunar.
- H2c: YOLOv8n sürümü, zeytin ağaçlarındaki böceklerin tespitinde en yüksek hız performansını sağlar.

1.3.2.2 Zararlı ve zararsız böceklerin ayrımında algoritmanın etkinliği

- H3: YOLOv8 algoritması, zeytin ağaçlarındaki zararlı böcekler ile zararsız böcekler arasında yüksek doğrulukla ayrım yapabilir.
- Erken tespit ve izleme sisteminin pratik uygulanabilirliği:
- H4: Geliştirilen görüntü işleme ve yapay zeka tabanlı erken tespit ve izleme sistemi, sahada uygulanabilir ve zeytin üreticileri tarafından etkin bir şekilde kullanılabilir.

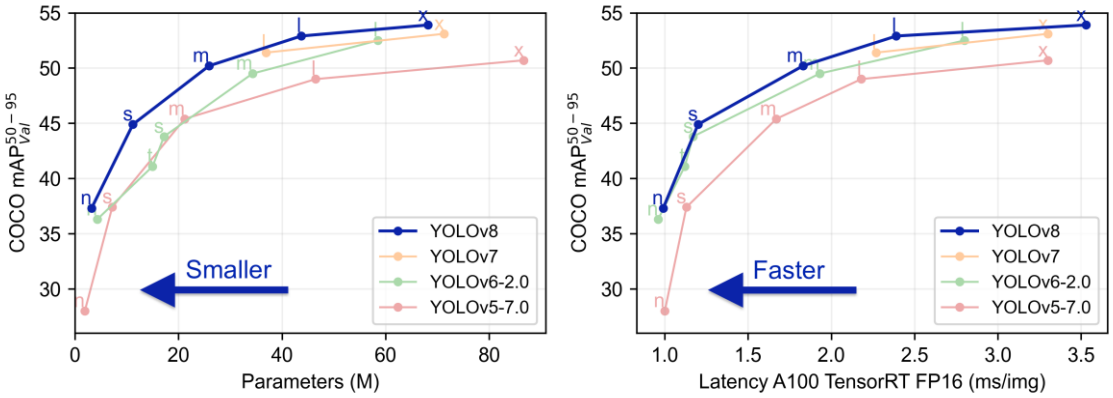
1.3.3 Hipotezlerin test edilmesi

Bu hipotezler, YOLOv8 algoritmasının farklı sürümlerinin (YOLOv8s, YOLOv8n) zeytin ağaçlarındaki zararlı ve zararsız böceklerin tespitindeki performansını doğruluk, hassasiyet ve hız gibi metrikler üzerinden değerlendirerek test edilecektir. Ayrıca, sistemin sahadaki uygulanabilirliği ve pratikliği kullanıcı geri bildirimleri ve saha testleri aracılığıyla değerlendirilecektir.

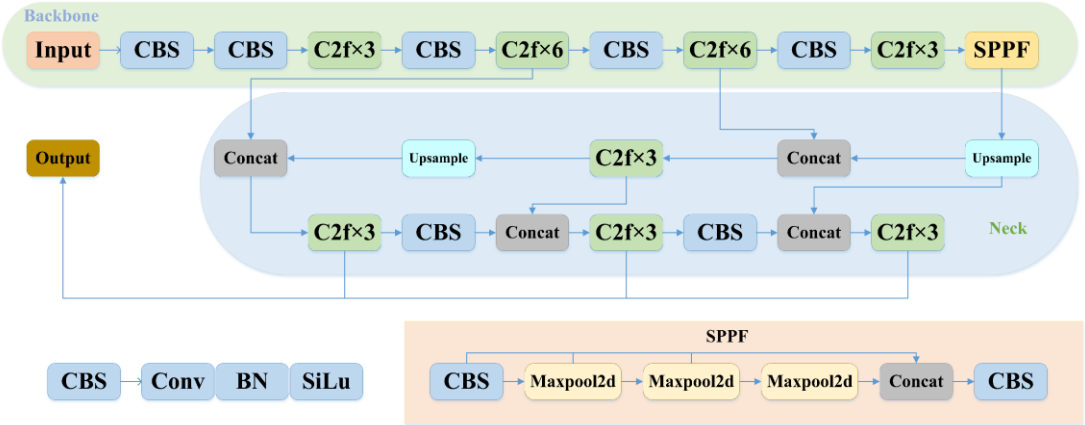
2. YÖNTEM

YOLOv8, derin öğrenmeye, özellikle Konvolüsyonel Sinir Ağlarına (CNN'ler) dayalı açık kaynaklı bir nesne algılama algoritmasıdır. Nesneleri gerçek zamanlı olarak algılamadaki hızları ve doğrulukları ile bilinen YOLO (You Only Look Only Once) nesne algılama algoritmaları ailesinin bir parçasıdır. Önceki YOLO sürümlerinden yararlanan YOLOv8 modeli, performans için eğitim modelleri için birleşik bir çerçeve sağlarken daha hızlı ve daha doğrudur . YOLOv8, hem hız hem de doğruluk açısından diğer birçok nesne algılama algoritmasından daha iyi performans gösteren son teknoloji bir nesne algılama algoritmasıdır. Derin öğrenme ve bilgisayar görüşüne yeni teknikler dahil ederek YOLOv3 gibi önceki YOLO sürümlerine göre bir gelişmedir. (Sakin, 2023)

2.1 YOLOv8 Yapısı



Şekil 2.2.1: YOLOv8 Başarı Grafiği



Şekil 2.2.2: YOLOv8'in Ağ Yapısı

Şekil 2.2.1'de YOLOv8 başarı grafiği yer almaktadır. YOLOv8 algoritmasını önceki çıkan YOLO algoritmalar ile karşılaştırılmıştır ve mAP50-90 ve paramtre vererek YOLOv8 algoritmasının performansı ve dorğluğu en iyi olduğunu gösteren bir graftir. YOLOv8'in önceki YOLO sürümlerine göre daha gelişmiş ve etkili bir nesne tespit algoritması olduğunu gösterir. Bu durum, gerçek dünya uygulamalarında daha güvenilir ve hassas sonuçlar elde etmek için YOLOv8'in tercih edilmesini destekler.

Şekil 2.2.2'de YOLOv8'in ağ yapısı, önceki YOLO sürümlerine dayanırken, yeni özellikler ve iyileştirmelerle güçlendirilmiştir. Bu model, hızlı, doğru ve kullanımı kolaydır. Nesne algılama, izleme, segmentasyon ve sınıflandırma gibi görevler için mükemmel bir seçenek sunar. YOLOv8'in optimize edilmiş mimarisi ve gelişmiş eğitim teknikleri, yüksek performans ve esneklik sağlar. Bu, gerçek zamanlı uygulamalarda etkili sonuçlar elde etmek için ideal bir çözüm haline getirir.

2.2 YOLOv8 Modeli için Metriklerin Hesaplanması:

2.2.1 Sınıf bazlı metrikler

Çıktının bir bölümü, performans metriklerinin sınıf bazında detaylandırılmasıdır. Bu ayrıntılı bilgi, özellikle çeşitli nesne kategorilerine sahip veri setlerinde modelin her bir sınıf için ne kadar iyi performans gösterdiğini anlamaya çalışırken faydalıdır. Veri setindeki her sınıf için şu bilgiler sağlanır:

Sınıf: Nesne sınıfının adını belirtir, örneğin "insan", "araba" veya "köpek".

Görüntüler: Bu metrik, doğrulama setinde nesne sınıfını içeren görüntü sayısını gösterir.

Örnekler: Sınıfın doğrulama setindeki tüm görüntülerde kaç kez göründüğünü belirtir.

Kutu (P, R, mAP50, mAP50-95):

P (Precision - Hassasiyet): Tespit edilen nesnelerin doğruluğu, kaç tespitin doğru olduğunu gösterir.

R (Recall - Duyarlılık): Modelin görüntülerdeki tüm nesne örneklerini tanımlama yeteneği.

mAP50: Intersection over union (IoU) eşik değeri 0.50'de hesaplanan ortalama hassasiyet. Modelin "kolay" tespitleri dikkate alan doğruluğunu ölçer.

mAP50-95: Değişen IoU eşik değerlerinde (0.50'den 0.95'e kadar) hesaplanan ortalama hassasiyetlerin ortalaması. Modelin farklı zorluk seviyelerindeki tespit performansının kapsamlı bir görünümünü sunar.

2.2.2 Hız metrikleri

Özellikle gerçek zamanlı nesne tespiti senaryolarında, çıkarım hızının doğruluk kadar kritik olabileceği bir gerçektir. Bu bölüm, doğrulama sürecinin çeşitli aşamaları için geçen süreyi, ön işleme aşamasından son işlem aşamasına kadar ayırır.

2.2.3 COCO metrik değerlendirmesi

COCO veri setinde doğrulama yapan kullanıcılar için, COCO değerlendirme scripti kullanılarak ek metrikler hesaplanır. Bu metrikler, farklı IoU eşik değerlerinde ve farklı boyutlardaki nesneler için hassasiyet ve duyarlılık hakkında bilgi sağlar.

2.2.4 Görsel çıktılar

model.val() fonksiyonu, sayısal metriklerin yanı sıra modelin performansını daha sezgisel bir şekilde anlamaya yardımcı olan görsel çıktılar da üretir. İşte bekleyebileceğiniz görsel çıktılar:

- **F1 Skoru eğrisi (F1_curve.png):** Bu eğri, çeşitli eşik değerlerinde F1 skorunu temsil eder. Bu eğriyi yorumlamak, modelin farklı eşik değerlerinde yanlış pozitifler ve yanlış negatifler arasındaki dengesini anlamanızı sağlayabilir.

- **Hassasiyet-Duyarlılık eğrisi (PR_curve.png):** Herhangi bir sınıflandırma problemi için önemli bir görselleştirme olan bu eğri, farklı eşik değerlerinde hassasiyet ve duyarlılık arasındaki değiş tokuşu gösterir. Özellikle dengesiz sınıflarla uğraşırken önemli hale gelir.

- **Hassasiyet eğrisi (P_curve.png):** Farklı eşik değerlerinde hassasiyet değerlerinin grafiksel bir temsilidir. Bu eğri, eşik değiştikçe hassasiyetin nasıl değiştiğini anlamanıza yardımcı olur.

- **Duyarlılık eğrisi (R_curve.png):** Benzer şekilde, bu grafik farklı eşik değerlerinde duyarlılık değerlerinin nasıl değiştiğini gösterir.

- **Karışıklık matrisi (confusion_matrix.png):** Karışıklık matrisi, her sınıf için doğru pozitifler, doğru negatifler, yanlış pozitifler ve yanlış negatiflerin sayısını gösteren ayrıntılı bir görünüm sunar.

- **Normalize edilmiş karışıklık matrisi (confusion_matrix_normalized .png):** Bu görselleştirme, karışıklık matrisinin normalize edilmiş bir versiyonudur. Verileri ham sayılar yerine oranlarla temsil eder. Bu format, sınıflar arasındaki performansı karşılaştırmayı kolaylaştırır.

- **Doğrulama parti etiketleri (val_batchX_labels.jpg):** Bu görüntüler, doğrulama veri setinden alınan belirli partiler için gerçeğe uygun etiketleri gösterir. Nesnelerin ne olduğunu ve veri setine göre konumlarını net bir şekilde gösterir.

- **Doğrulama parti tahminleri (val_batchX_pred.jpg):** Etiket görüntülerinin aksine, bu görseller, YOLOv8 modelinin ilgili partiler için yaptığı tahminleri gösterir. Bu görüntüleri etiket görüntüleriyle karşılaştırarak, modelin nesneleri ne kadar iyi tespit edip sınıflandırdığını görsel olarak değerlendirebilirsiniz.

2.2.5 Sonuçların Saklanması

Gelecekte referans almak için sonuçlar genellikle runs/detect/val adlı bir dizine kaydedilir.

2.2.5.1 Doğru Metrikleri Seçmek

Doğru metrikleri değerlendirmek, genellikle özel uygulamaya bağlıdır:

- **mAP:** Model performansının geniş bir değerlendirmesi için uygundur.
- **IoU:** Nesne konumunun kesin olduğu durumlarda gereklidir.
- **Hassasiyet:** Yanlış tespitleri en aza indirmek önemli olduğunda.
- **Duyarlılık:** Bir nesnenin her örneğini tespit etmek önemli olduğunda.
- **F1 Skoru:** Hassasiyet ve duyarlılık arasında bir denge gerektiğinde.

Gerçek zamanlı uygulamalar için, FPS (Saniye Başına Kare) ve gecikme gibi hız metrikleri, zamanında sonuçların sağlanması için çok önemlidir.

2.3 Veri Seti

Kullanılan veri seti, sahada yapılan çekimlerden ve internetten toplanan videoların işlenmesiyle oluşturulmuştur. Veri seti, zeytin ağaçlarındaki zararlı (pamuklubiti ve zeytin Güvesi) ve zararsız (uğur böceği ve arılar) böcek türlerini içermekte ve aşağıdaki yöntemlerle elde edilmiştir:

3.1.1 Saha çekimleri

Teknofest'teki takım arkadaşlarımla birlikte zeytin bahçelerine giderek saha çekimleri gerçekleştirdik. Bu çekimlerdeki ana hedefimiz, pamuklu bitlerin yuvalarını ve zeytin ağaçlarını görüntülemektir. Saha çekimleri, zeytin ağaçlarının doğal ortamlarında çekilen yüksek çözünürlüklü fotoğraflar ve videolar kullanılarak gerçekleştirilmiştir. Bu görüntüler, hem arka plan olarak kullanılacak hem de pamuklu bitlerin tanımlanmasında veri sağlayacaktır. Pamuklu bitlerin internette yeterli veri bulunmaması nedeniyle bu çekimler özellikle önem taşımaktadır.

2.3.1 İnternetten Toplanan Videolar:

Arı, uğur böceği ve zeytin güvesi gibi diğer böcek türlerine ait görüntüler, internette bulunan videoların işlenmesiyle elde edilmiştir. OpenCV modülü kullanılarak bu videolar karelere (frames) ayrılmış ve uygun olan kareler veri setine dahil edilmiştir. Böylece, zeytin bahçelerindeki farklı böcek türlerinin daha geniş bir yelpazesi veri setine eklenmiştir.

3.1.3 Roboflow Kullanımı ve Veri Artırma (Augmentation):

Toplanan tüm görüntüler ve videolar, Roboflow platformu kullanılarak etiketlenmiş ve organize edilmiştir. Roboflow, görüntü işleme ve etiketleme süreçlerini kolaylaştırarak veri setinin daha etkili bir şekilde oluşturulmasını sağlamıştır. Platform, veri setinin farklı sınıflara ayrılmasını ve her bir görüntünün uygun şekilde etiketlenmesini mümkün kılmıştır. Ayrıca, veri setinin daha çeşitli ve dengeli hale getirilmesi için veri artırma (augmentation) teknikleri uygulanmıştır. Bu teknikler arasında görüntülerin döndürülmesi, kesilmesi, parlaklık ve kontrast ayarlarının değiştirilmesi gibi işlemler yer almaktadır. Bu sayede, modelin farklı koşullarda daha iyi performans göstermesi sağlanmıştır.Bu yöntemler sayesinde, zeytin ağaçlarındaki zararlı ve zararsız böceklerin tespiti için zengin ve çeşitli bir veri seti oluşturulmuştur. Toplanan veri, YOLOv8 algoritmasının YOLOv8n ve YOLOv8s sürümlerinin eğitimi ve değerlendirilmesi için kullanılmıştır.

2.4 Colab Eğitimi

YOLOv8 modelinin eğitimi, Google Colab platformu üzerinde gerçekleştirilmiştir. Bu platform, güçlü donanım kaynaklarına erişim sağlayarak modelin hızlı ve etkili bir şekilde eğitilmesine olanak tanımıştır. Eğitim süreci aşağıdaki adımları içermiştir:

2.4.1 Veri Setinin Yüklenmesi ve Düzenlenmesi

Öncelikle, Roboflow aracılığıyla oluşturulan veri seti Google Drive'a yüklenmiş ve Google Colab ortamına aktarılmıştır. Veri seti, eğitim için uygun formata dönüştürülmüş ve eğitim için hazır hale getirilmiştir.

2.4.2 Modelin Eğitimi

Eğitim süreci, Ultralytics YOLOv8 modelinin Colab üzerindeki Python betiği kullanılarak gerçekleştirilmiştir. Bu betik, modelin belirlenen sayıda epoch boyunca eğitilmesini ve ağırlıkların güncellenmesini sağlamıştır. Eğitim süreci boyunca, modelin performansı gözlemlenmiş ve gerekli iyileştirmeler yapılmıştır.

2.4.3 Sonuçların Analizi

Eğitim tamamlandıktan sonra, modelin performansı çeşitli metrikler kullanılarak değerlendirilmiştir. Bu metrikler arasında doğruluk, hassasiyet ve ortalama kesinlik

(mAP) gibi ölçümler bulunmaktadır. Modelin başarısı, elde edilen sonuçlar üzerinden değerlendirilmiş ve iyileştirmeler için gereken adımlar belirlenmiştir.

2.4.4 Görselleştirme ve Sonuçlar:

Eğitim sürecinde elde edilen sonuçlar görselleştirilmiş ve analiz edilmiştir. Bu aşamada, eğitim sırasında oluşturulan grafikler, karşılaştırmalı sonuçlar ve hata matrisleri kullanılarak modelin performansı detaylı bir şekilde incelenmiştir.

Colab üzerinde gerçekleştirilen eğitim süreci, YOLOv8 modelinin başarıyla eğitilmesini sağlamış ve zeytin bahçelerindeki zararlı böceklerin tespiti için güçlü bir araç oluşturmuştur. Bu yöntem, hem verimlilik hem de kullanılabilirlik açısından avantajlar sağlayarak hedeflerine ulaşmasını sağlamıştır.

3. SONUÇLARIN KARŞILAŞTIRILMASI

3.1 YOLOv8 Modellerin Hesaplamaları

Çizelge 3.2.1'deki veriler, YOLOv8 modelinin farklı boyutlardaki varyasyonlarına (n, s, m, l, x) ve bu varyasyonların performansına ilişkin bilgileri içeriyor.

Model Boyutu (piksel): Giriş görüntüsünün çözünürlüğünü belirtir.

mAPval: Ortalama Hassaslık-Precizyon eğrisi altındaki alanı belirleyen ortalama doğruluk ölçümü. Yüksek bir değer, daha iyi performans anlamına gelir.

Hız (CPU ONNX): Modelin işlemek için gereken ortalama süreyi milisaniye cinsinden belirtir. Daha düşük bir değer, daha hızlı bir işlem anlamına gelir.

Hız (A100 TensorRT): Nvidia A100 Tensor Core GPU ile işlemek için gereken ortalama süreyi milisaniye cinsinden belirtir.

Parametreler (M): Modelin eğitilebilir parametre sayısını belirtir. Daha büyük bir değer, daha karmaşık bir model anlamına gelebilir.

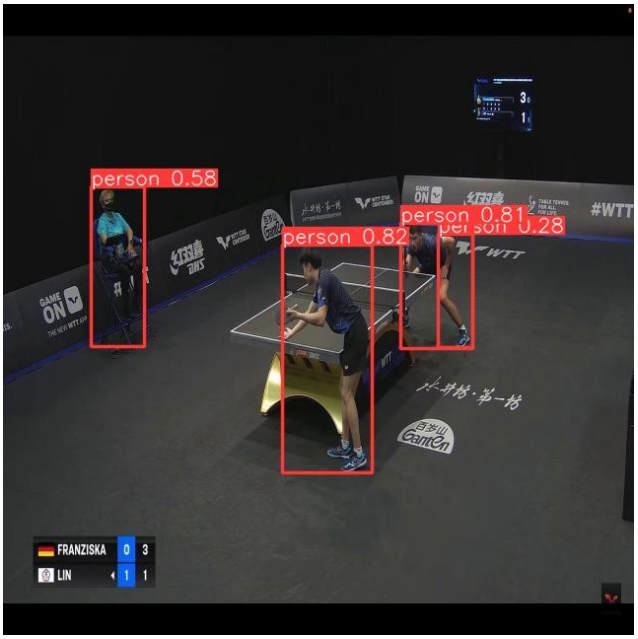
FLOPs (B): Modelin tahmini işlem sayısını belirtir ve modelin karmaşıklığını yansıtır. Daha yüksek bir değer, daha fazla işlem gerektiren bir model anlamına gelir.

Tablodan, model boyutunun arttıkça (n'den x'e), mAPval ve FLOPs değerlerinin arttığını görebiliriz. Ancak, hız (CPU ve GPU üzerinde) da büyük ölçüde artar. Bu nedenle, model seçimi yaparken, performans ve hız arasında bir denge kurmak önemlidir.

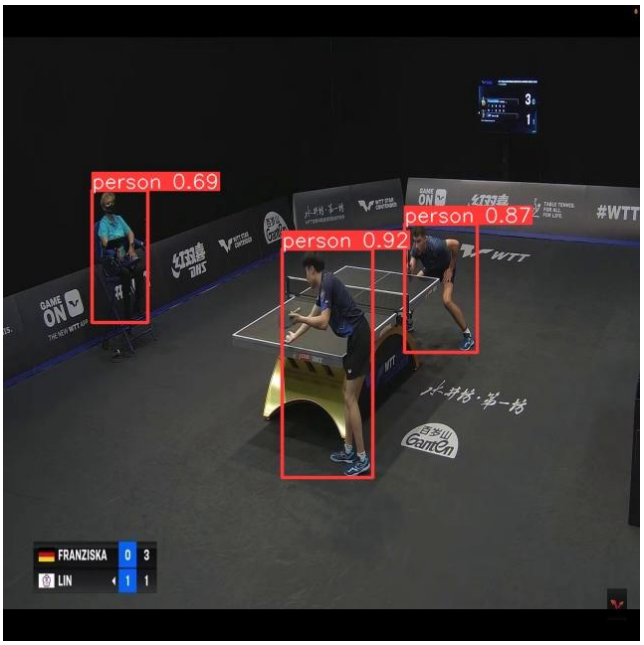
Çizelge 3.2.1

Model	Boyut (pixels)	mAP değeri (50-95)	Hız CPU ONNX (ms)	Hız A100 TensorRT (ms)	Params (M)	FLOPS (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	275.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.56	68.2	257.8

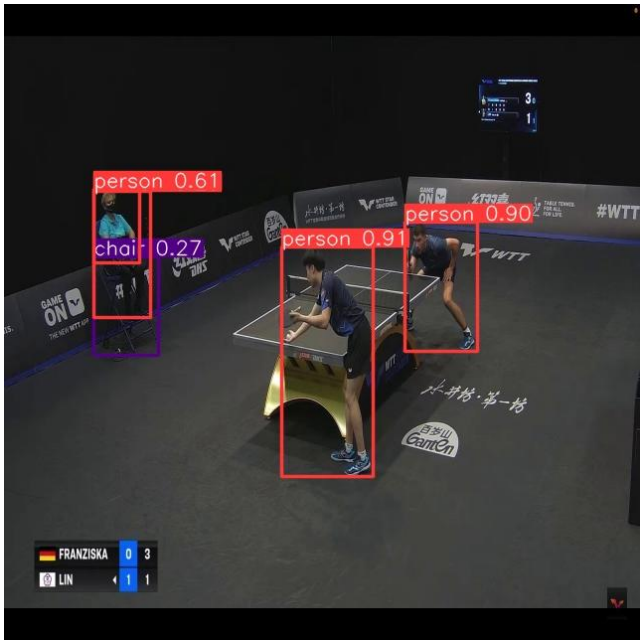
3.2 YOLOv8 Versiyonların Tahmin Sonuçları



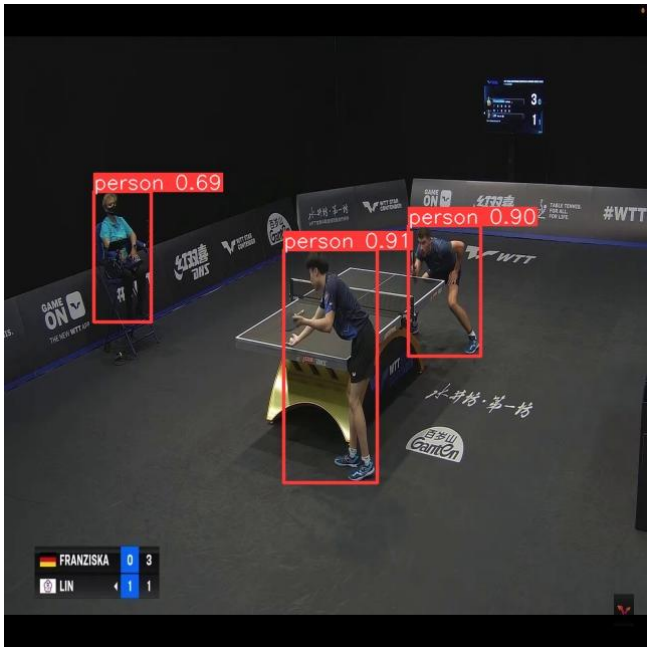
Şekil 3.3.2: YOLOv8n



Şekil 3.3.3 : YOLOv8s



Şekil 3.3.4: YOLOv8m



Şekil 3.3.5: YOLOv8l

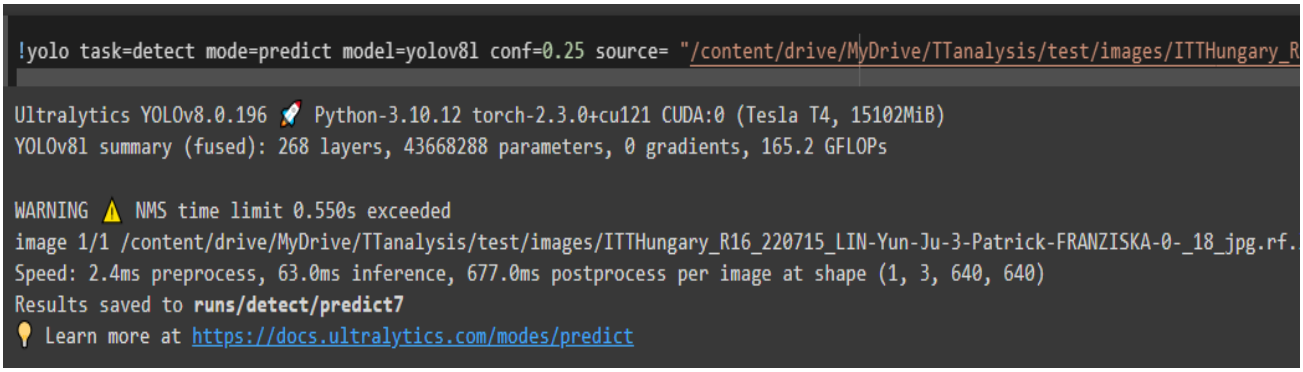
Yukarıda verilen görütüler **şekil 3.3.2'den** başlayarak **3.3.5**'e kadar YOLOv8 versiyonlarını tahmin yaparak elde edilen görüntülerdir.

Şekil 3.3.2 YOLOv8n, en hafif modeldir ve en az parametreye ve GFLOPs'a sahiptir. Daha hızlı işleme süreleri sunar, ancak daha düşük doğruluk tahmin yapılmıştır.

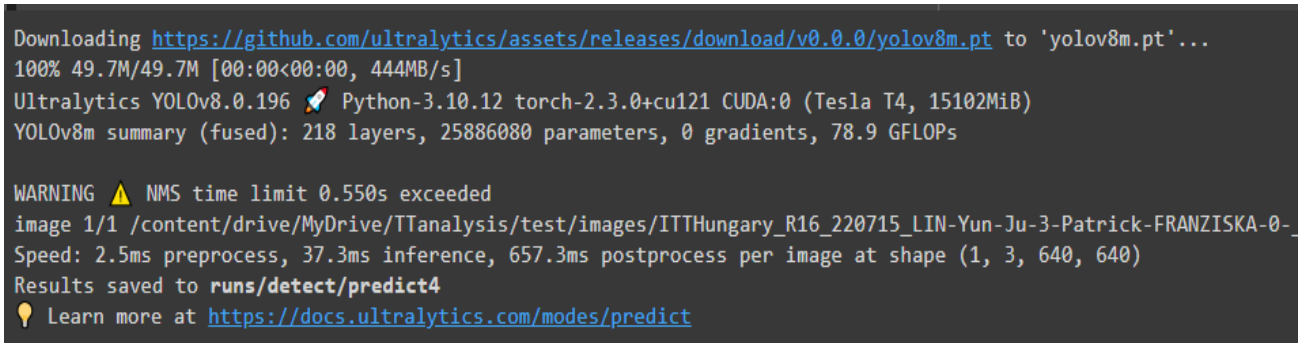
Şekil 3.3.3 YOLOv8s, daha hafif bir modeldir ve daha az parametreye ve GFLOPs'a sahiptir. Bu model, hızlı işleme süreleri sağlar, ancak daha düşük doğrulukla birlikte sonuçlandırılmıştır.

Şekil 3.3.4 YOLOv8m, parametre sayısı ve GFLOPs açısından daha azdır, ancak yine de orta seviye bir modeldir. İşleme süreleri diğerlerine kıyasla daha hızlıdır, ancak doğruluk açısından daha düşük olabilir. Yukarıdaki görüntüye bakılacaksa sandalyayı düşük oran ile olasa bile tahmin etti. Sandalyadaki hakemi 7% farkla düşük oran la tahmin ettiğini fark edilir.

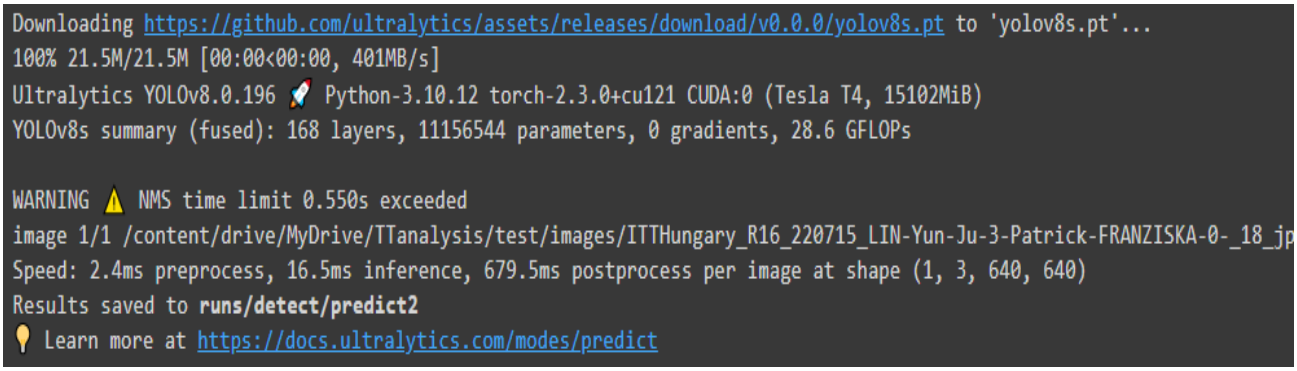
Şekil 3.3.5 YOLOv8l, bir önceki modele göre daha az parametreye ve daha düşük bir GFLOPs'a sahiptir, ancak hala oldukça karmaşık bir yapıya sahiptir. Bu model, yüksek doğruluk sağlayabilirken, işleme süreleri YOLOv8x'e göre biraz daha düşüktür.



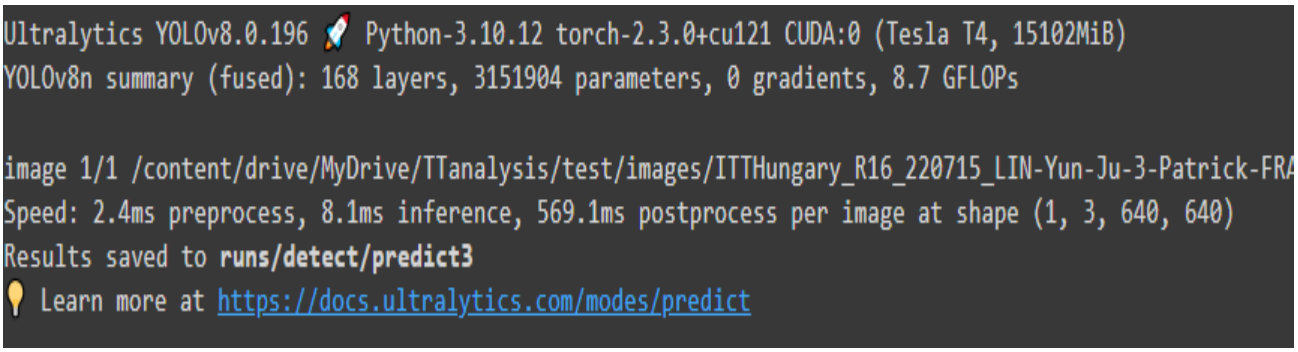
Şekil 3.3.7: YOLOv8l Tahmin Sonucu



Şekil 3.3.8: YOLOv8m Tahmin Sonucu



Şekil 3.3.10: YOLOv8s Tahmin Sonucu

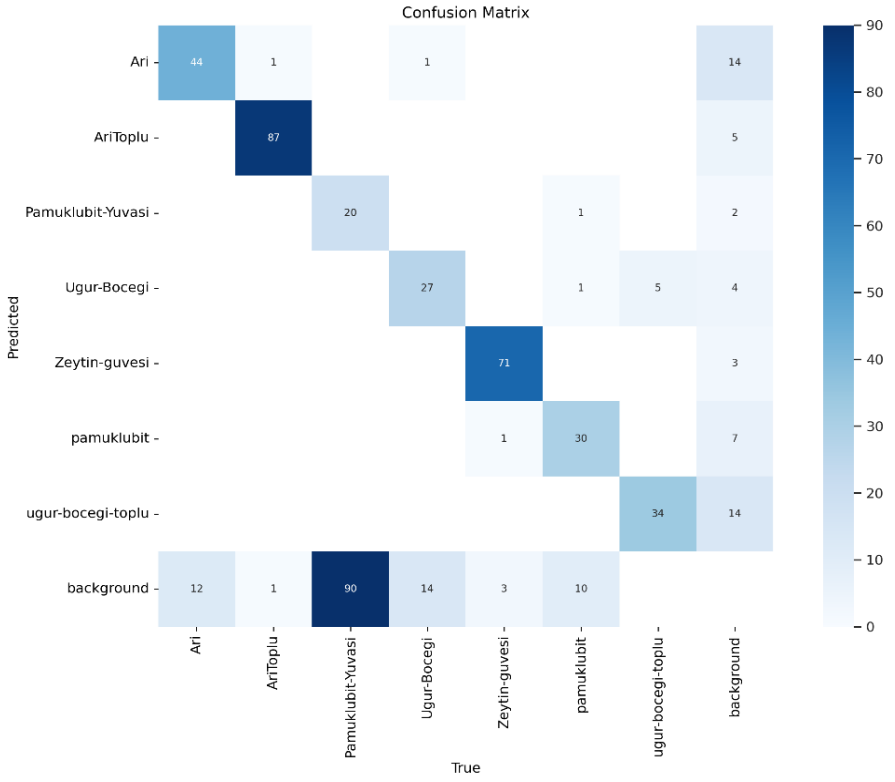


Şekil 3.3.9: YOLOv8n Tahmin Sonucu

Şekil 3.3.6'dan **şekil 3.3.10'**a kadar yukarıdaki YOLOv8n, YOLOv8s, YOLOv8m ve YOLOv8l algoritmaların sonuçlarını göstermektedir. Sonuçlarda katman sayısı, kullanılan parametre sayısı ve GFLOPs sonucu vardır. Sonuçlara bakılırsa GFLOPs genel olarak parametre sayısı artıkça artmaktadır. Dolayısıyla, YOLOv8l, en yüksek GFLOPs değerine sahip olduğundan, en karmaşık ve hesaplama yoğun

modeldir. YOLOv8n ise en düşük GFLOPs değerine sahip olduğundan, daha hafif ve daha az karmaşık bir modeldir. Bu bilgiler, model seçimi yaparken veya performansı değerlendirirken dikkate alınabilir.

3.3 Eğitilmiş Modellerin Analizi



Şekil 3.3.1: YOLOv8n Confusion Matrics

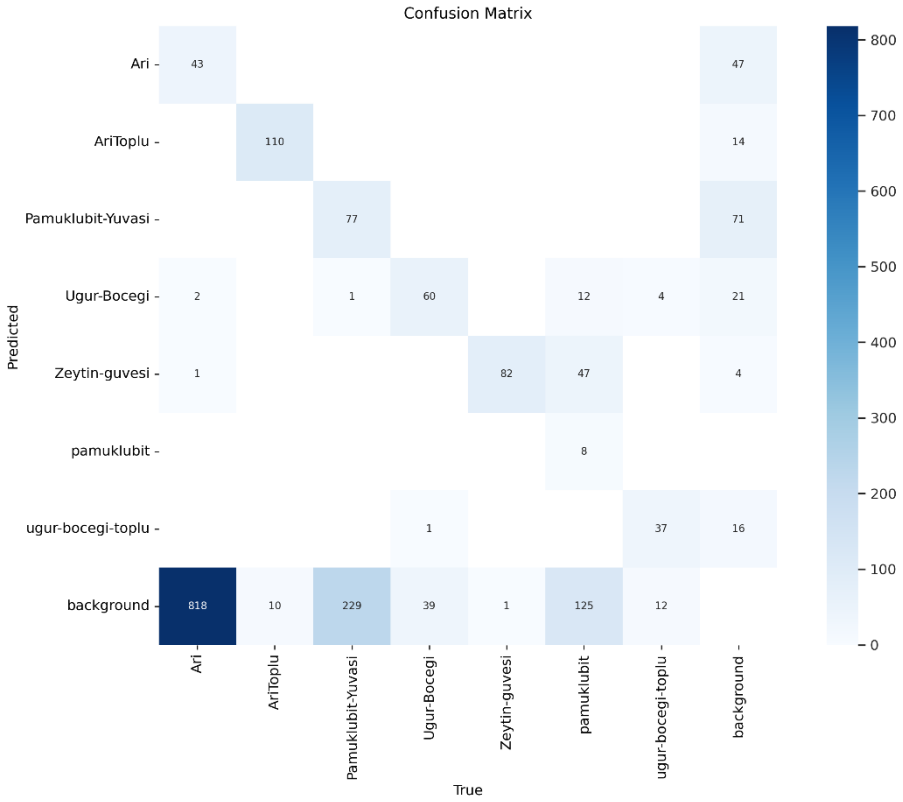
Şekil 3.3.1'de YOLOv8n ile Eğitilen modelin karışıklık matrisi (Confusion Matrics) sonucudur. Gösterildiği üzere 8 sınıftan oluşan bir model vardır. True Positive (TP) Modelin pozitif sınıfı doğru bir şekilde tahmin ettiğini gösterir. True Negative (TN) Modelin negatif sınıfı doğru bir şekilde tahmin ettiğini gösterir. False Positive (FP) Modelin pozitif sınıfı yanlış bir şekilde tahmin ettiğini gösterir. False Negative (FN) Modelin negatif sınıfı yanlış bir şekilde tahmin ettiğini gösterir.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy = 313 / 502 = 0.581

$$\textbf{Precision} = \frac{TP}{TP + FP} \quad \text{or} \quad \frac{\text{True Positive}}{\text{Predictive Results}}$$

Precision = 156.5 / (156.5+130) = 0.58



Şekil 3.3.2: YOLOv8s Confusion Matrics

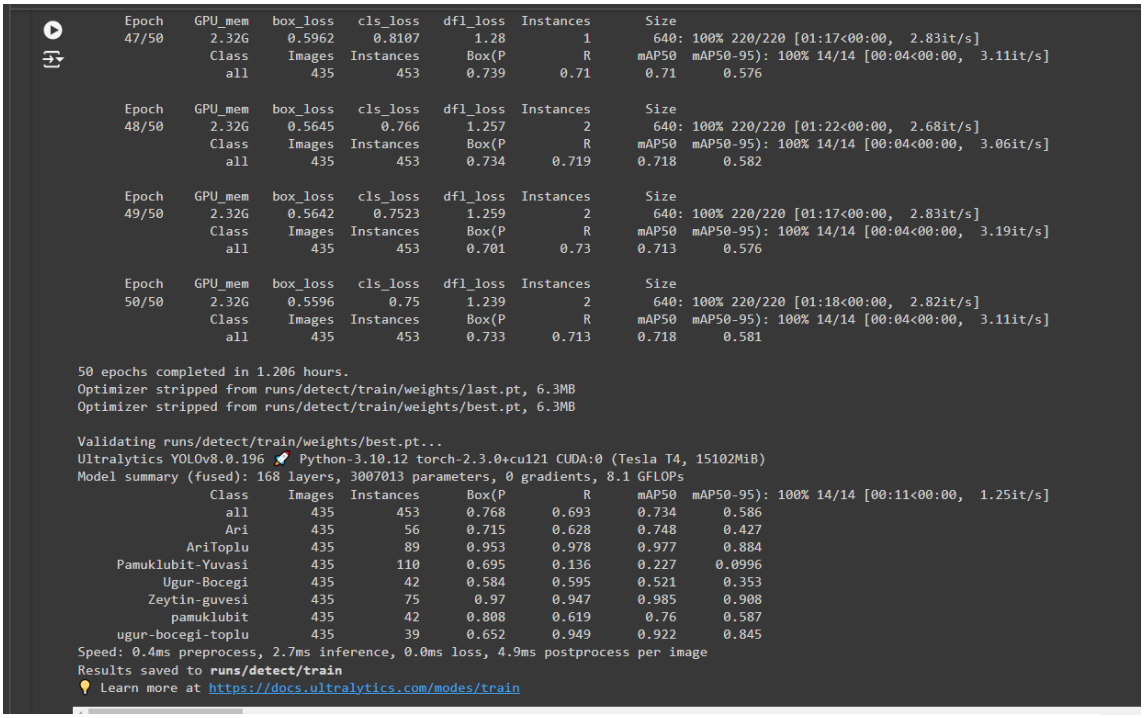
Şekil 3.3.2'de YOLOv8s ile Eğitilen modelin karışıklık matrisi (Confusion Matrics) sonucudur.

Accuracy = 0.58

Precision = 156.5 / (156.5+130) = 0.578

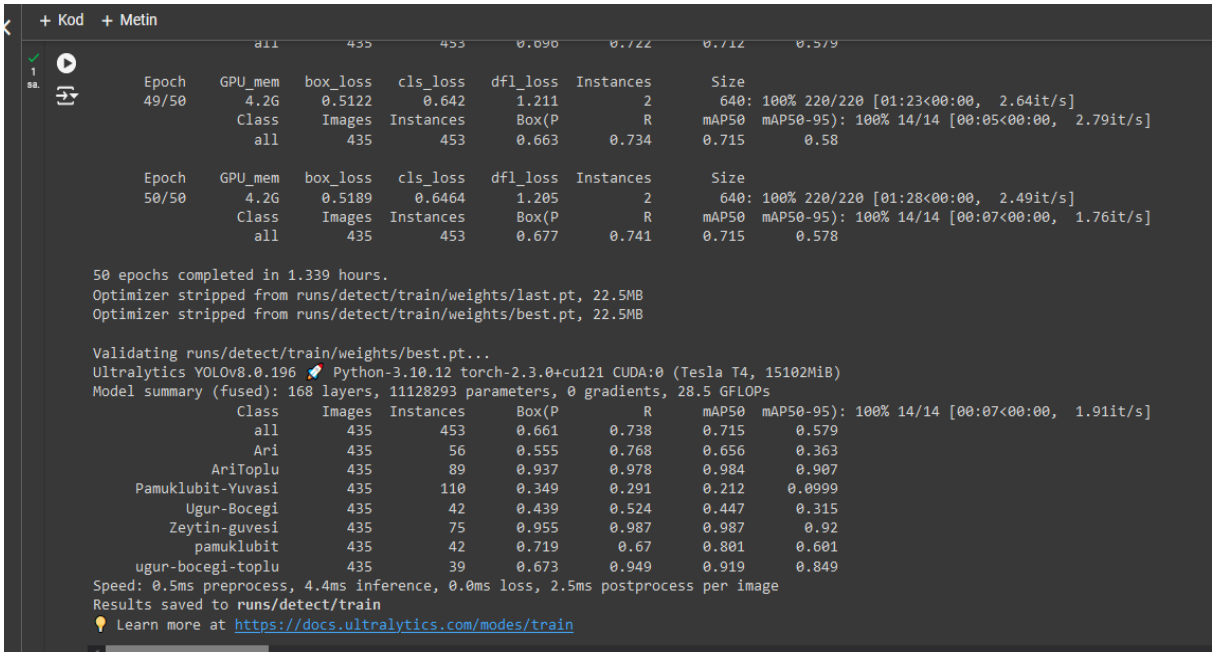
İki algoritmanın soncunda pek fark göstermediğini anlaşılabilir. YOLOv8s eğitim süreci daha fazladır. Yukarıdaki şekilde anlaşıldığı gibi daha fazla parametre kullanarak dorğu değerlerin tahmin değerlerinen daha fazla çıktığını fark edilmiştir.

Şekil 3.3.1 ve **Şekil 3.3.2** şu aşağıdaki iki resim colab platformdan eklenmiştir.



```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      47/50      2.32G     0.5962     0.8107       1.28          1        640: 100% 220/220 [01:17<00:00,  2.83it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:04<00:00,  3.11it/s]
                   all        435        453      0.739       0.71       0.71      0.576

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      48/50      2.32G     0.5645      0.766      1.257          2        640: 100% 220/220 [01:22<00:00,  2.68it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:04<00:00,  3.06it/s]
                   all        435        453      0.734      0.719      0.718      0.582

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      49/50      2.32G     0.5642     0.7523      1.259          2        640: 100% 220/220 [01:17<00:00,  2.83it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:04<00:00,  3.19it/s]
                   all        435        453      0.701       0.73      0.713      0.576

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      50/50      2.32G     0.5596       0.75      1.239          2        640: 100% 220/220 [01:18<00:00,  2.82it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:04<00:00,  3.11it/s]
                   all        435        453      0.733      0.713      0.718      0.581

50 epochs completed in 1.206 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.3MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3007013 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:11<00:00,  1.25it/s]
                   all        435        453      0.768      0.693      0.734      0.586
                   Ari        435         56      0.715      0.628      0.748      0.427
              AriToplu        435         89      0.953      0.978      0.977      0.884
     Pamuklubit-Yuvasi        435        110      0.695      0.136      0.227     0.0996
           Ugur-Bocegi        435         42      0.584      0.595      0.521      0.353
         Zeytin-guvesi        435         75       0.97      0.947      0.985      0.908
             pamuklubit        435         42      0.808      0.619       0.76      0.587
     ugur-bocegi-toplu        435         39      0.652      0.949      0.922      0.845
Speed: 0.4ms preprocess, 2.7ms inference, 0.0ms loss, 4.9ms postprocess per image
Results saved to runs/detect/train
💡 Learn more at https://docs.ultralytics.com/modes/train
```

Şekil 3.3.3: YOLOv8n'nin Colabtaki Eğitim Sonucu



```
+ Kod  + Metin
      all        435        453      0.696      0.722      0.712      0.579

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      49/50      4.2G      0.5122      0.642      1.211          2        640: 100% 220/220 [01:23<00:00,  2.64it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:05<00:00,  2.79it/s]
                   all        435        453      0.663      0.734      0.715       0.58

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      50/50      4.2G      0.5189     0.6464      1.205          2        640: 100% 220/220 [01:28<00:00,  2.49it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:07<00:00,  1.76it/s]
                   all        435        453      0.677      0.741      0.715      0.578

50 epochs completed in 1.339 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11128293 parameters, 0 gradients, 28.5 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:07<00:00,  1.91it/s]
                   all        435        453      0.661      0.738      0.715      0.579
                   Ari        435         56      0.555      0.768      0.656      0.363
              AriToplu        435         89      0.937      0.978      0.984      0.907
     Pamuklubit-Yuvasi        435        110      0.349      0.291      0.212     0.0999
           Ugur-Bocegi        435         42      0.439      0.524      0.447      0.315
         Zeytin-guvesi        435         75      0.955      0.987      0.987       0.92
             pamuklubit        435         42      0.719       0.67      0.801      0.601
     ugur-bocegi-toplu        435         39      0.673      0.949      0.919      0.849
Speed: 0.5ms preprocess, 4.4ms inference, 0.0ms loss, 2.5ms postprocess per image
Results saved to runs/detect/train
💡 Learn more at https://docs.ultralytics.com/modes/train
```

Şekil 3.3.3: YOLOv8s'in Colabtaki Eğitim Sonucu

Şekil 3.3.2'de YOLOv8n Sonucu Eğitim sonucunda elde edilen doğruluk değeri 0.58 olarak kaydedildi. Eğitim süresi 72.36 dakika olup, modelin eğitiminde 50 epoch kullanıldı. YOLOv8n sürümü ile yapılan eğitim, zeytin ağaçlarındaki zararlı ve zararsız böcek türlerini sınıflandırmada makul bir doğruluk sağlamıştır.
Yukarıdaki şekilde, her bir sınıfın doğrulukları detaylı olarak gösterilmektedir. Modelin performansı, eğitim süresi ve doğruluk açısından dengeli bir sonuç ortaya koymuştur. Doğruluğu artırmak için veri setinin genişletilmesi veya farklı augmentasyon tekniklerinin uygulanması düşünülüyor.

Şekil 3.3.3'de Eğitim sonucunda elde edilen doğruluk değeri 0.576 olarak kaydedildi. Eğitim süresi 80.34 dakika olup, modelin eğitiminde 50 epoch kullanıldı. YOLOv8s sürümü ile yapılan eğitim, zeytin ağaçlarındaki zararlı ve zararsız böcek türlerini sınıflandırmada YOLOv8n fazla sürdü ve doğrulukta değişkenlik göstermedi. Yukarıdaki şekilde, her bir sınıfın doğrulukları detaylı olarak gösterilmektedir.
Sonuç olarak YOLOv8n, YOLOv8s'e kıyasla daha hızlı bir eğitim süresi gösterdi, bu da zamanın kritik olduğu uygulamalarda daha verimli olabileceğini gösterir.
YOLOv8n, eğitim süresi ve doğruluk açısından dengeli bir performans sağladı, bu da gerçek zamanlı ve kaynak kısıtlı ortamlara uygunluğunu gösterir.YOLOv8s ise daha uzun bir eğitim süresine rağmen, elde edilen doğruluk YOLOv8n ile karşılaştırılabilir düzeyde olup, ek eğitim süresi doğrulukta önemli bir artış sağlamadı.

Şekil 3.3.4: Performans değerlendirme

Şekil 3.3.4'de modelin farklı böcek türlerini tanıma performansının çeşitlilik gösterdiğini ve bazı türlerde iyileştirme gerektiğini ortaya koymaktadır. Gelecekte, doğruluğu artırmak için veri setinin genişletilmesi ve farklı veri artırma tekniklerinin uygulanması düşünülebilir.

4. SONUÇ VE ÖNERİLER

4.1 Öneriler

Tez çalışması kapsamında, zeytin ağaçlarındaki zararlı ve zararsız böceklerin tespiti için YOLOv8 algoritmasına dayalı bir sistem geliştirilmiştir. Teknofest'teki yapıyor olduğumuz projenin sayesinde montajı yapılacak olan drone, Raspberry Pi 4 kullanılarak donatılacak ve zeytin bahçelerinde gerçek zamanlı böcek tespiti yapacak şekilde çalıştırılacaktır.

Model Eğitimi: YOLOv8 algoritmasının YOLOv8n ve YOLOv8s versiyonları kullanılarak model eğitimi gerçekleştirilmiştir. Sonuç olarak bu iki modelin sınırlı veri setlerde çok büyük fark yapmadığı ulaşılmıştır. O nedenle hız açısından ve deneme amaçlı eğitim yapılamak istenirse, YOLOv8n versiyonu kullanılması sonuca varılmıştır.

Gerçek Zamanlı Tespit: Geliştirilen sistem, zeytin bahçelerinde canlı olarak böceklerin tespit edilmesini sağlayacaktır. Bu amaçla, drone üzerinde çalışacak olan model, önceden eğitilerek en iyi performansı gösterecek şekilde optimize edilmiştir.

4.2 Çalışmanın Uygulama Alanı ve Faydaları

Bu çalışmanın uygulama alanı, zeytin yetiştiricilerinin günlük operasyonlarını iyileştirmek ve verimliliklerini artırmak için geliştirilen sistemle ilgilidir. Zeytin bahçelerindeki zararlı böceklerin tespiti ve izlenmesi, zeytin üretiminin kalitesini ve miktarını doğrudan etkileyen önemli bir faktördür. Geliştirilen sistem, zeytin yetiştiricilerine şu şekilde fayda sağlayabilir:

- **Zararlı Böcek Tespiti:** Sistem, zeytin ağaçlarında zararlı böcekleri otomatik olarak tespit ederek erken müdahale imkanı sağlar. Bu, zararlı böceklerin yayılmasını engeller ve zeytin verimliliğini artırır.
- **Zararsız Böceklerin İzlenmesi:** Aynı zamanda, zararsız böcek türlerini de izleyerek biyolojik dengeyi korur ve doğal mücadeleyi destekler. Örneğin, uğur böcekleri gibi yararlı böceklerin popülasyonunu izlemek, zararlıların kontrol altında tutulmasına yardımcı olabilir.
- **Kimyasal Mücadele İhtiyacını Azaltma:** Erken teşhis ve müdahale sayesinde, kimyasal mücadele yöntemlerine olan ihtiyacı azaltır. Bu da hem çevresel etkiyi azaltır hem de zeytin ürünlerinin organik ve sağlıklı olmasını sağlar.
- **Verimlilik ve Kaliteyi Artırma:** Zararlıların kontrol altında tutulması, zeytin verimliliğini artırır ve ürün kalitesini yükseltir. Bu da zeytin yetiştiricilerinin daha rekabetçi olmalarını sağlar.
- **Maliyet ve İş Gücü Tasarrufu:** Otomatik böcek tespiti ve izleme, işçilik maliyetlerini azaltır ve iş gücünden tasarruf sağlar. Bu da işletme maliyetlerini düşürür ve karlılığı artırır.

Zeytin yetiştiricileri, geliştirilen sistem sayesinde zeytin bahçelerindeki böcekleri etkin bir şekilde izleyebilir ve müdahale edebilir. Bu da daha sağlıklı ve verimli bir zeytin üretimine olanak tanır, böylece hem çiftçiler hem de tüketiciler için önemli faydalar sağlar.

5. KAYNAKLAR

Wikipedia contributors. (9 April 2024). "Confusion matrix." Wikipedia, The Free Encyclopedia. Retrieved 10 June 2024, from https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=1218102928.

Ye, R., Gao, Q., Qian, Y., Sun, J., & Li, T. (2024). Improved Yolov8 and Sahi Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea. Agronomy, 14(5), 1034.

Quimbiamba, F., Pérez-Pérez, N., Benítez, D., Riofrío, D., Grijalva, F., Yépez, F., & Baldeon-Calisto, M. (2023, October). Using YOLOv8 and Active Contour Models to Detect and Segment Ladybird Beetles in Natural Environments. In 2023 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (Vol. 7, pp. 1-6). IEEE.

Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., ... & Zhang, Y. (2022). Deep learning based detector YOLOv5 for identifying insect pests. Applied Sciences, 12(19), 10167.

Guo, J., Lou, H., Chen, H., Liu, H., Gu, J., Bi, L., & Duan, X. (2023/07/01). A new detection algorithm for alien intrusion on highway. Scientific Reports, 13. doi:10.1038/s41598-023-37686-w

Sakin, M. Medium - YOLOv8'in Gücünü Keşfedin https://medium.com/meryemmsakinn/yolov8in

ChatGPT. (2024). OpenAI Large Language Model. OpenAI. https://chatgpt.com/

University of Florida. (2024). Cottony Cushion Scale. Retrieved from https://entnemdept.ufl.edu/creatures/

Shutterstock. (2024). Video Resources. Retrieved from https://www.shutterstock.com/tr/video

Ultralytics. (2022). YOLOv8 Model Performansı. GitHub. https://github.com/ultralytics/ultralytics

Roboflow. (2024). Roboflow Documentation. Roboflow. Retrieved from https://roboflow.com/

ÖZGEÇMİŞ

Ad-Soyad	**:Wahib Hael Abdulwareth MOQBEL**
Doğum Tarihi ve Yeri	**: 01/01/1995 / Yemen**
E-posta	**:** abdulwareth119@gmail.com