



Application of YOLOv8L Deep Learning in Robotic Harvesting of Persimmon (*Diospyros kaki*)

Erhan KAHYA^{*1}, Fatma Funda ÖZDÜVEN², Berat Can CEYLAN³

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

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

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

*Sorumlu Yazar (Corresponding author): ekahya@nku.edu.tr

Abstract

Deep learning has been a branch of science that has been used by many researchers and has gained popularity in recent years. Deep learning techniques perform better than traditional methods by providing high accuracy in analyzing and processing agricultural data. Therefore, the use of deep learning techniques in agriculture is increasing. The persimmon used in this study is a fruit tree belonging to the Ebenaceae family and is cultivated in various regions of Turkey, including the Trabzon region. The persimmon harvest is typically done during the fall season when the fruits reach optimal maturity. It is recommended to harvest the persimmon when they are hard but slightly soft to the touch. In this study, using the deep learning method, the classification was made by considering the color feature of the fruit. The aim here is to develop a method to be used in robotic harvesting systems. YOLOv8L was chosen as the method. The metric values of the model were analyzed and it was observed that the 'metrics/precision', 'metrics/recall', 'metrics/mAP_0.5' and 'metrics/mAP_0.5:0.95' values of the model increased as the number of epochs increased. In the last epoch, precision was measured at about 71%, recall was measured at 79%, mAP_0.5 was measured at about 84% and mAP_0.5:0.95 was measured at about 76%. These values indicated that the model was able to detect and classify objects with high accuracy in the validation set. Measured value Size: 640x640, Batch: 16, Epoch: 102, Algorithm: YOLOv8L. It was concluded that YOLOv8L was the best detection model to be used in robotic persimmon harvesting to separate the persimmon from branch.

Research Article

Article History

Received :13.03.2023
Accepted :18.04.2023

Keywords

Persimmon
YOLOv8
classification

1. Introduction

Deep learning is a machine learning method designed to solve complex problems using artificial neural networks. With deep learning, we can make a calculation consisting of multiple processing layers of artificial neural networks (Aktaş, 2022). It is a method used to learn complex functions and can create complex structures that can represent high-level abstractions (Bengio, 2009). It is used in many areas such as image processing, biomedical signal processing, face recognition, health applications and recommender systems (Karasulu et al., 2022). In addition, the deep learning model is also used in tasks such as automatic segmentation and classification in medical imaging (Şentürk, 2023). Deep learning enables the construction of deeper models using unsupervised learning methods. This method enables single-layer models such as Restricted Boltzmann Machines to be learned with unsupervised learning and deeper models, such as Deep Belief Networks, to be constituted. Deep learning uses learning algorithms to learn the parameters of these deep models (Bengio, 2009). Deep learning has found its place in various applications today. It is used in different areas such as air pollution prediction, network traffic classification analysis and autonomous road planning (Aydin et al., 2021; Kirişoğlu et al., 2022; Tan and Karakose, 2022). Deep learning also plays an important role in healthcare field. A deep learning-based decision support software has been developed for magnetic resonance imaging scans (Eren et al., 2021). In addition, deep learning is also used as a method used in the examination of molecular genetic predisposition problems and disease analysis (Deng, 2014). Deep learning has been a field that has gained popularity in recent years and is actively studied by many researchers (Çetiner et al., 2022).

Deep learning, which has been widely used in the field of agriculture, has started to be used in many areas such as plant disease detection and classification, robotic harvesting systems, crop yield estimation. Especially with deep residual learning, it has given successful results in tasks such as fruit classification. Deep residual networks provide a framework to facilitate the training of networks that are deeper than previously used networks. These networks clearly reformulate layers as residual functions that are learned with reference to layer inputs. By this means, deeper networks can be trained more easily and higher accuracy can be achieved. One of the reasons why deep learning methods are successful in tasks such as fruit classification is the better representing ability of the complexity in data sets of multilayer structures. Deep residual networks especially provided significant improvement in visual recognition tasks and an improvement of up to 28% were achieved on the COCO object detection dataset (He et al., 2016). The success of deep learning methods on tasks such as fruit classification shows that deep neural networks are also effective on large-scale image recognition tasks more generally. Such deep neural networks are pioneers in the use of deep visual representations and achieve the best results on other datasets (Simonyan, 2014). In a study carried out by (Arserim et al., 2023), the objects in videos obtained from unmanned aerial vehicles were detected by using deep learning methods. In another study, a system for fruit detection was developed using deep neural networks (Sa et al., 2016). In this study, fruit detection was performed using images from color (RGB) and near-infrared (NIR) imaging modalities. Early and late fusion methods were investigated to combine multimodal (RGB and NIR) information. (Yaman et al., 2022) used deep learning and feature extraction methods to detect plant diseases in leaf images. In this study,

disease detection was performed using deep learning and feature selection methods applied to plant leaf images. In their review, (Liakos et al., 2018) examined various areas where machine learning was used in agricultural production systems. In this review, agricultural yield estimation, disease detection, weed detection, plant quality and species recognition issues were focused on. (Zang et al., 2018) developed a deep learning-based tomato harvesting robot in their research. In their study, they achieved an average accuracy of 91.9% with a prediction time of less than 0.01 seconds. In another study, (Luna et al., 2019) studied the size classification of tomato fruit with deep learning. With the deep learning approach they created, they found a training verification-test accuracy of 82.31%- 78.21%- 55.97% and low performances with 48.17%-41.44%-37.64% for VGG16. (Mu et al., 2020) determined the presence of tomato on seedlings and the degree of maturity with the help of deep learning in their study. (Afonso et al., 2020) used deep learning method in the detection system of tomato fruits grown in greenhouse.

Based on the examples given, deep learning methods can be effectively used in visual recognition tasks such as fruit classification. Deep residual learning and deep neural networks achieve high accuracy in such tasks due to their ability to represent deeper and more complex structures. These methods generally give successful results in large-scale image recognition tasks. This is because the model is capable of solving complex image recognition and classification problems. The aim of this study was to classify persimmon images for use in robotic harvesting systems using the YOLOv8L architecture. Classification is the process of categorizing a dataset into specific categories. For persimmon, ripeness color is considered as a class. Machine learning algorithms were used for classification. Machine learning is a branch

of artificial intelligence that enables to perform certain tasks automatically by learning data as a result of training.

2. Material and Method

2.1. Persimmon

Persimmon is a fruit species cultivated mostly in subtropical climatic conditions in the world. Kaki (*Diospyros kaki* L.), commonly known as persimmon because it was distributed to other regions through Trabzon, it spread from China, which was considered its homeland, first to Japan and then to other countries (Tuzcu et al., 2000; Özkan et al., 2013). Persimmon is an important fruit in human nutrition because it is rich in carbohydrates, pectin, tannins, vitamins A, C and E, and its leaves are used as green tea in some countries. On the other hand, it is a fruit that many people consume with pleasure with its color, appearance and different taste (Kuzucu et al., 2004).

In persimmon, the color of the fruit peel usually changes to greenish yellow, orange yellow, orange red at harvest time. At the time of physiological maturity, the shell colors of the varieties become orange, dark orange, red orange, red (Onur, 1990).

Since persimmon fruits show climacteric characteristics, ethylene production increases in the affected areas, and these parts of the fruit ripen earlier and reach eating maturity. In fruits with different ripening times, losses increase and quality and market values decrease. Fruits should be protected against all kinds of damage such as crushing, scratching, bruising and injury during and after harvest. If this is not done, a faster maturation occurs in the injured areas and this prevents homogeneous maturation and increases product and quality losses. Besides the fresh consumption of persimmons, their use by drying and in various food areas should be encouraged. This makes important contributions to the evaluation of fruits, especially in small production areas where

many different varieties are grown (Özcan, 2018).

2.2. Preparation of the data set

While preparing the dataset of persimmon fruit, which was targeted for object detection and analysis within the project, many persimmon images taken in the vineyard during harvesting and growth

were used. Among the images obtained within the scope of this study, the images that we could not evaluate were eliminated. We identified 123 images that would be reliable for our object identification study. In addition, there may be more than one persimmon in each image. In Figure 1, 3 of the photos used during the training are shown.



Figure 1. Photos used in the training set

2.2.1. Labelling

In order for an object detection model to be able to train on a dataset, the objects targeted to be detected must be labeled/signed in the dataset to be trained. For this reason, the parts containing the persimmon image in each of the 123 images should be marked with the bounding box area and assigned to the "PERSIMMON" class, which is the object class it belongs to. There are many programs, websites and utilities available in the open source communities

for image labeling. One of these tools is Roboflow, a popular program that is frequently used in object detection projects. Roboflow is a website that provides all the tools needed to transform raw images into a specially trained computer vision model and distribute it for use in applications, as well as to perform field selections, markings and class labeling on images. This marking and labeling is easily done through the graphical user interface of the website. Label screen is shown in Figure 2.

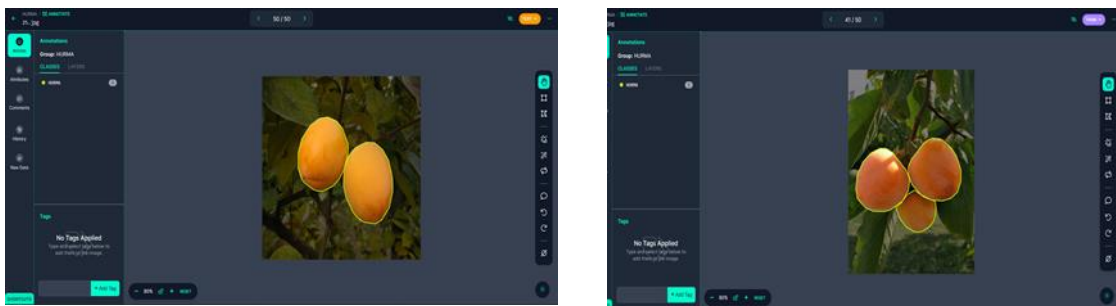


Figure 2. Labeling screen

The general functions of the program are performed by using the left and right parts of the opened visual interface. The marking of the opened image is done with the marking tool activated with the "Polygon Tool" option from the right menu. After determining the area of the object related to the tool, the class name of the specified object should be labeled. Within the scope of the project, these class names were determined as "Permission". Since the training of the relevant object detection model will be done with these images, the selection process must be done correctly including the object. After the selection and labeling process, the output format of the markings should be selected from the "generate" tab in the main menu. Within the scope of the project, the YOLOv8 option, suitable for the model to be used, was selected. After all these processes, the image can be processed and saved again automatically.

2.2.2. Training model selection

This project we carried out, YOLOv8, the latest version of the YOLO model family developed with the CNN method, was used. The YOLOv8 model of the YOLO model family, which has a significant advantage over models using a two-stage network similar to RCNN, was preferred because it provided advantages in terms of accuracy values and speed ratio to versions developed before it. As explained in detail in the upper sections, the YOLOv8 model also contains models within itself. The YOLOv8L model was preferred for deep learning training.

2.2.3. Initiation of training

In order to start the training of the model that will perform persimmon detection, the location of the YOLOv8L model on the computer was visited and a Python executable editor was opened there. The program, which is in the main directory and provides the YOLOv8L training, was

checked to be run. The execution of this Python program can be customized with various parameters.

Within the project, the parameters and regulations in the code written below were preferred for persimmon.

```
!yolo task=detect mode=train  
model=yolov8L.pt  
data={dataset.location}/data.yaml  
epochs=102 imgsz=640 plots=True
```

--task: Indicates which task the model will perform.

--mode: Indicates the mode in which the model will be used.

--imgsz: The pixel size at which the images to be trained will be reduced by the YOLOv8L model. Its default value is determined as 640x640, and it was chosen in this way here as well.

--epochs: The number of times all training data is shown to the trained network and the weights are updated while training the model.

--data: The path to the .yaml file containing the general path and class information of the file containing the dataset.

--model: The location of the weight file containing the training coefficients to be used in training the model.

--plots: Opens visualizing of training and validation metrics during the training process. Creates graphs to track the performance of the model during training.

As a result of running this line of code correctly, the training process of the model has started. The program first checks the YOLOv8L files and checks for any update status. Then, the training process is carried out during the determined number of cycles (epoch).

3. Research Findings

Examination of the results of Size: 640x640, Batch: 16, Epoch: 102, Algorithm: YOLOv8L algorithm according to error matrix metrics.

Detailed statistics generated by adding a column at the end of each epoch, resulting from the completion of the training of the models are given in Table 1 (For summarization, only the first 3 and last 3 epochs from the training of each model are shown).

Table 1: 640x640, Batch: 16, Epoch: 102, Algorithm: YOLOv8L

epoch	0	1	2		99	100	101
train/box_loss	0.91511	0.77347	0.68087		0.29944	0.29536	0.30488
train/dfll_loss	1.204	1.0911	1.0237		0.83512	0.8186	0.8256
train/cls_loss	3.2369	1.3051	0.82784		0.18854	0.1888	0.19037
metrics/precision	0.51465	0.67321	0.75394		0.778	0.71794	0.71491
metrics/recall	0.46903	0.66372	0.73215		0.73451	0.81092	0.79892
metrics/mAP_0.5	0.46985	0.71931	0.77345		0.84974	0.84749	0.84643
metrics/mAP_0.5:0.95	0.35037	0.60709	0.66654		0.76861	0.77063	0.76697
val/box_loss	0.73747	0.61486	0.57854		0.48712	0.48591	0.48871
val/dfll_loss	1.0897	1.0263	0.99916		0.95516	0.95272	0.95354
val/cls_loss	1.8329	1.0255	0.83184		0.71141	0.71098	0.70642
lr/pg0	0.0937	0.086485	0.079255		0.00048824	0.00039118	0.00029412
lr/pg1	0.0007	0.001485	0.0022554		0.00048824	0.00039118	0.00029412
lr/pg2	0.0007	0.001485	0.0022554		0.00048824	0.00039118	0.00029412

The analyzed dataset contains a set of metrics and parameters detailing the training process of the YOLOv8 Large model. The dataset provides loss values, learning rates and performance metrics for each epoch of the model.

A set of loss functions including bounding box regressions (box loss), class loss (cls_loss) and direction feature learning (dfll_loss) were used to classify persimmon during training.

Box loss is often used for bounding box regressions and allows the model to correctly enclose objects with bounding boxes.

Cls loss is often used for object classification and allows the model to correctly classify each object.

Dfl loss is used for direction feature learning (dfl), a unique characteristics of YOLOv8. DFL is used to learn and predict the direction of the object.

The loss functions work together for the model to correctly detect and classify objects. By minimizing these loss functions, the model continuously improves during the training process. Different values of learning rates (lr/pg0, lr/pg1, lr/pg2) were also used in the training process of the model. The learning rates used generally decrease during the training process. This allows the model to learn fast at the beginning and then slow down as it approaches the optimum. Precision, Recall, mAP50 and mAP50-95 metric sets were used to measure the performance of the model. These metrics indicate how accurately the model detects and classifies

objects. This information allows us to better understand the model's training process and provides ideas on which steps can be taken to evaluate and improve the model's performance. This is important to increase generalization ability of the model and perform better on new data.

3.1. Box loss, class loss ve dfl loss

The model, which we prepared with YOLOv8 Large, made a noticeable progress in the training process by making use of loss functions such as `box_loss`, `cls_loss` and `dfl_loss`. Progress can be determined by observing the falling trend of each loss during epochs. `Box_loss` measures the model's ability to accurately predict boxes that bound objects. `cls_loss` measures the model's ability to assign objects to the correct classes. `DFL_loss` measures how

well the model has learned the details and characteristics of the objects.

It was seen that the model consistently reduced these losses throughout the training process. This showed that the learning capacity and capability of the model was impressive. According to the measured values, it appeared that the model was getting better and better at predicting the bounding boxes, classes and characteristics of the objects. According to the obtained values, it showed that the model was successful in learning the characteristics and classes of persimmon.

In Figures 3, 4, 5 below, it is seen that how the `box_loss`, `cls_loss` and `dfl_loss` values changed throughout the epochs. The development of the model in the training process is shown in the graphs.

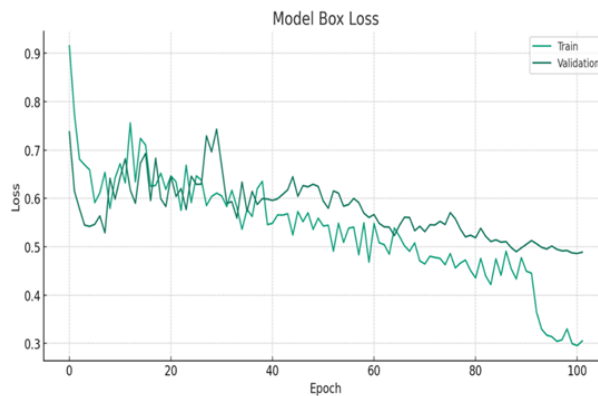


Figure 3. Model Box Loss graph



Figure 4. Model DFL Loss graph

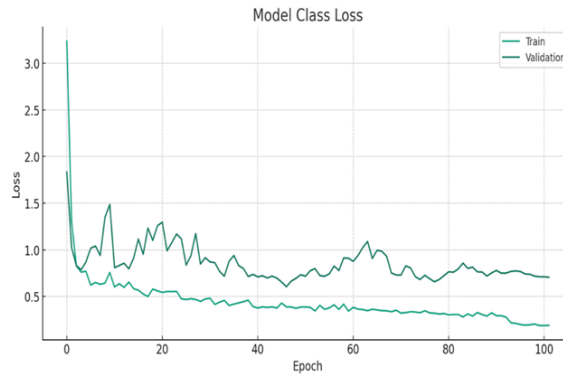


Figure 5. Model Class Loss graph

In the graphs above, it can be seen that `box_loss`, `cls_loss` and `dfl_loss` generally decreased during the training process of the model. According to the graphic values, it was understood that the model recognized the persimmon better at each epoch and better modeled its relationship with the target variables.

`Box_loss` graphics show that the model's ability to predict the bounding boxes of objects has improved. When the `box_loss` graph was examined, it was seen that there was a significant improvement in our ability to locate the persimmon in the images. The fact that the training and validation `box_loss` values were close to each other indicated that the model learned this characteristic consistently in the training and validation data.

The `cls_loss` graphics show that the model's ability to assign objects to the correct classes has improved. The closeness of the training and validation `cls_loss` values indicates that the classification ability of the model is consistent in the training and validation data. This indicated a significant improvement in the model's ability to learn persimmon classes.

`DFL_loss` graphics show how well the model has learned the details and characteristics of the objects. The fact that the training and validation `dfl_loss` values are close to each other indicates that the

model has learned this feature consistently in the training and validation data. This indicated a significant improvement in the model's ability to learn the details and characteristics of persimmon.

From the examination of the results, it was understood that the model made a stable improvement in the training process and the ability to recognize persimmon improved. This showed that the model was successful in this task and can be a valuable tool for similar projects in the future.

3.2. Learning rates

The learning rates of our model (`lr/pg0`, `lr/pg1`, `lr/pg2`) were carefully managed throughout the training process. The learning rate is a hyperparameter that controls the learning rate of the model. A higher learning rate allows the model to learn faster, but this may cause the model to "omit" the optimal solution or cause the training process to be unstable. A lower learning rate provides the model learn more slowly, but this generally provides a more stable training process and better generalization ability.

Learning rate graphs show that the model's learning rates are reduced throughout the training process. Here it is done using a strategy known as "learning rate decay". This strategy helps the model balance faster training progress and finer adjustments later in the training process.

This enables us to approach the optimum solution faster and more effectively, especially with large and complex models.

The graph in Figure 6 below shows how learning rates changed throughout epochs. The Performance Metrics graph shows how the model's learning strategy and learning

rate hyperparameter are managed. This graph shows that the model successfully managed the learning rate and thus made steady progress in the training process; the model was successful in this task and can be a valuable tool for similar projects in the future.

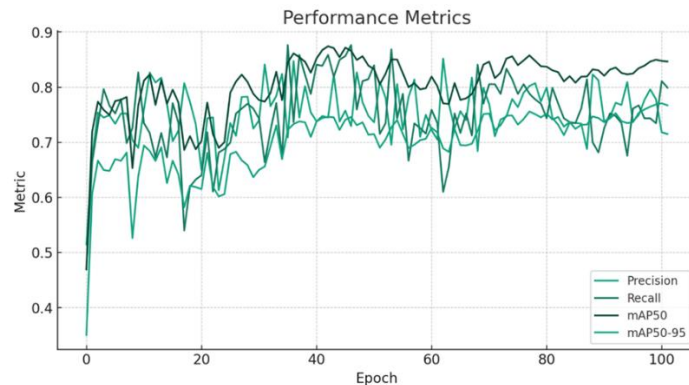


Figure 6. Performance Metrics graph

The Performance Metrics graph shows how the model's learning rates ($lr/pg0$, $lr/pg1$, $lr/pg2$) decrease during the training process. The downtrend in the graph indicates that the learning rate hyperparameter is effectively managed. This indicates that it starts with a fast learning rate at the beginning and then slows down as it approaches the optimum. This strategy allows the model to learn quickly and make precise adjustments in later stages.

In particular, this learning rate strategy is very significant for our YOLOv8 Large model, which is a large and complex model. Due to the strategy used and the size and complexity of the model, effective management of the learning rate provides that the model learns quickly and effectively in the training process. This helps the model to learn the characteristics and classes of persimmon quickly and effectively. Besides, this learning rate strategy increases the generalization ability of the model. Decreasing the learning rate over time makes the model more robust to

new data and exhibits better generalization ability. This improves the model's performance on new and unprecedented data. Consequently, the learning rate strategy of our model played an important role in the success of the training process. This strategy allows the model to learn quickly and increase its generalization ability. It helps the model to learn the characteristics and classes of persimmon effectively and show high performance against new data.

3.3. Performance metrics

Performance metrics are used to measure the prediction performance of the model. YOLOv8 Large showed a significant improvement during the training process of our model. These metrics include Precision, Recall, mAP50 and mAP50-95. Precision points out the ratio of positive samples that the model correctly identifies out of all the samples it identifies as positive. A high precision value indicates that the model keeps false positives (samples that the model identifies as positive but are actually

negative) to a minimum. During the training process, we see that the precision value of the model generally increases. This showed that the model was becoming more and more successful in correct classification of persimmon. Recall (Sensitivity or Precision) points out how many of the true positive samples are correctly identified. A high sensitivity value indicates that the model does not miss positive samples. The sensitivity of our model generally increased during the training process. This indicated that the model became more sensitive in the classification of persimmon. mAP50 and mAP50-95 are metrics that measure the overall performance of the model. mAP50

represents the Mean Average Precision value for the estimates with an Intersection over Union (IoU) value greater than 0.50. mAP50-95 represents the Average Sensitivity value for estimates where the IoU value is between 0.50 and 0.95. The mAP50 and mAP50-95 values of our model showed an overall improvement throughout the training process. This indicated that the overall performance of the model increased.

The graph below shows how Precision, Recall, mAP50 and mAP50-95 values change during the epochs. The graph shows the performance of the model during the training process more clearly.

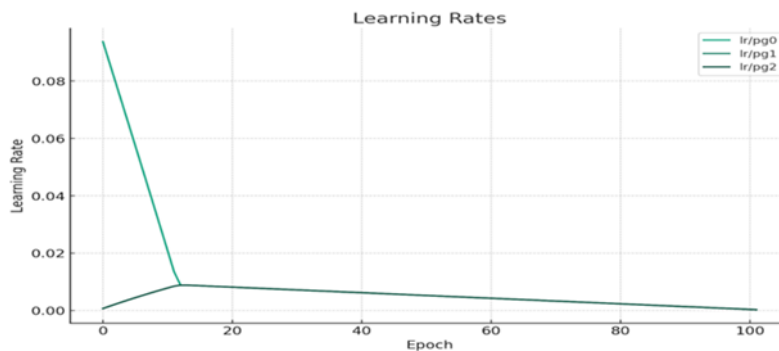


Figure 7. Learning Rates graph

As we can see in the graph in Figure 7 above, the model's performance metrics (Precision, Recall, mAP50, mAP50-95) generally increased throughout the training process. This increase indicated that the model was becoming increasingly successful in recognizing and classifying persimmon. In particular, the Precision metric indicates that the model reduces the number of false positive predictions. The Precision metric indicates a significant improvement in the model's ability to classify objects. The Recall (sensitivity) metric also showed a similar improvement, indicating that the model was able to correctly classify positive samples without missing them. The mAP50 and mAP50-95 metrics measure the overall performance of

the model. The values of these metrics indicate that the overall performance of the model has improved throughout the training process. The mAP50 value represents the Mean Average Precision value for estimates where the Intersection over Union (IoU) value is greater than 0.50. The mAP50-95 value represents the Mean Average Precision value for estimates where the IoU value is between 0.50 and 0.95. The increase in these metrics indicates an overall improvement in the accuracy and reliability of the model's predictions.

In consideration of these parameters, the studied model was successful in the recognition and classification of persimmon plant based on performance metrics. The rise of these metrics indicated that the

model improved in the recognition and classification of the persimmon plant. It showed that our model was effective in this task and can be a valuable tool for similar projects in the future.

3.4. F1 score

The F1 score is a widely used metric to measure the classification performance of a model. The F1 score is calculated as the harmonic mean of the Precision and Recall metrics. That is, taking into account both metrics, it allows us to evaluate the classification performance of the model with an overall measure. The F1 score is especially valuable when working on unstable datasets (where there are many more samples of one class than other classes). The F1 score takes a value between 0 and 1, with 1 indicating the best performance and 0 the worst performance. A high F1 score indicates that the model handles well the cases where both precision and sensitivity are high. This indicates that the model both makes its positive predictions accurately (high precision) and does not miss most of the true positive examples (high sensitivity).

Since there is no column related to the F1 score in the dataset, we can calculate the F1 score from the Precision and Recall columns. The formula for the F1 score is as follows:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Using the formula above, we can calculate the model's F1 score and evaluate the model's performance with this metric. This provides an overall evaluation of the model's classification performance. We can also compare the training and validation F1 scores to check if the model is overfitting. We can also examine how the learning rates (lr/pg0, lr/pg1, lr/pg2) of the model change during the training process. We see that the learning rates decrease over time, which indicates that the learning rate of the model slows down during the training process. This is the result of learning rate decay, a technique often used to slow the learning rate of the model as it approaches the optimum. These analyzes allow us to evaluate the performance of the model from various perspectives, to determine in which areas the model performs well and need improvement. The F1 score plot indicates that the model showed an overall improvement throughout the training process. The F1 score graph shows that the model showed improvement in both Precision and Recall metrics, which indicated that the model was successful in classifying the persimmon plant. The F1 Score graph is shown in Figure 8.

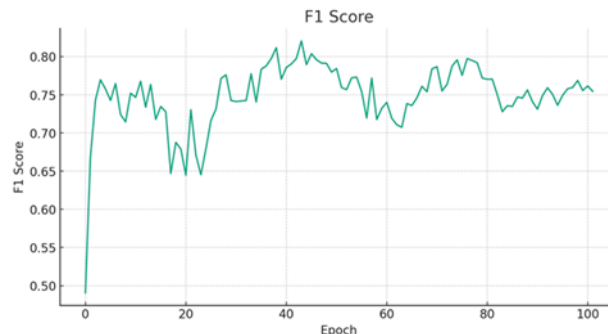


Figure 8. F1 Score graph

An increase in the F1 score indicates a significant improvement in the model's ability to accurately predict positive samples and not miss true positive samples. This indicated a significant improvement in the model's ability to learn the characteristics and classes of persimmon.

Consequently, our model was successful in identifying and classifying the persimmon plant based on the F1 score

metric. The rise of this metric indicated that the model improved in the recognition and classification of persimmon. It was understood that the model we prepared was effective in this task and can be a valuable tool for similar projects in the future.

3.5. Data set used in training

Images from the training set used during the training of the models are shown in Figure 9.



Figure 9. Training set images

3.6. Training result

The "Validation Batch" prediction markings resulting from the training of the

models are shown. The training result is shown in Figure 10.



Figure 10. Training result

4. Discussion and Conclusion

The most important input parameter in robotic harvesting systems is the correct classification of the products. Correct classification according to the given characteristics will increase the success rate of the detection of the product to be harvested robotically. In their deep learning classification study, (Seo et al., 2021) achieved 88.6% detection accuracy in the classification system for tomato greenhouse when completely hidden fruits were not captured. When hidden fruits were excluded, they determined the accuracy rate of the system as 90.2%. In their study on tomato ripeness with deep learning, (Mutha et al., 2021) found the degrees of ripeness as follows. The accuracy value for ripening was 99.2%, 94.34% for immature and 90.23% for damaged product. (Sa et al., 2016) developed a fruit detection system for automatic harvesting using deep learning in their study. In this study, deep neural networks were used to build a fast, accurate and reliable fruit detection system. The system was an important component for agricultural robotic platforms, enabling fruit yield estimation and automatic harvesting. In another study, a deep learning model developed by (He et al., 2016) showed superior performance in image recognition and classification tasks. This model was used to recognize objects on the ImageNet dataset and achieved high accuracy rates. Considering these similar studies, it was determined that the results of our research were in parallel with the results of these researches. The results of this study showed that the Yolov8L model for persimmon was successful. During the training process, the loss value of the model gradually decreased and the accuracy value increased. In the verification phase, the results were at the desired level, the loss value was low and the accuracy value was high.

Declaration of Author Contributions

The authors declare that they have contributed equally to the article. All authors declare that they have seen/read and approved the final version of the article ready for publication.

Declaration of Conflicts of Interest

All authors declare that there is no conflict of interest related to this article.

References

- Afonso, M., Fonteijn, H., Fiorentin, F.S., Lensink, D., Mooij, M., Faber, N., Wehrens, R., 2020. Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in Plant Science*, 11: 571299.
- Aktaş, H., 2022. Classification of pistachios based on outer shell color using deep learning. *Niğde Ömer Halisdemir University Journal of Engineering Sciences*, 11(3): 461–466.
- Arserim, M.A., Usta, A., 2023. Object detection by deep learning approach using images taken from unmanned aerial vehicle. *Dicle University Journal of Engineering*, 14(1): 9-15.
- Aydın, S., Taşyürek, M., Öztürk, C., 2021. Air pollution prediction for Central Anatolia Region and its surroundings with deep learning method. *European Journal of Science and Technology*, Special Issue 29: 168-173.
- Bengio, Y., 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1–27.
- Çetiner, H., Çetiner, I., 2022. Classification of citrus diseases with convolutional neural network based deep learning model. *Bitlis Eren University Journal of Science*, 11(1): 352–368.

- DeLuna, R.G., Dadios, E.P., Bandala, A.A., Vicerra, R.R.P., 2019. Size classification of tomato fruit using thresholding, machine learning and deep learning techniques. *Agrivita*, 41(3): 586–596.
- Deng, L., Yu, D., 2014. Deep learning: methods and applications. *Foundations And Trends® In Signal Processing*, 7(3–4): 197–387.
- Eren, H.A., Okyay, S., Adar, N., 2021. Adoken: Deep learning based decision support software for MRI. *Journal of Engineering Sciences and Design*, 9(2): 406–413.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Conference Proceedings Book, December, pp. 770–778.
- Karasulu, B., Yücalar, F., Borandağ, E., 2022. A hybrid approach based on deep learning for gender recognition using human ear images. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37(3): 1579–1594.
- Kirişoğlu, S., Kotan, B., Kotan, K., 2022. Network traffic classification analysis with multi-layer sensor. *Düzce University Journal of Science Technology*, 10(2): 837–846.
- Kuzucu F.C., Kaynaş, K., 2004. Chemical and Physiological changes in persimmons (*Diospyros kaki* L.) harvested in different time periods. *Bahçe*, 33(1-2): 17 – 25.
- Liakos, K.G., Busato, P., Moshou, D., Pearson, S., Bochtis, D., 2018. Machine learning in agriculture: A review. *Sensors (Basel)*, 18(8):2674.
- Mu, Y., Chen, T.S., Ninomiya, S., Guo, W., 2020. Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors*, 20(10): 2984.
- Mutha, S.A., Shah, A.M., Ahmed, M.Z., 2021. Maturity detection of tomatoes using deep learning. *SN Computer Science*, 2(6): 441.
- Onur, S., 1990. Trabzon hurması. *Derim*, 7(1): 4-46.
- Özcan, M., 2018. The problems and future of persimmon (*Diospyros Kaki* L.) cultivation in Turkey. *Black Sea Journal of Agriculture*, 1(2): 38-43.
- Özkan, H.U., Can, H.Z., 2013. Research on the quality properties of persimmon (*Diospyros Kaki* L.) fruits at different harvest stages. *Journal of Agriculture Faculty of Ege University*, 50(2): 137-144.
- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., Mccool, C., 2016. Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8).
- Seo, D., Cho, B.H., Kim, K., 2012. Development of monitoring robot system for tomato fruits in hydroponic greenhouses. *Agronomy*, 11(11): 2211.
- Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. (<http://arxiv.org/abs/1409.1556>).
- Şentürk, T., Latifoğlu, F., 2023. Deep learning based methods for biomedical image segmentation: A review. *Dicle University Journal of the Institute of Natural and Applied Science*, 12(1): 161–187.
- Tan, Z., Karaköse, M., 2022. Comparative analysis for autonomous path planning approaches based on deep reinforcement learning in dynamic environments. *Journal of Engineering Science of Adiyaman University*, 9(16): 248–262.

Tuzcu, Ö., Yıldırım, B., 2000. Trabzon hurması (*Diospyros kaki*) ve Yetiştiriciliği. Tübitak Tarp Yayınları, Adana.

Yaman, O., Tuncer, T., 2022. Deep feature extraction and machine learning method for leaf disease detection in plants. *Fırat*

University Journal of Engineering Science, 34(1): 123–132.

Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., Wang, M., 2018. Deep learning based improved classification system for designing tomato harvesting robot. *IEEE Access*, 6: 67940-67950

To Cite

Kahya, E., Özdüven, F.F., Ceylan, B.C., 2023. Application of YOLOv8L Deep Learning in Robotic Harvesting of Persimmon (*Diospyros Kaki*). *ISPEC Journal of Agricultural Sciences*, 7(3): 587-601.
DOI: <https://doi.org/10.18016/10.5281/zenodo.8320097>.
