




Salata-Marulda Derin Öğrenme Yöntemleri Kullanılarak Robotik Hasat Kriterlerinin Tespiti

Erhan KAHYA^{1*} , Fatma Funda ÖZDÜVEN² , Yasin ASLAN³ 

¹Tekirdağ Namık Kemal Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Elektronik ve Otomasyon Bölümü, Tekirdağ

²Tekirdağ Namık Kemal Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bitkisel ve Hayvansal Üretim Bölümü, Tekirdağ

³Tekirdağ Namık Kemal Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Tekirdağ

*Sorumlu Yazar: ekahya@nku.edu.tr

Geliş Tarihi: 18.05.2023 Düzeltme Geliş Tarihi: 06.09.2023 Kabul Tarihi: 07.09.2023

ÖZ

Bu çalışmada, salata marul yetiştiriciliğinde derin öğrenme metodlarından YOLOv5n, YOLOv5s ve Yolov5m kullanılarak hasat zamanı tespiti belirlenmeye çalışılmıştır. Herbir metod için 640x640 çözünürlük üzerinden eğitim yapılmıştır. Oluşturan üç modelin tüm metrik değerleri incelenmiştir. Denemeler sonucunda en başarılı modelin YOLOv5n olduğu görülmüştür. Model değerleri sonuçları "metrics/precision", "metrics/recall", "metrics/mAP_0.5" ve "metrics/mAP_0.5:0.95" olarak incelenmiştir. Bunlar, bir modelin tespit başarısını ölçen anahtar metriklerdir ve ilgili modelin doğrulama veri kümesinde gösterdiği performansı belirtmektedir. "Model 1" modelinin metrik verileri, diğer modellerle kıyaslandığında daha yüksek olduğu tespit edilmiştir. Buradan "Model 1" in robotik marul hasadında, marulun hasat kriterin bulunması için kullanılacak en iyi tespit modeli olduğu anlaşılmıştır.

Anahtar kelimeler: Derin öğrenme, Yolov5, marul, hasat

Detection of Robotic Harvesting Criteria Using Deep Learning Methods in Salad-Lettuce

ABSTRACT

In this study, harvesting time detection was tried to be determined by using deep learning methods YOLOv5n, YOLOv5s and Yolov5m in salad lettuce cultivation. For each method, training was performed over 640x640 resolution. All metric values of the three models were analysed. As a result of the experiments, YOLOv5n was found to be the most successful model. The results of the model values are analysed as "metrics/precision", "metrics/recall", "metrics/mAP_0.5" and "metrics/mAP_0.5:0.95". These are key metrics that measure the detection success of a model and indicate the performance of the respective model on the validation dataset. The metric data of the "Model 1" model was found to be higher compared to the other models. Therefore, it is concluded that "Model 1" is the best detection model to be used in robotic lettuce harvesting to find the harvesting criteria of lettuce.

Key words: Deep learning, Yolov5, lettuce, harvesting

GİRİŞ

Derin öğrenme, yapay sinir ağları olarak bilinen öğrenme modellerinin kullanıldığı bir makine öğrenmesi yöntemidir. Derin öğrenme, karmaşık problemleri çözmek ve daha doğru sonuçlar elde etmek için büyük miktarda veri kullanarak karmaşık veri yapılarını otomatik olarak öğrenir. Bu makine öğrenmesinde yapay sinir ağları adı verilen bir öğrenme modeli kullanır. (Anonim 1) belirtildiği gibi derin öğrenme (aynı zamanda derin yapılandırılmış öğrenme, hiyerarşik öğrenme ya da derin makine öğrenmesi) bir veya daha fazla gizli katman

içeren yapay sinir ağları ve benzeri makine öğrenme algoritmalarını kapsayan çalışma alanıdır. (Anonim 2) açıklandığı üzere yapay sinir ağları insan beynindeki sinir ağlarına benzer bir şekilde işleyen bir ağıdır. Yapay zekâ, insan zekâsına benzer şekilde çeşitli görevleri yerine getiren ve sürekli kendini geliştiren sistemler veya makinalardır. 1950'lerde ortaya çıkan yapay zekâ hatalarından öğrenebilen sistemler olduğu için sistemi sürekli iyileştirmektedir. Makine öğrenmesi ise 1980'lerde ortaya çıkmış olup verilen bir veri kümesini işlemden geçirip tahminlerde bulunmak veya sınıflandırmaktır. Bu ağ, katmanlar halinde düzenlenmiştir ve her katman, girdileri işleyerek çıktıları üretir. Derin öğrenme, çok katmanlı yapay sinir ağlarının kullanımı ile bu katmanların sayısının artması ile gerçekleştirilir. Karmaşık verilerin işlenmesine olanak tanır ve öğrenme sürecindeki hata oranlarını en aza indirmek için ağırlık ayarlamaları yaparlar. Bu yöntem özellikle robotik sistemlerin geliştirilmesinde ve kullanılmasında temel unsur olarak kullanılmaktadır. Tıptan ziraata kadar birçok alanda kullanılan derin öğrenmenin güncel kullanılmaktadır.

Bu araştırmada dünyada ve Türkiye yaygın üretimi yapılan ve tüketilen salata marul yetiştiriciliğinde hasat kriterlerini belirlemek için en uygun derin öğrenme metodları belirlenmeye çalışılmıştır. (Anonim 3) 'de derin öğrenme yöntemleri şu şekilde verilmiştir. Bunlar Multilayer Perceptron Neural Network (MLPNN), Backpropagation, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Generative Adversarial Network (GAN), Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN)'dir. Bu yöntemler ile robotik marul hasadı yapılabildiği gibi bitki yapısını, büyüme sürecini ve marulun hasat edileceği de belirlemeye yardımcı olur. Derin öğrenme özellikle hassas tarım uygulamalarından olan seçici tarım uygulamalarında kullanılmaktadır. Seçici hasat uygulamaları, ürünlerin hasat edilmesi veya ayrıştırılması sırasında doğru tanımlama ve sınıflandırma yapılmasını gerektiren bir işlemdir. Ayrıca seçici hasat uygulaması, hasat edilecek bitkilerin belirli bir bölümünün veya belirli özelliklerinin seçilerek hasat edilmesi anlamına da gelir. Bu işlem genellikle insanların manuel olarak gerçekleştirdiği bir işlem olsa da, derin öğrenme yöntemleri bu işlemi otomatikleştirmek için kullanılabilir. Hassas hasat, tarım ürünlerinin optimize edilmiş bir şekilde hasat edilmesi için geliştirilmiş bir teknolojidir. Bu teknolojinin amacı, ürünlerin kalitesini arttırmak, kayıpları minimize etmek ve işletme maliyetlerini azaltmaktır. Seçici hasat uygulaması ise hassas hasat teknolojileri içinde yer alan bir yöntemdir. Bu yöntemde, hasat makinesi veya diğer tarım makineleri, ürünlerin özelliklerine göre ayarlanmış sensörler ve diğer teknolojilerle donatılmıştır. Bu sayede, ürünlerin en iyi durumda hasat edilmesi ve hasat edilemeyecek olanlarının bırakılması sağlanır.

Derin öğrenme tarımsal alanda sınıflandırma ve hasat sistemlerinde kullanılmaya başlanmıştır. Son yıllarda ise yapay zeka ile sınıflandırma teknikleri tarım alanında çok önemli bir rol oynamaktadır. Sadece bitki ve toprak hastalıklarını tespit etmeye yardımcı olmakla kalmaz, aynı zamanda hasadı ve ekonomik çıktıları optimize etmeye de yardımcı olmaktadır. Derin Öğrenme ile entegre Makine Öğrenimi, bitkilerin hastalıklarını ve toprak kalitesini tespit etmek, sınıflandırmak veya tahmin etmek, yeterli miktarda sulamayı ve gübreyi optimize etmek amacıyla "teknikler" geliştirmek için büyük ölçüde kullanılmaktadır (Diop ve ark., 2020). Derin öğrenme ile şeftali hastalıklarının tespitini deneysel olarak çalışmıştır. Yapılan çalışma TRB1 bölgesinden alınan gerçek hastalık görüntülerinden oluşan veri seti ile gerçekleştirilmiştir. Deneysel çalışmada %99,30 doğruluk oranında hastalık tespiti yapılmıştır. Mevcut çalışmalardan %1,44 daha yüksek doğruluk oranına sağlanmıştır (Aslan 2021). Bir diğer çalışmada (Doğan,2019) bitkilerde görülen hastalıkların derin öğrenme yöntemleriyle tespiti ve sınıflandırılmasını yapmıştır. Bu deneysel çalışma domates ve elma yaprakları için ayrı ayrı ele alınmıştır. Domates yapraklarından oluşan veri setinde 4 farklı hastalıklı yaprak ve 1 sağlıklı yaprak olmak üzere toplam 5 farklı sınıf için deneysel çalışmalar gerçekleştirilmiştir. CNN modeli kullanılan bir çalışmada 7000 yaprak ile eğitim seti hazırlanmıştır. 7 çeşit marulun sınıflandırılması için 1800 yaprak ile test edilmiştir. Genel sınıflandırma doğruluğu %97,8 olarak tespit edilmiştir. Bununla birlikte seçilen marul çeşitleri için, bireysel sınıflandırma doğrulukları oranları %94 ile %100 arasında değişiklik göstermiştir. Bu çalışmanın sonucunda, belirli bir marul çeşidini tanımlamak için bir makine öğrenme tekniği, yani CNN kullanmanın yüksek etkililiğini kanıtlamıştır (Hassim & Chuah, 2020). (Brahimi ve ark., 2017), domatesin sınıflandırılması ve semptomların görselleştirmesi çalışmalarında, veri seti olarak dokuz hastalık tarafından enfekte edilmiş 14828 domates yaprağı görüntüsünü kullanmışlardır. %99,18'lik bir doğruluk oranına ulaşmışlardır.

Bu araştırmada sera ve benzeri alanlarda salata marulun robotik hasadın yapılması için , derin öğrenme yardımıyla görüntü işleme yapılmıştır. Seçici olarak ürün toplanması, tüm hasadın robotik sistemler ile yapılması için kullanılacak YOLOv5 algoritmasında hangi modelin daha uygun olduğu denemeler yardımıyla tespiti yapılmıştır. Yazılan sistem yazılımı ile bu tür benzer ürünlerin hasadında standardizasyon ve hassas hasat yapılması beklenmektedir.

MATERYAL ve METOT

Materyal

















Bu araştırmada materyal olarak kullanılan Salata-marul (*Lactuca sativa*), en geniş bitki familyalarından biri olan Asteraceae (Compositae) familyasından dünya genelinde üretimi yaygın olarak yapılan, yaprakları yenilen, açıkta ve örtü altında yıl boyu yetiştirilebilen bir serin iklim sebzesidir. (Vural ve ark.,2000). Taze yaprakları sebze olarak değerlendirilen salata marul, ekonomik önemi yüksek sebze türleri içinde yer almaktadır. Vitamin ve mineral madde bakımından zengin içeriğe sahip yaprağı tüketilen sebze türlerinin insan beslenmesinde önemli bir yeri vardır. Salata marulun yüksek miktarda lif içermesi ve düşük kaloriye sahip olması nedeniyle Akdeniz beslenme modelinde önemli yerinin olması, modern yaşamda birçok kişi tarafından tercih edilmesini sağlamaktadır. Birçok sebze türü ile karşılaştırıldığında kısa vejetasyon süresine sahip salata marul, uygun yetiştirme koşullarında diğer sebze türleri ile ekim nöbetine alınarak yıl içerisinde aynı üretim alanında birden fazla ürün alınabilmesine imkân tanır (Eşiyok 2012).

Salata -marulun morfolojik özellikleri arasında en önemli farkı yaprak şekli ve özellikleri belirlemektedir. Yaprakların düz kıvrıkcık olması veya farklı renk tonlarında olması belirleyici bir faktördür. Yapraklardaki renk skalası değişimi koyu yeşilden açık yeşile, açık yeşilden sarımtıraka, kahverengiden kırmızımsı gibi renklere kadar değişim gösterir ama genellikle daha çok yeşil renk tercih edilir (Eşiyok 2012). Yapraklar büyüklük açısından; büyük, orta ve küçük, yaprak şekli bakımından ise yuvarlak, oval-yuvarlak, basık-yuvarlak, basık, uzun ve orta-uzun olarak tanımlanmaktadır (Günay 1993).

Günümüzde yaprak şekli ve yaprak renginin yanı sıra göbek ve baş oluşturma, yaprak etliliği, yaprak gevrekliliği erkencilik ile hastalık ve zararlılara mukavemet gibi çeşitli özellikler dikkate alınarak çok çeşitli araştırmalar yapılmakta olup yeni çeşitler geliştirmeye devam edilmektedir. Bu özellikler dikkate alındığında salata marullar genelde kıvrıkcık yapraklı salatalar, yağlı salatalar ve marullar olmak üzere üç ana gruba ayrılmıştır (Thompson ve ark., 1957). Salata marul kuvvetli ve oldukça derine inen etli bir kazık köke sahiptir. Bu kazık kök besin deposu olup etrafında bol miktarda saçak kök taşır. Saçak kökler doğrudan kazık kök üzerinden çıkar ve toprağın 20-30 cm derinliğinde dağılım göstermişlerdir (Bayraktar 1981).Araştırma kapsamında nesne tespiti ve analizi hedeflenen salata-marul için, veri setini hazırlanmıştır. Veri setinin hazırlanmasında kullanılan kıvrıkcık salata ve lolorossolar Namık Kemal Üniversitesi Teknik Bilimler MYO Seracılık Programı Uygulama Seralarında yetiştirilmiştir. Yetiştirilen bitkilerin fotoğrafları çekilmiş ve ayrıca gerçek zamanlı olarak alınan video görüntüleri de kullanılmıştır. Bu araştırmada bölgede yaygın olarak yetiştiriciliği yapılan standart çeşit arapsaçı kıvrıkcık salata ve akdeniz yeşilliği (lolo rosso) elde edilen fotoğraflar mevcuttur. Bitkilerin seraya şaşırtılmasından hasat dönemine kadar her hafta düzeli olarak fotoğraflar çekilmiştir. Elde edilen görseller arasında, yapmış olduğumuz çalışma kapsamında değerlendiremeyeceğimiz görseller elenmiştir. Fotoğraflar 4.11.2022 tarihi itibarıyla çekilmeye başlanmış ve her hafta olarak düzenli çekilmiştir. Hasattan önceki tarih olan 23.12.2022 tarihinde fotoğraf çekimi sonlandırılmıştır.Yapacağımız nesne tespit çalışması için sağlıklı olacak 378 adet görsel belirlenmiştir. Şekil 1’de çekilmiş olan fotoğraflardan birer örnek resimler , Şekil 2 ve 3’de modellerin eğitilmesi sırasında kullanılan eğitim setinden görüntüler gösterilmiştir.

Yöntem

Bir nesne tespiti modelinin, bir veri seti üzerinde eğitim gerçekleştirebilmesi için, tespit edilmesi hedeflenen nesnelere eğitileceği veri setinde etiketlenmiş/izmlenmiş olması gerekmektedir. Bu sebeple 378 görselin her birinde marul görüntüsünü içeren kısımların sınırlayıcı kutu alanı ile işaretlenmesi ve ait olduğu nesne sınıfı olan “4 haftalık fide veya olgunlaşmış fide” sınıfına atanması gerekmektedir. Görsel etiketleme yapmak için, açık kaynak topluluklarında çok sayıda program, internet sitesi ve yardımcı araç bulunmaktadır. Bu araçlardan bir tanesi ise, nesne tespiti çalışmalarında en sıklıkla kullanılan, popüler bir web sitesi olan Roboflow’dur. Roboflow, ham görüntüleri özel olarak eğitilmiş bir bilgisayar görme modeline dönüştürmek ve onu uygulamalarda kullanmak üzere dağıtmak için gereken tüm araçları sağlayan; görüntüler üzerinde alan seçimleri, işaretlemeleri ve sınıf etiketlemeleri yapılmasını sağlayan bir web sitesidir. Bu işaretleme ve etiketleme işlemi, sitenin sahip olduğu grafiksel kullanıcı arayüzü aracılığıyla kolaylıkla yapılmaktadır. Şekil 4 ve Şekil 5’de etiket ekranı gösterilmiştir.

	4.11.2022	11.11.2022	18.11.2022	25.11.2022
Lolo-Rosso				
Kıvırcık				
	02.12.2022	09.12.2022	16.12.2022	23.12.2022
Lolo-Rosso				
Kıvırcık				

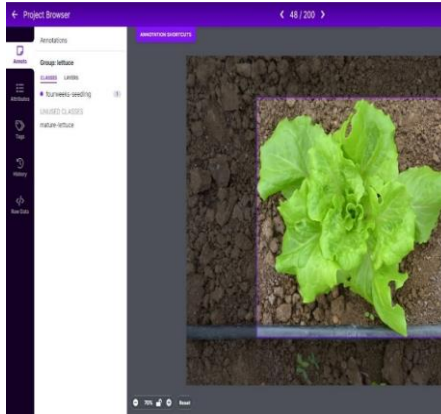
Şekil 1.Haftalık çekilen fotoğraf örnekleri



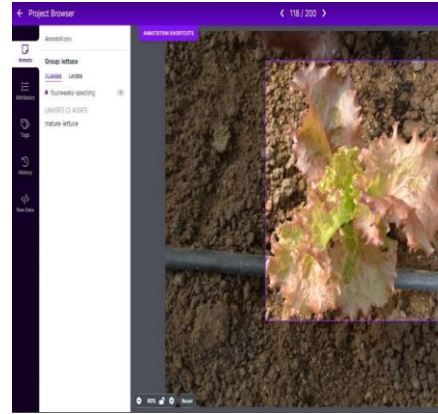
Şekil 2.Modellerin eğitilmesi sırasında kullanılan eğitim setinden görüntüler (dört haftalık marul fidesi)



Şekil 3.Modellerin eğitilmesi sırasında kullanılan eğitim setinden görüntüler (olgunlaşmış marul)



Şekil 4.Etiket Ekranı(Kıvrıkcık marul)



Şekil 5.Etiket Ekranı(Lolo-rosso)

Açılan görsel arayüzün sol ve sağ kısımları kullanılarak, programın genel işlevleri yerine getirilmektedir. Açılan görüntünün işaretlemesi, sağdaki menüden “Bounding Box Tool” seçeneği ile aktifleşen işaretleme aracı ile yapılmaktadır. Araç ile ilgili nesnenin sınırlayıcı kutu (Bounding Box) alanının belirlenmesinin ardından, belirtilen nesnenin sınıf adının etiketlenmesi gerekmektedir. Yaptığımız çalışma dahilinde, bu sınıf adlarını “fourweeks-seedling ve mature-lettuce” olarak belirledik. İlgili nesne tespit modelinin eğitimi bu görüntülerle yapılacağından dolayı, seçim işleminin nesneyi içeren şekliyle doğru bir biçimde yapılması gerekmektedir. Seçim ve etiketleme işleminin ardından, ana menü kısmındaki “generate” sekmesinden yapılan işaretlemelerin çıktı formatı seçilmelidir. Çalışma kapsamında, kullanılacak modele uygun olan YOLO seçeneği seçilmiştir. Tüm bu işlemlerin ardından görüntü işlenerek otomatik kaydedilebilmektedir.

Derin Öğrenme Eğitim Modelin Seçimi

Bu çalışmada, CNN yöntemiyle geliştirilmiş YOLO model ailesinin açık kaynak olarak geliştirilen YOLOv5 ailesi tercih edilmiştir. RCNN benzeri iki aşamalı ağ kullanan modellere göre oldukça avantaja sahip olan YOLO model ailesinin YOLOv5 modeli, kendisinden önce geliştirilmiş sürümlere doğruluk değerleri ve hız oranı açısından avantaj sağladığı için tercih edilmiştir.YOLOv5 modeli kendi içerisinde modeller barındırmaktadır. YOLOv5n/s/m (nano, small ve medium) modeli derin öğrenme eğitimi için tercih edilmiştir.

Derin Öğrenme Eğitiminin Başlatılması

Marul tespiti yapacak olan modelin eğitiminin başlatılması için, YOLOv5 modelinin bilgisayardaki konumuna gidildi ve burada bir Python çalıştırıcı editör açıldı. Ana dizinde bulunan ve YOLOv5 eğitimini sağlayan train.py programı, çalıştırılmak üzere kontrol edildi. Bu Python programının çalıştırılması, çeşitli parametreler ile özelleştirilebilmektedir.

Çalışma dahilinde, dört haftalık fideler ve olgunlaşmış marul fideleri için sırasıyla aşağıda yazılı olan kod içerisindeki parametreler ve düzenlemeler tercih edildi.

```
python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5n.pt
```

```
python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5s.pt
```

```
python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5m.pt
```

--img: Eğitimi yapılacak görsellerin YOLOv5 modeli tarafından düşürüleceği piksel boyutu. Varsayılan değeri 640x640 olarak belirlenmektedir ve burada da bu şekilde seçilmiştir.

--batch: Modeli eğitirken, ekran kartı tarafından bir seferde kullanılacak veri noktası paketi sayısıdır.

--epochs: Modelin eğitimi yapılırken, tüm eğitim verilerinin eğitilen ağa gösterilme, ağırlıkların güncellenmesi sayısıdır.

--data: Veri setini içeren dosyanın genel yol ve sınıf bilgisini içeren .yaml dosyasının yoludur.

--weights: Modelin eğitiminde kullanılacak eğitim katsayılarını içeren ağırlık dosyasının konumudur.

Bu kod satırlarının doğru şekilde çalıştırılması sonucunda, modellerin eğitim işlemi başlamıştır. Program, öncelikle YOLOv5 dosyalarını kontrol edip, herhangi bir güncelleme durumunu kontrol etmektedir. Ardından, belirlenen döngü sayısı (epoch) boyunca eğitim işlemi yapılmaktadır.

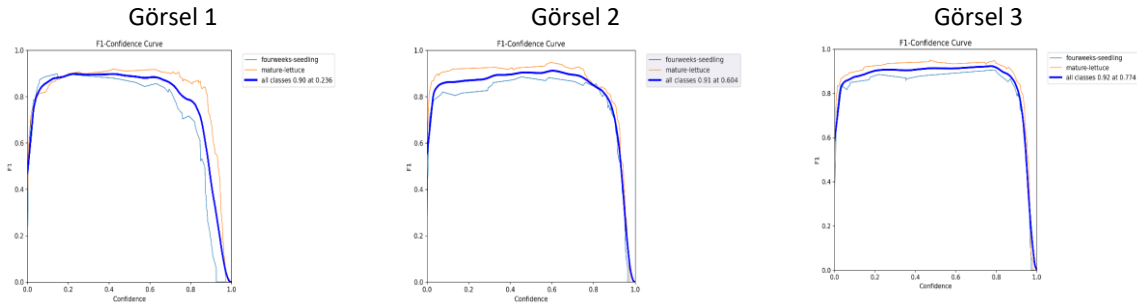
YOLOv5 algoritmalarının sonuçlarının hata matrisi metriklerine göre incelenmesi:

F1 Score:

Görsel 1: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n

Görsel 2: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s

Görsel 3: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m



YOLOv5 modelinin farklı varyasyonları olan Nano, Small, Medium modellerin sonuçları grafiksel olarak incelenmiştir. Her bir modelin öğrenme sonuçlarındaki farklılıkları analiz ederek en başarılı ve en başarısız modeli belirlenmesi yapılmıştır. Bu sonuçları desteklemek için metrikler üzerinde bir karşılaştırma yapılmış ve F1 skorlarına göre sonuç analizleri aşağıdaki gibidir.

YOLOv5 Nano modeli;

İlk 3 epoch boyunca, modelin eğitim kaybı (box_loss, obj_loss ve cls_loss) düşerken, precision (kesinlik) ve recall (hatırlama) metrikleri artmıştır. Bu, modelin nesnelere daha iyi algılamaya başladığını ve sınıflandırmayı daha doğru bir şekilde gerçekleştirdiğini gösterir. Ancak, mAP (ortalama hassasiyet) değeri ilk 3 epoch boyunca düşük kalmıştır. Son 3 epoch boyunca, modelin tüm metriklerinde iyileşme görülmektedir. Kayıp değerleri düşmüş, precision ve recall değerleri artmış, aynı zamanda mAP değerleri de yükselmiştir. Bu, modelin daha iyi bir nesne tespit performansı sergilediğini gösterir. Sonuç olarak, YOLOv5 Nano modeli eğitim süreci boyunca gelişim göstermiş ve başarılı bir performans elde etmiştir.

YOLOv5 Small modeli;

İlk 3 epoch boyunca, bu modelde de eğitim kaybı azalmış ve precision ile recall değerleri artmıştır. Ancak, mAP değeri hala düşüktür. Son 3 epoch boyunca ise tüm metriklerde bir iyileşme gözlenmektedir. Kayıp değerleri düşmüş, precision ve recall değerleri artmış, aynı zamanda mAP değeri de yükselmiştir. YOLOv5 Small modeli, eğitim süreci ilerledikçe daha iyi bir performans sergilemiştir. Ancak, Nano modeline kıyasla daha düşük bir performans göstermiştir.

YOLOv5 Medium modeli;

Bu modelde de benzer bir eğilim görülmektedir. İlk 3 epoch boyunca kayıp değerleri azalırken, precision ve recall değerleri artmıştır. Ancak, mAP değeri hala düşüktür. Son 3 epoch boyunca ise tüm metriklerde bir

iyileşme gözlenmektedir. Kayıp değerleri düşmüş, precision ve recall değerleri artmış, aynı zamanda mAP değeri de yükselmiştir. YOLOv5 Medium modeli, daha yüksek bir performans sergilemektedir, ancak Small modele kıyasla benzer bir başarı elde etmiştir.

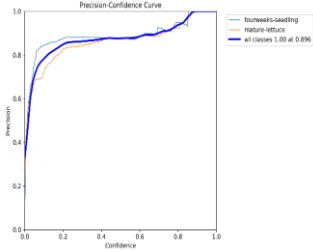
Precision:

Görsel 4: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n

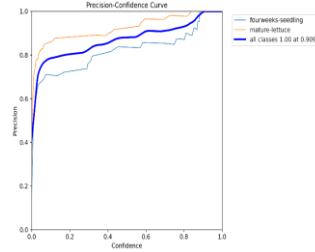
Görsel 5: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s

Görsel 6: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

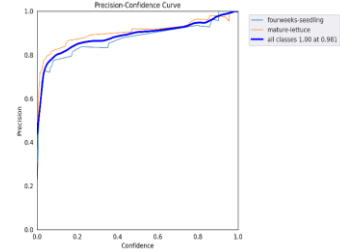
Görsel 4



Görsel 5



Görsel 6



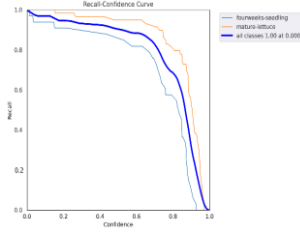
Recall:

Görsel 7: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n

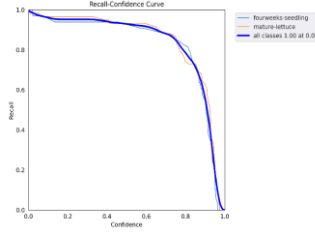
Görsel 8: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s

Görsel 9: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

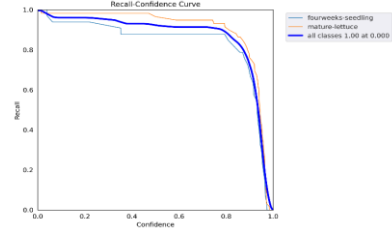
Görsel 7



Görsel 8



Görsel 9



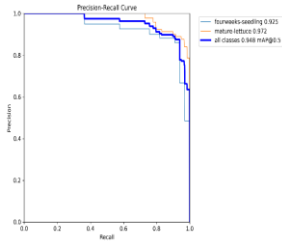
Precision Recall:

Görsel 10: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n

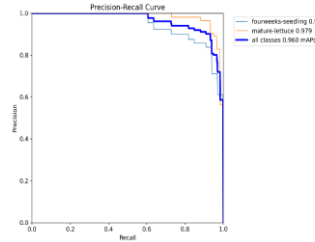
Görsel 11: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s

Görsel 12: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

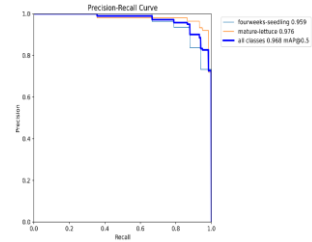
Görsel 10



Görsel 11



Görsel 12



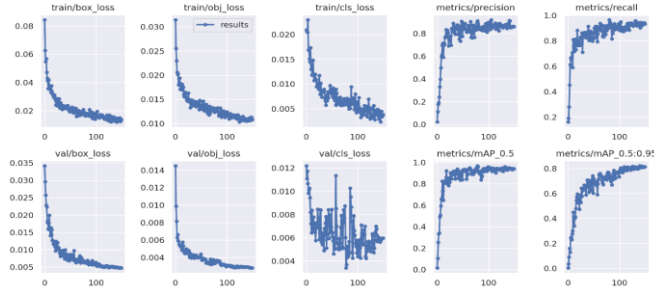
Loss Function:

Görsel 13: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n

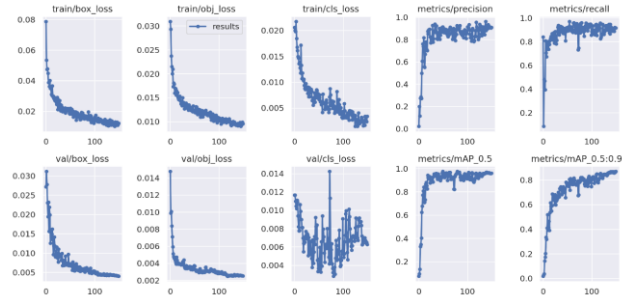
Görsel 14: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s

Görsel 15: Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

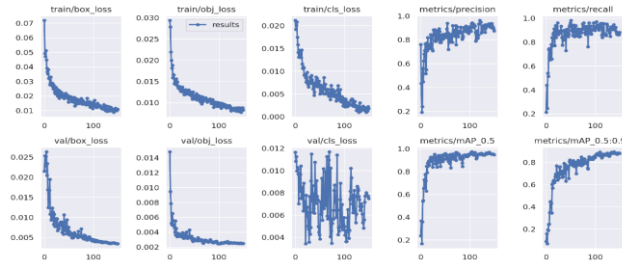
Görsel 13



Görsel 14



Görsel 15



YOLOv5 modellerinin grafiksel karşılaştırmasında elde edilen sonuçlara göre;

1. Kayıp Değerleri (Loss Values): Genel olarak, her bir modelin son epochta (epoch 150) daha düşük kayıp değerleri elde ettiği görülmektedir. Bu, modelin eğitim süresince veri setinden öğrendiği anlaşılmalıdır. YOLOv5 Nano modeli, kutu kaybı (box loss), obje kaybı (obj loss) ve sınıf kaybı (cls loss) açısından diğerlerinden daha düşük değerlere sahip, bu durum modelin nispeten daha iyi öğrendiğini göstermektedir.
2. Hassasiyet (Precision) ve Geri Çağırma (Recall): Hassasiyet, modelin doğru olarak tespit edilen nesnelerin ne kadarının gerçekten doğru olduğunu gösteren değerdir. Geri çağırma ise modelin tespit etmesi gereken tüm nesnelere ne kadar doğru bir şekilde tespit ettiğini ölçer. YOLOv5 Nano modeli, bu iki metrik açısından en yüksek performansı sergilediği görülmektedir.
3. mAP (mean Average Precision): mAP, modelin tüm sınıflar ve farklı IoU (Intersection over Union) eşiklerindeki performansının bir ölçüsüdür. Genel olarak, daha yüksek mAP, daha iyi bir model anlamına gelir. Burada YOLOv5 Nano modeli, mAP_0.5 ve mAP_0.5:0.95 değerlerinde en iyi sonucu vermektedir.
4. "train/obj_loss" metriği incelendiğinde, YOLOv5 Nano modelindeki nesne kaybının diğer modellerdeki başlangıç değerlerinden daha düşük seviyeler olduğu ve eğitim süreci boyunca azaldığı görülmektedir. YOLOv5 Nano modelinin nesnelere daha iyi algıladığını ve bu nedenle daha düşük bir nesne kaybı elde ettiği görülmektedir.
5. "train/cls_loss" metriği üzerinde yapılan karşılaştırmada, YOLOv5 Nano modelinin sınıfları daha doğru bir şekilde tahmin ettiğini ve bu nedenle daha düşük bir sınıf kaybı elde ettiği görülmektedir.
6. "metrics/precision" metriği incelendiğinde, YOLOv5 Nano modelinin daha kesin tahminler yaptığını ve bu nedenle daha yüksek bir kesinlik değeri elde ettiğini görülmektedir.
7. "metrics/recall" metriği üzerinde yapılan karşılaştırmada, YOLOv5 Nano modelinin daha fazla nesneyi doğru bir şekilde hatırladığını ve bu nedenle daha yüksek bir hatırlama değeri elde ettiğini görülmektedir.

8. "metrics/mAP_0.5" metriği incelendiğinde, YOLOv5 Nano modelinin daha doğru sınıflandırma yapabildiğini ve bu nedenle daha yüksek bir ortalama hassasiyet değeri elde ettiği görülmektedir.
9. "metrics/mAP_0.5:0.95" metriği üzerinde yapılan karşılaştırmada, YOLOv5 Nano modelinin daha doğru ve tutarlı sonuçlar elde ettiğini ve bu nedenle daha yüksek bir ortalama hassasiyet değeri elde ettiği görülmektedir
10. "val/box_loss", "val/obj_loss" ve "val/cls_loss" metriklerine bakıldığında, YOLOv5 Nano modelinin daha doğru tahminler yaptığını ve daha düşük bir hata oranına sahip olduğu görülmektedir.

Bu değerlerin ölçülmesine göre YOLOv5 Nano modeli en iyi performansı veridiği görülmektedir. Model seçimimizde, kullanılabilir kaynaklar ve uygulamanın gereksinimleri arasında bir denge olmalıdır. Buradan bir uygulamanın gerçek zamanlı çalışması gerekiyorsa ve kaynaklar sınırlıysa, biraz daha düşük performansla daha hafif bir modellemelerden olan YOLOv5 Small veya Medium 'un kullanılmasının daha uygun olacağı anlaşılmaktadır.

Derin Öğrenme Eğitim Sonucu

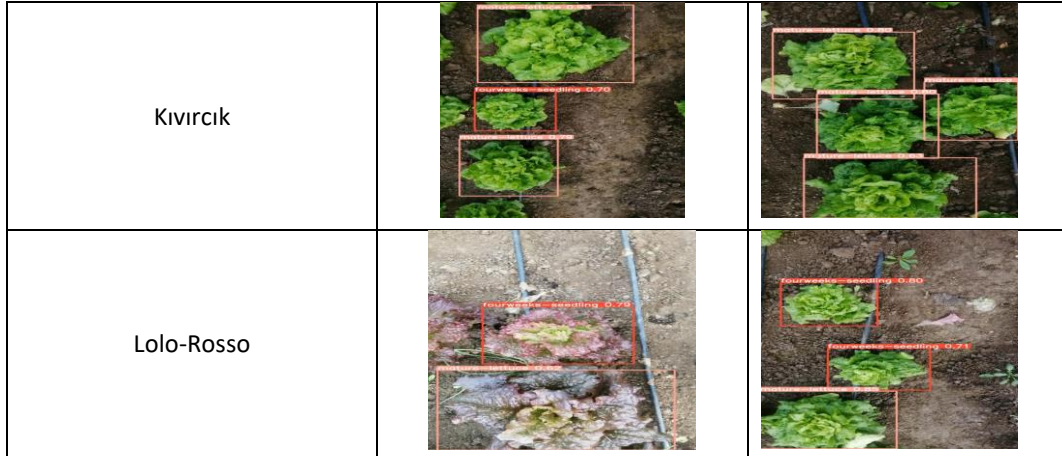
Eğitim sonuçları Şekil 6,7 ve 8'de gösterilmiştir.

Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n	Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s	Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

Şekil 6. Modellerin eğitimlerinin sonucunda ortaya çıkan "Validation Batch" tahmin işaretlemeleri (dört haftalık marul fidesi)

Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5n	Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5s	Size: 640x640, Batch: 10, Epoch: 150, Algorithm: YOLOv5m

Şekil 7. Modellerin eğitimlerinin sonucunda ortaya çıkan "Validation Batch" tahmin işaretlemeleri (olgunlaşmış marul)



Şekil 8. Gerçek Zamanlı Kamera ile marulların ekili olduğu serada, test esnasında çekilmiş fotoğraflar

BULGULAR ve TARTIŞMA

Her epoch sonuna bir sütun eklenerek oluşturulan ve modellerin eğitiminin tamamlanmasıyla ortaya çıkan ayrıntılı istatistikler Çizelge 1, 2, 3 'de verilmiştir (Özetleme amacıyla, her modelin eğitiminden yalnızca ilk 3 ve son 3 epoch gösterilmiştir).

Çizelge 1. YOLOv5 NANO / python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5n.pt

Epoch	0	1	2	...	147	148	149
train/box_loss	0.084339	0.062578	0.053494	...	0.012522	0.013253	0.013777
train/obj_loss	0.031427	0.025445	0.022811	...	0.011181	0.011133	0.010957
train/cls_loss	0.020945	0.020588	0.020176	...	0.0027973	0.0035055	0.0036015
metrics/precision	0.020008	0.18018	0.75864	...	0.90069	0.89309	0.89109
metrics/recall	0.15383	0.28762	0.24576	...	0.92912	0.92912	0.92912
metrics/mAP_0.5	0.015107	0.15271	0.33177	...	0.94774	0.94765	0.94113
metrics/mAP_0.5:0.95	0.003356	0.060218	0.13171	...	0.8291	0.82535	0.82434
val/box_loss	0.03424	0.026293	0.023515	...	0.0044853	0.0044776	0.0044665
val/obj_loss	0.014556	0.0097866	0.0086721	...	0.0026424	0.0026518	0.0026605
val/cls_loss	0.012178	0.011461	0.011026	...	0.0048048	0.0047606	0.0046988
x/lr0	0.0766	0.052265	0.027894	...	0.000364	0.000298	0.000232
x/lr1	0.0026	0.005265	0.0078944	...	0.000364	0.000298	0.000232
x/lr2	0.0026	0.005265	0.0078944	...	0.000364	0.000298	0.000232

Çizelge 2. YOLOv5 SMALL / python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5s.pt

Epoch	0	1	2	...	147	148	149
train/box_loss	0.078837	0.05359	0.048799	...	0.010948	0.011467	0.011777
train/obj_loss	0.030914	0.029203	0.023101	...	0.0098081	0.010099	0.0095335
train/cls_loss	0.020605	0.020063	0.020005	...	0.0020569	0.0029358	0.0034602
metrics/precision	0.015923	0.24149	0.12794	...	0.89902	0.90934	0.80864
metrics/recall	0.81587	0.35076	0.33179	...	0.8395	0.8395	0.94541
metrics/mAP_0.5	0.083419	0.23124	0.10038	...	0.94791	0.95	0.95137
metrics/mAP_0.5:0.95	0.021502	0.089706	0.029414	...	0.85886	0.86176	0.864
val/box_loss	0.026558	0.026188	0.032579	...	0.0039587	0.0039578	0.00393
val/obj_loss	0.015022	0.0096242	0.0076129	...	0.0025498	0.0025424	0.0025293
val/cls_loss	0.011699	0.010542	0.010688	...	0.0080677	0.0081483	0.0078755
x/lr0	0.0766	0.052265	0.027894	...	0.000364	0.000298	0.000232
x/lr1	0.0026	0.005265	0.0078944	...	0.000364	0.000298	0.000232
x/lr2	0.0026	0.005265	0.0078944	...	0.000364	0.000298	0.000232

Çizelge 3. YOLOv5 MEDIUM / python train.py --img 640 --batch 10 --epochs 150 --data dataset.yaml --weights yolov5m.pt

Epoch	0	1	2	...	147	148	149
train/box_loss	0.071902	0.049314	0.048759	...	0.009626	0.010045	0.010676
train/obj_loss	0.029362	0.027931	0.021981	...	0.008705	0.008719	0.008328
train/cls_loss	0.021156	0.019114	0.020071	...	0.001302	0.002006	0.002216
metrics/precision	0.55637	0.2404	0.19041	...	0.90335	0.91072	0.90077
metrics/recall	0.2156	0.46507	0.39908	...	0.8902	0.89431	0.90717
metrics/mAP_0.5	0.23543	0.23623	0.22082	...	0.94984	0.94923	0.94787
metrics/mAP_0.5:0.95	0.078493	0.083561	0.086335	...	0.89119	0.88735	0.88959
val/box_loss	0.021734	0.029776	0.027471	...	0.003465	0.003476	0.003487
val/obj_loss	0.014719	0.009621	0.006224	...	0.002241	0.002254	0.002264
val/cls_loss	0.011597	0.010796	0.010596	...	0.005281	0.005245	0.005146
x/lr0	0.0766	0.052265	0.027894	...	0.000364	0.000298	0.000232
x/lr1	0.0026	0.005265	0.007894	...	0.000364	0.000298	0.000232
x/lr2	0.0026	0.005265	0.007894	...	0.000364	0.000298	0.000232

Tablo 1,2,3 üç farklı YOLOv5 modelinin (Nano, Small, Medium) eğitim sonuçlarını sunmaktadır. Her modelin parametreleri, yürütülen Python eğitim komutu ile belirlenmiştir. Eğitim sürecinde, modelin her bir epochu için çeşitli metrikler toplanmış ve kaydedilmiştir. Bunlar arasında kutu kaybı (box loss), nesne kaybı (obj loss), sınıf kaybı (cls loss), doğruluk (precision), geri çağırma (recall) ve Ortalama Hassasiyet (mAP) değerleri yer almaktadır. Her epochun sonunda, bu metrikler ayrıca doğrulama (validation) seti üzerinde de hesaplanmıştır.

Her modelin başarı düzeyini değerlendirmek için metriklerin son epochlardaki değerlerine göre;

1. YOLOv5 Nano: Son epochda (150), mAP_{0.5} değeri 0.94113 , mAP_{0.5:0.95} değeri ise 0.82434'tür. Bu, modelin hassasiyet ve geri çağırma açısından kabul edilebilir bir performans gösterdiği,
2. YOLOv5 Small: Son epochda (150), mAP_{0.5} değeri 0.95137, mAP_{0.5:0.95} değeri ise 0.864'dur. Bu, Nano modeline kıyasla daha düşük performans gösterdiği,
3. YOLOv5 Medium: Son epochda (150), mAP_{0.5} değeri 0.94787, mAP_{0.5:0.95} değeri ise 0.88959'dir. Bu, Small modeline kıyasla mAP_{0.5:0.95} metriği açısından daha iyi bir performans gösterdiği,

Tablodaki sonuçların analizinde YOLOv5 Nano modeli, mAP_{0.5:0.95} metriği üzerinde en yüksek skoru elde ettiği için en başarılı model olarak görülebilir. Bu, modelin yüksek çözünürlüklü görüntüler üzerinde daha karmaşık ve detaylı nesne algılama görevlerini başarıyla gerçekleştirebileceği görülmektedir.

Çizelge 4. Algoritma Değerlerinin Karşılaştırılması

Model	size (pixels)	mAP ^{val} ₅₀₋₉₅	mAP ^{val} ₅₀	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	Params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0

Eğitimi yapılan üç modelde; YOLOv5n algoritması, YOLOv5s algoritması ve YOLOv5m algoritması kullanılmaktadır. Bu algoritmaların karşılaştırılması Çizelge 4'deki gibidir:

Model "1" in metrik verileri ve diğer modellerin bu verilere göre farkı Çizelge 5'de verildiği gibidir.

Çizelge 5. Metrik verilere göre modellerin karşılaştırılması

Model	metrics/precision	Fark(Model 1'e)	Model	Metrics/recall	Fark(Model 1'e)
Model 1	0,89109		Model 1	0,92912	
Model 2	0,80864	0,08245	Model 2	0,94541	-0,01629
Model 3	0,90077	-0,00968	Model 3	0,90717	0,02195

Model	metrics/mAP _{0,5}	Fark(Model 1'e)	Model	Metrics/mAP _{0,5:0,95}	Fark(Model 1'e)
Model 1	0,94113		Model 1	0,82434	
Model 2	0,95137	-0,01024	Model 2	0,864	-0,03966
Model 3	0,94787	-0,00674	Model 3	0,88959	-0,06525

Tablo 4 ve 5 'deki değerlere göre modellerin başarılarını ölçmek için modellerin nesne tespitlerindeki doğru tahmin başarıları ve ortalamaları önemli rol oynasa da, tek başına yeterli değildir. Modellerin, hem eğitim hem de doğrulama veri kümelerindeki kayıp değerleri de model başarısının incelenmesinde rol oynayan önemli parametrelerdir. Eğitim ve doğrulama verilerindeki sınıflandırma kayıplarını ifade eden train/cls_loss ve val/cls_loss parametreleri çok sayıda nesne sınıfının tespitini gerektiren modellerde önemli bir rol oynar. Modellerin kayıp değerlerini gösteren Çizelge 6 'daki dört parametre sütunu (train/box_loss, train/obj_loss, val/box_loss, val/obj_loss) değerleri incelendiğinde, eğitim setinde en az kayıp değerlerine sahip modelin, "Model 1" olduğu görülmektedir. Fakat doğrulama verilerindeki kutu tahmini ve nesne tespiti kayıplarında en az kayıp değerlerine sahip modelin, "Model 2" olduğu görülmektedir.(Çizelge 6)

Çizelge 6. Metrik verilere göre modellerin karşılaştırılması

Model	train/box_loss	Fark(Model 1'e)	Model	train/obj_loss	Fark(Model 1'e)
Model 1	0,013777		Model 1	0,010957	
Model 2	0,011777	0,002	Model 2	0,0095335	0,0014235
Model 3	0,010676	0,003101	Model 3	0,008327	0,002629

Model	val/box_loss	Fark(Model 1'e)	Model	Val/obj_loss	Fark(Model 1'e)
Model 1	0,0044665		Model 1	0,0026605	
Model 2	0,00393	0,0005365	Model 2	0,0025293	0,0000675
Model 3	0,0034874	0,0009791	Model 3	0,002264	0,0003965

Son olarak, modellerin optimizasyon parametreleri ($x/lr0-1-2$) incelenmiştir. Tüm modellerin bu parametrelerdeki değerleri eşittir(Çizelge 7).

Çizelge 7. Modellerin optimizasyon parametreleri

Model	$x/lr0-1-2$	Fark(Model 1'e)
Model1	0,000232	
Model2	0,000232	0
Model3	0,000232	0

Üç model arasında en yüksek parametre sayısına sahip olan YOLOv5 Nano modeli, genel olarak en iyi performansı göstermektedir. Diğer modellere göre daha yüksek doğruluk, hatırlama ve mAP değerleriyle başlamakta ve eğitim sırasında bunları daha da geliştirmektedir. Kayıpların azalmasıyla model hızlı bir şekilde yakınsayacaktır. Genel olarak, en başarılı modelin seçimi, uygulamanın belirli gereksinimlerine bağlıdır. Göreceli olarak iyi performans gösteren daha küçük bir model boyutu tercih ediliyorsa, YOLOv5 Medium veya YOLOv5 Small modelleri uygun bir seçenek olabilir. Ancak, daha yüksek doğruluk öncelikli olan ve daha büyük bir modele işleyebilecek hesaplama kaynaklarına sahip uygulamalar için YOLOv5 Nano modeli en iyi sonuçları sunar.

SONUÇ ve ÖNERİLER

Yaptığımız çalışmada üç model arasından en başarılı modelin tespitinde tüm parametre değerleri ve metrikler incelenmiştir. Bir önceki bölümde yer alan, modellerin örnek eğitim ve doğrulama görsellerindeki nesne tespiti doğrulukları incelenmiştir. Modellerin nesne tespit başarılarını belirten metrik verileri ve doğruluk tahmin oranları incelendiğinde, kayıp verileri incelenirken eğitim ile doğrulama verileri arasındaki fark oranları göz önüne alındığında ve doğrulama verileri üzerindeki kayıp değerlerinin daha genel bir sonucu ifade ettiği düşünüldüğünde, optimizasyon ve öğrenme hızı değerleri incelendiğinde bu üç model arasında en başarılı modelin aşağıdaki parametrik özelliklere sahip "Model 1" olduğu doğrulanmıştır.

Tarımsal robot sistemlerinin en önemli çalışma kriteri hasadı yapılacak ürünlerin doğru sınıflandırılmasıdır. Girdi değeri olarak verilen hasat kriter parametreleri sayesinde ürünlerin hasadının robotik hasat sistemleri yardımıyla yapılması mümkündür. Doğru sınıflandırma hassas hasadın başarı oranını artıracaktır. Bu konuda birçok araştırma yapılmakta ve sistemler geliştirilmektedir. Seo ve ark (2021) yılında yaptıkları ürün sınıflandırma için derin öğrenme çalışmasında domates serası için sınıflandırma yapmışlardır. Bu çalışmada fide üzerinde tamamen gizlenmiş meyveleri yakalama oranları %88,6'dır. *Bir başka çalışma da* Mao ve ark.(2020) yılında yapmış oldukları çalışmada salatalığın yerinin tespitinde derin öğrenme modellemesi yapmışlardır. Modellemenin başarı oranını % 90 olarak belirlemişlerdir. Sane ve ark.(2021) yılında robotik hasat uygulaması için derin öğrenme metodu kullanmışlardır. Oluşturdukları derin öğrenme metodu ile robotik hasat uygulamalarında mahsulün yerini tespit etmişlerdir. Bai ve ark(2023) yılında yapmış oldukları çalışmada robotik domates hasadı için derin öğrenme yöntemi kullanmışlardır. Bu çalışma ile robotik domates hasat sistemi için kesme noktasını belirlemişlerdir. Yapılan bu benzer araştırmalar göz önüne alındığında yaptığımız araştırmanın sonuçları ile paralellik gösterdiği tespit edilmiştir.

Çıkar Çatışması Beyanı: Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

Araştırmacıların Katkı Oranı Beyan Özeti: Yazarlar makaleye eşit oranda katkı sağlamış olduklarını beyan ederler.

YAZAR ORCID NUMARALARI

Erhan KAHYA  <http://orcid.org/0000-0001-7768-9190>

Fatma Funda ÖZDÜVEN  <http://orcid.org/0000-0003-4286-8943>

Yasin ASLAN  <http://orcid.org/0009-0007-8042-9729>

KAYNAKLAR

Anonim 1, https://tr.wikipedia.org/wiki/Derin_öğrenme(Erişim tarihi:15.05.2023)

Anonim 2,https://www.beyaz.net/tr/yazilim/makaleler/derin_ogrenme_deep_learning_nedir.html(Erişim tarihi:15.05.2023)

Anonim 3,<https://alitudanacanor.medium.com/derin-ogrenme-yontemleri-ve-uygulamaları-1ce215de40e8> (Erişim tarihi:15.05.2023)

ASLAN, M. ,(2021). Derin Öğrenme ile Şeftali Hastalıklarının Tespiti. *European Journal of Science and Technology*, 23, 540–546. <https://doi.org/10.31590/ejosat.883787>

Bai, Y., Mao, S., Zhou, J., Zhang, B., (2023). Clustered tomato detection and picking point location using machine

learning-aided image analysis for automatic robotic harvesting. *Precision Agriculture*, 24(2), 727–743. <https://doi.org/10.1007/s11119-022-09972-6>

Bayraktar, K. ,(1981). Sebze Yetiştirme. Cilt II.(Kültür Sebzeleri). E.Ü. Ziraat Fakültesi Yayınları, Bornova-İzmir, No:169 480s.

Brahimi, M., Boukhalfa, K., Moussaoui, A., (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence*, 31(4), 299–315. <https://doi.org/10.1080/08839514.2017.1315516>

Diop, P. M.,Takam, O., J.,Nakamura, Y.,Nakamura, M. ,(2020). A Machine Learning Approach to Classification of Okra;35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Nagoya, Japan, 2020, pp. 254-257.

Doğan M.S, (2019), Bitkilerde Görülen Hastalıkların Derin Öğrenme Yöntemleriyle Tespiti Ve Sınıflandırılması, Yalova Üniversitesi ,Fen Bilimleri Enstitüsü ,Bilgisayar Mühendisliği Ana Bilim Dalı , Bilgisayar Mühendisliği Bilim Dalı,Yüksek Lisans Tezi.

Eşiyok, D. ,(2012). Kışlık ve Yazlık Sebze Yetiştiriciliği. 404 s. Bornova, İzmir.

Günay, A. ,(1993). Özel Sebze Yetiştiriciliği Cilt V. A.Ü. Ziraat Fakültesi Bahçe Bitkileri Bölümü, Ankara, s. 531.

Hassim, S. A., Chuah, J. H. (2020). Lettuce classification using convolutional neural network. *Food Research*, 4, 118–123. [https://doi.org/10.26656/fr.2017.4\(S6\).029](https://doi.org/10.26656/fr.2017.4(S6).029)

Mao,S.,Li,Y.,Ma,Y.,Zhang,B.,Zhou,J.,Wang,K.,(2020),Automatic Cucumber Recognition Algorithm For Harvesting Robots In The Natural Environment Using Deep Learning And Multi-Feature Fusion, *Computers and Electronics in Agriculture* 170 (2020) 105254, Elsevier Publishing.

Sane, T. U., Sane, T. U., (2021). Artificial Intelligence and Deep Learning Applications in Crop Harvesting Robots-A Survey. 3rd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2021. <https://doi.org/10.1109/ICECCE52056.2021.9514232>

Seo,D.,Cho,B.,Kim,K., (2021). Development of monitoring robot system for tomato fruits in hydroponic greenhouses. *Agronomy*, 11(11), 2211, doi: 10.3390/agronomy11112211

Thompson, C.H., Kelly, W.C., (1957). *Vegetable Crops*. Mc Graw Hill BookCompany Inc., New York

Vural, H., D. Eşiyok ve İ. Duman, (2000). *Kültür Sebzeleri (Sebze Yetiştirme)*. Ege Üniversitesi Basım Evi, Bornova, İzmir.