Yuzuncu Yil University Journal of Agricultural Sciences, Volume: 34, Issue: 4, 31.12.2024

Yuzuncu Yil University
Journal of Agricultural Sciences
(Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi)
https://dergipark.org.tr/en/pub/yyutbd

ISSN: 1308-7576 e-ISSN: 1308-7584

Research Article

Use of YOLOv5 Trained Model for Robotic Courgette Harvesting and Efficiency Analysis

Erhan KAHYA*[1]

[1]Tekirdağ Namık Kemal University, Vocational School of Technical Sciences, Department of Electronics and Automation, Control and Automation Technology Programme, Tekirdag, Türkiye

[1]https://orcid.org/0000-0001-7768-9190

*Corresponding author e-mail: ekahya@nku.edu.tr

Article Info

Received: 16.07.2024
Accepted: 22.10.2024
Online published: 15.12.2024
DOI: 10.29133/yyutbd.1517109

Keywords

Courgette,
Deep learning,
Product forecasting,
YOLOv5

Abstract: The utilization of machine learning in vegetable harvesting not only enhances efficiency and precision but also addresses labor shortages and improves overall agricultural productivity. In this study, a machine learning method was developed for harvesting courgette fruit. Courgette is a fruit that can take a long time to select and harvest in the agricultural area where it is grown. The YOLOv5 models (nano, small, medium, and large) were used as a deep learning method. All metric values of the models were analyzed. The most successful model was the one trained with the YOLOv5m algorithm using 20 batches and 160 epochs with 640x640 images. The results of the model scores were analyzed as "metrics/precision", "metrics/recall", "metrics/mAP_0.5" and "metrics/mAP_0.5: 0.95". These metrics are key indicators that measure the recognition success of a model and reflect the performance of the respective model on the validation dataset. The metrics data of the "YOLOv5 medium" model proved to be higher compared to the other models. The measured values were YOLOv5m = size: 640x640, batch: 20, epoch: 160, algorithm: YOLOv5m. It was concluded that "YOLOv5m" is the best recognition model that can be used in robotic courgette harvesting to separate the courgette from the branch.

To Cite: Kahya, E, 2024. Use of YOLOv5 Trained Model for Robotic Courgette Harvesting and Efficiency Analysis. *Yuzuncu Yil University Journal of Agricultural Sciences*, 34(4): 669-689. DOI: https://doi.org/10.29133/yyutbd.1517109

1. Introduction

Agriculture is the most important factor required for our survival since the existence of humanity. Food production and consumption are of great cultural and economic importance. As the world population increases and resources decrease, the need for new and innovative approaches to the agricultural sector is increasingly perceived. Although this rapid population growth suppresses food production, new technologies are also being developed to increase productivity and use resources more sustainably. Deep learning, one of the new technologies, has started to play an important role in the agricultural sector and offers pioneering innovations in many areas of the agricultural sector. This developed technology increases productivity and allows us to use resources more sustainably. In the future, technologies such as deep learning and artificial intelligence will be important tools to meet the food needs of the world's population, while contributing to making agriculture more efficient, profitable, and environmentally friendly. Deep learning is a machine learning method developed to solve complex problems using artificial neural networks and large data sets. The working principle of this method is that it imitates the way the human brain works and analyzes the data using multilayered artificial neural networks. Deep learning has found its place in many different areas of application. It is especially used in areas such as image and sound recognition, natural language processing, automatic driving, medical diagnosis, and robotics (Deng et al., 2014). This method uses a deep neural network to understand the complexity and structure of the data. These layers interact with each other to learn the characteristics of the data and create more complex representations. Deep learning methods perform the learning process using large data sets. These datasets usually consist of labeled data and are used for the deep learning model to predict accurate results. The model analyzes the data, learns the characteristics, and predicts the results. Then, the predictions are compared with the actual results, and the weights are updated to correct the errors of the model. Deep learning has several advantages over other machine learning methods. These are that it has more layers to better understand the complexity of the data and can produce more general and generalizable results using large data sets. Due to these advantages, deep learning is rapidly developing in the field of machine learning and is actively used by many researchers and industry experts (Deng et al., 2014). This method provides a powerful tool to solve complex problems and is expected to become more widespread in the future. Deep learning is a technological application with great potential in many areas in the agricultural sector. With the use of this technology, we can increase productivity, improve product quality, and optimize resource use. Image processing is widely used in autonomous harvesting robots and disease detection. The YOLOv5 model, which is one of the deep learning models, has high accuracy rates in object detection and has the potential to be used in agricultural applications.Redmon et al. (2017) and Redmon et al. (2018) showed in their studies that YOLOv5 models were generally an effective method for object detection and were constantly being improved. Bai et al. (2023) and Du et al. (2022) stated that YOLOv5 models could also be used successfully in non-agricultural applications. These studies showed that the YOLOv5 models were generally an effective method of object detection and could be used in different areas. Zhu et al. (2018) examined the usability of deep learning methods in the classification of courgettes. In this study, courgette images were classified using the AlexNet deep learning model. Experimental results showed that the deep learning method had a higher accuracy rate than other methods. Xiao et al. (2023) examined the use of deep-learning models in courgette harvesting. In their study, they focused on object detection and recognition techniques for courgette harvesting robots based on digital image processing and traditional machine learning techniques. They stated that deep learning models could be used effectively to detect and classify courgettes. In their study, Arad et al. (2020) placed an RGB-D camera on an industrial arm with six degrees of freedom. The robot they designed was tested and verified in current greenhouses. At the end of the trials, an average cycle time of 24 seconds per fruit and a harvest success rate of 61% were achieved under optimal crop conditions. Mao et al. (2020) developed a recognition algorithm for cucumber harvesting robots. In this study, they performed the identification of cucumbers on seedlings using color analysis. As a result of the study, they found the correct recognition rate to be more than 90%, the false recognition rate to be lower than 22% and the true error rate to be more than 4%. Droukas et al. (2023) stated in their research on robotic harvesting systems and enabling technologies, especially the deep learning method would provide great convenience in the autonomous running of robotic systems.Gholipoor et al. (2019) conducted a prediction study by planting the seeds of 692 local genotypes in their study titled Fruit Yield Prediction in Pepper using an artificial neural network. In the study, they found that ANN achieved the highest accuracy as a result of the yield prediction. Wang et al. (2013) added PCNN to the system to solve the grayscale in determining the location of the cucumber. They found the success rate at 82.9%. They stated that successful results could be achieved by using deep learning techniques in the detection of plant diseases. Barman et al. (2024) investigated the potential of deep transfer learning techniques for the early detection of diseases in rice plants. As a result of their study, they found that ResNet and MobileNet models are able to detect diseases with higher accuracy than other architectures. Altınbilek et al. (2022) developed a CNN model to implement two important diseases in paddy rice (rice blast and blight) using deep learning extractions. The developed model classified the diseases with a high accuracy rate of 91.70%. Atalay et al. (2017) showed that plant diseases were detected using deep-learning algorithms from images of plant leaves. In this study, the classification of plant diseases was carried out using deep features obtained from images of plant leaves. Deep learning algorithms can help early diagnosis of plant diseases by automatically extracting features from images of plant leaves. The classification of agricultural products is also an application that can be achieved with deep learning. İmak et al. (2023) extracted features from vine leaf images using deep learning algorithms and classified leaf types using these features.In the agricultural sector, the use of deep learning methods to identify the growth stages of sunflower plants has contributed to improved efficiency and productivity in agriculture by enabling autonomous irrigation, fertilization, and harvesting systems (Karahanlı et al, 2024). In addition, computer vision methods used in precision agriculture are utilized in various areas of the agricultural sector and play a significant role in the agricultural digital transformation. These technologies are used in various ways to increase agricultural productivity, utilize resources more efficiently, and support sustainable agricultural practices. The use of computer vision technologies contributes to the development of more efficient, sustainable, and intelligent practices in the agricultural sector. Images captured by drones and sensors in agricultural fields can be analyzed using computer vision techniques to identify plant species and increase agricultural productivity (Nath, 2024). Computerized image processing techniques are also used for early detection and management of plant diseases. Image analysis and artificial intelligence algorithms can identify disease symptoms on plant leaves and thus prevents the spread of diseases (Wang et al., 2022). Computer vision technologies that use cameras integrated into agricultural machinery can detect plant diseases, enabling early intervention and increased productivity (Kini et al., 2023). Computer vision systems integrated into agricultural robots can automatically recognize fruits and perform harvesting (Rudenko et al., 2023). In addition, the use of computational imaging technologies for weed and pest detection in agricultural areas can enable more precise and effective application of pesticides (Kaldarova et al., 2023). Bati et al. (2023) investigated the performance of the YOLOv5s model for the identification of individual cattle. Their research emphasised that the YOLOv5s model has significantly improved the performance of the model by increasing its effectiveness in automatic cattle identification systems.

Computer vision methods are used to detect and control pests in agricultural areas. Image analysis enables the identification of harmful organisms and targeted interventions against these organisms (Alam et al., 2022). Computer vision technologies are used to monitor and analyze plant growth processes. Characteristics such as plant growth rate, number of leaves, and flowering periods can be monitored using computer vision methods (Štaka et al., 2023). Computer vision techniques are used for estimating the productivity of crops and to manage the timing of harvesting. Image analysis and data processing algorithms are significant tools for determining the quantity and quality of crops (Xu et al., 2023). Computer vision technologies are also used to determine irrigation needs and optimize irrigation systems in agricultural areas. Plant water requirements can be determined for efficient water use through image analysis and data processing methods (Chen et al., 2022). Computer vision systems integrated into agricultural robots can automatically recognize fruits and perform harvesting (Rudenko et al., 2023). In addition, the use of computer vision technologies for weed and pest detection in agricultural areas can enable more precise and effective application of pesticides (Kaldarova et al., 2023).

The common features of the systems performed are deep learning models. The most important and primary input element for robotic harvesting systems to be designed is that the deep learning network finds the product to be harvested on the branch or seedling. Deep learning models are effective for object detection in agricultural applications. These models possess the capability to be utilized in agricultural applications due to their rapid processing capabilities and high accuracy rates. However, factors such as size, diversity, hyperparameters, and running parameters of the data set should be considered to select the most appropriate model for each application. Also, speed performance should be considered in agricultural applications because fast object detection may be more important for some applications.

YOLOv5 is a deep learning framework specifically developed for the tasks of object detection and classification. It utilizes a convolutional neural network (CNN) architecture to recognize and classify objects present in images. In this model, images are first passed through a series of convolution and activation layers, and then the images are scaled to adjust for the different sizes and properties of the objects to be detected. The model uses "estimators" of special structures to predict object bounding boxes in images. Each estimator is designed to detect objects at a specific scale and location. These estimators allow the neural network to detect objects at different scales. Estimators generate probability values for each class and then classify objects into different classes. In this way, multiple objects can be detected in an image, and accurate class labels and bounding box predictions can be obtained for each object. The YOLOv5 model can be implemented in Python using the PyTorch library and can be customized through experimentation using data from experiments. Consequently, the objective of this study was to assess and analyze the effectiveness of a YOLOv5-trained model in the robotic harvesting of courgettes.

2. Material and Methods

2.1. Preparation of the data set

Images captured in both the greenhouse and the field were employed in the development of the courgette dataset for object detection and analysis in this study. Specifically, the research utilized photographs taken during the harvesting and growth phases within a producer greenhouse located in Naip Village, Tekirdağ Province, Türkiye. Figure 1 shows the sample photographs. A total of 120 images were used in this study conducted for object detection. These 120 images contain photographs of more than one courgette.

Figure 1. Examples of photos taken in the producer's greenhouse in Tekirdağ Naip Village (Original).

2.2 Labeling

In order for an object recognition model to be trained effectively on a data set, the objects to be recognized in this data set must be appropriately labeled. In the context of deep learning, the labeling of objects is a fundamental aspect that significantly influences the accuracy and effectiveness of the model. This process is crucial for the training, evaluation, and improvement of deep learning models. The precision and consistency of the labels have a direct impact on the overall performance of the model. High-quality labeling ensures that the training data is reliable so that the model can accurately identify, classify, differentiate, and learn from different objects. In addition, labeled data is crucial for assessing the accuracy of the model's predictions. Labeling objects facilitates the identification of key features and thus improves the effectiveness and efficiency of the model. In this study, each of the 120 images was labeled with bounding boxes around the courgettes and assigned to the object class "courgette". The open-source community offers a wide range of programs, websites, and tools for image labeling. Among these, Roboflow is a notable tool that is often used in object recognition projects. Roboflow offers a comprehensive platform that provides users with important resources to convert raw images into a specially trained computer vision model suitable for various applications. It also simplifies the processes of field selection, labeling, and class assignment of images. The website's graphical user interface greatly simplifies the labeling process, as shown in Figure 2.

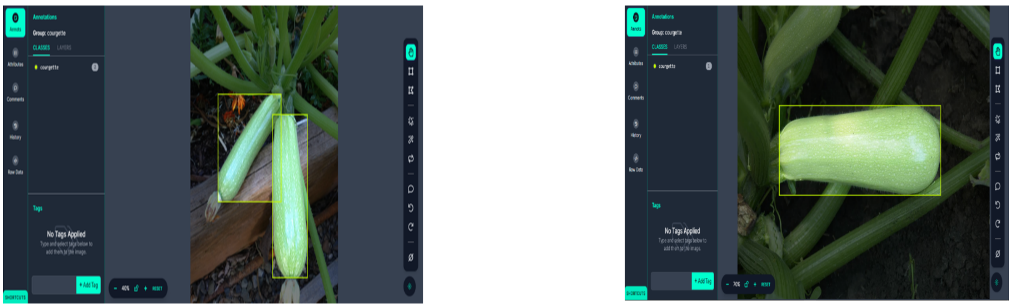


Figure 2. Labeling screen.

The basic operations of the program are carried out via the left and right areas of the displayed visual interface. The image is labeled using the marking tool, which is activated via the “Bounding Box Tool” option in the right-hand menu. Once the bounding box area corresponding to the object is defined, the class name for the identified object must be assigned. In this project, we have chosen the class name “courgette”. As the training of the associated object recognition model is based on these annotated images, it is important that the selection process accurately captures the object. The primary features defined in the training class are based on the color and shape attributes observed in the images during the labeling phase. In the testing phase, the goal is to identify images that most closely match the defined features within the courgette class. After the selection and labeling process, the output format for the annotations was selected from the 'Generate' tab in the main menu. For this project, the YOLO option was selected, which is suitable for the intended model. Once these steps were completed, the images were automatically processed and saved. When you access the text file associated with an image containing the labeled object classes, the system retrieves both the class information and the coordinates of the annotations.

2.3 Selection of training models

In this study, the open-source YOLOv5 family, part of the broader YOLO model series developed using the CNN approach, was selected for implementation. The YOLOv5 model is preferred due to its considerable advantages over two-level network models such as RCNN, especially in terms of accuracy and speed compared to previous versions. As described in the previous sections, the YOLOv5 framework includes several sub-models. The YOLOv5s (small), YOLOv5n (nano), YOLOv5m (medium), and YOLOv5l (large) variants were used for deep learning training. The images from the training set that were used to train the model are shown in Figure 2.

2.4. Initiation of training

To start the training process for the courgette recognition model, the directory containing the YOLOv5 model was opened on the computer and an executable Python editor was started. The script train.py, which is located in the main directory and facilitates the training of YOLOv5, was prepared for execution. This Python script can be customized using various parameters. In this project, the parameters and configurations specified in the following code were selected specifically for the courgette fruit.

```
python train.py --img 640 --batch 20 --epochs 160 --data dataset.yaml --weights yolov5n.pt
python train.py --img 640 --batch 20 --epochs 160 --data dataset.yaml --weights yolov5s.pt
python train.py --img 640 --batch 20 --epochs 160 --data dataset.yaml --weights yolov5m.pt
python train.py --img 640 --batch 20 --epochs 160 --data dataset.yaml --weights yolov5l.pt
```

img: The YOLOv5 model adapts the size of the images for training to a specific pixel size. The default setting is 640x640 pixels, which was also selected for this implementation.

Batch: This parameter defines the number of data point batches that the graphics processing unit (GPU) will process simultaneously during model training.

Epochs: This refers to the total number of times the entire training dataset is presented to the neural network during which the weights of the model are adjusted.

Data: This specifies the path to the “.yaml” file that contains the overall path and class information for the dataset.

Weights: This specifies the location of the weighting file that contains the training coefficients to be used in the training process of the model.

Once these lines of code have been successfully executed, the training process for the model is started. The program first checks the YOLOv5 files and searches for updates. Training is then carried out over the specified number of epochs.

3. Results

To evaluate the performance of the learning results, metrics such as true positive, false positive, false negative, and true negative accuracy values were used. These metrics evaluate the effectiveness of the classification model and include accuracy, precision, recall, and F1 score. The true positive value indicates the correct prediction of the positive class by the model, while the true negative value reflects the accurate prediction of the negative class. Conversely, the false positive value means that the positive class was incorrectly predicted, and the false negative value means that the negative class was incorrectly predicted. Accuracy is a measure of the success of the model and is calculated based on the rate of correct predictions for all samples. Precision indicates the percentage of results that were correctly classified, while recall measures the proportion of true positives that were correctly identified. The F1 score is the harmonic mean of precision and recall. The formulas for these metrics are explained below. Accuracy refers to the proportion of correct classifications or predictions concerning the entire data set. While an accuracy metric close to 1 can be considered an indicator of success, it is insufficient to rely solely on this metric for a comprehensive evaluation.

$$\text{Accuracy} = \frac{\text{TN+TP}}{(\text{TP+FP+TN+FN})} \tag{1}$$

The error rate is the rate of frequency of incorrect classifications/predictions in the problem.

$$\text{Error Rate} = \frac{\text{FN+FP}}{(\text{TP+FP+TN+FN})}\ or\ (1 - \text{Accuracy}) \tag{2}$$

Precision is the rate from the positive predictions made in the problem to the positive ones, in other words, the correct ones.

$$\text{Precision} = \frac{\text{TP}}{(\text{FP+TP})} \tag{3}$$

Recall measures the proportion of actual positive observations that were correctly identified by the model.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \tag{4}$$

F1 score: This metric serves as an alternative to precision and is crucial for the interpretation and understanding of the problem at hand. It represents the harmonic mean of the precision and recall values.

$$\text{F1 Score} = \frac{\text{2xPrecision}}{\text{Precision+Recall}} \tag{5}$$

Examination of the results of YoloV5 algorithms according to error matrix metrics

The analysis graphs of how the performance of YOLOv5 models changed over time over the F1 score are shown in Figure 3. The F1 score serves as a harmonic mean of the precision and sensitivity metrics (recall) for YOLOv5 models. The F1-score of YOLOv5 models shows a general upward trend over time. This showed that the precision and sensitivity values of YOLOv5 models generally improved and therefore their overall performance increased. However, the fact that the F1 score fluctuated throughout the training process indicated that YOLOv5 models experienced performance changes at certain stages or on certain subsets of data. However, the overall upward trend indicated that the model generally improved. Consequently, all YOLOv5 models showed a decrease in loss values and improvement in metrics throughout the training process. However, performance increased as model size increased. The YOLOv5 medium model came to the forefront as the most successful model. The analyzed YOLOv5 medium model showed generally good performance. The F1 score of the YOLOv5 medium model was increasing and it was observed that it reached a certain saturation point.

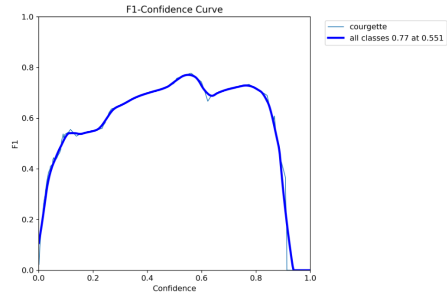
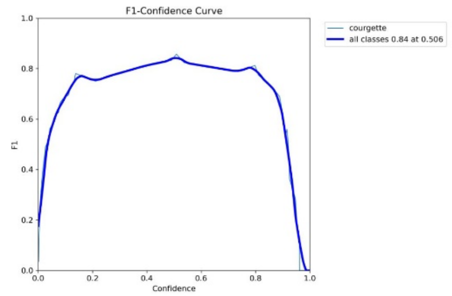
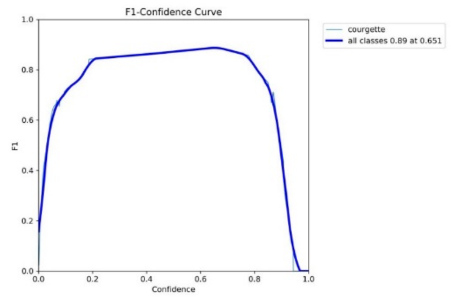
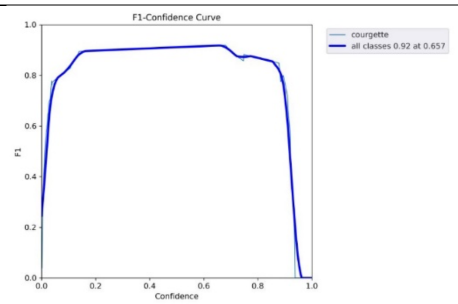
<table>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l</td><td>
</td></tr>
</table>

Figure 3. F1 performance score graphs of Yolov5 models.

Figure 4 shows the analysis graphs representing the precision values obtained in the object recognition tasks with YOLOv5 models. Precision is defined as the ratio of true-positive predictions to the sum of true-positive and false-positive predictions and serves as a critical metric for evaluating the accuracy of the model. When the Precision graph of the YOLOv5 medium model was examined, it was seen that the model's precision score was generally high and showed a slightly increasing trend over time. Precision is a metric that evaluates the classification performance of a model. A high precision score here indicated that most of the examples that the model classified as positive were correct. Fluctuations observed during the training process may indicate that the model misclassified some samples. The generally high level of precision and the slight increase trend indicated that the model generally made correct positive predictions.

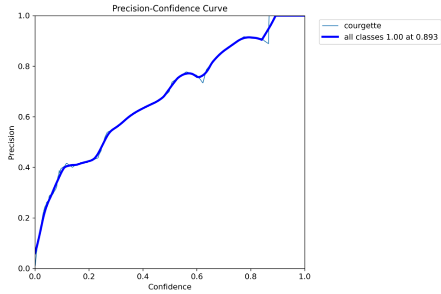
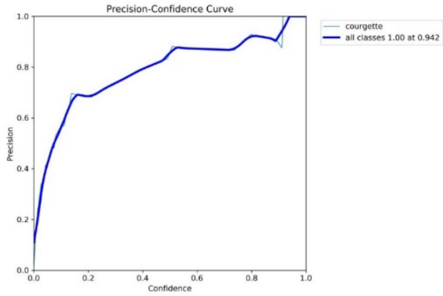
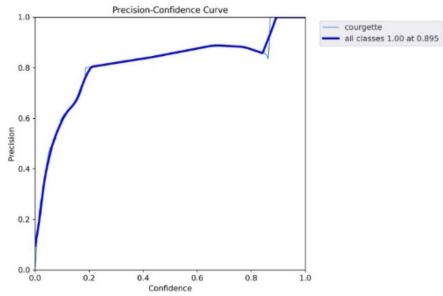
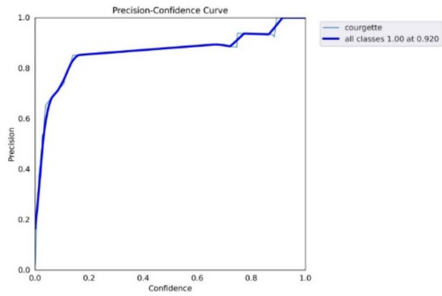
<table>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m</td><td>
</td></tr>
<tr><td>Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l</td><td>
</td></tr>
</table>

Figure 4. Analysis graphs of precision values obtained in object detection of the YOLOv5 models.

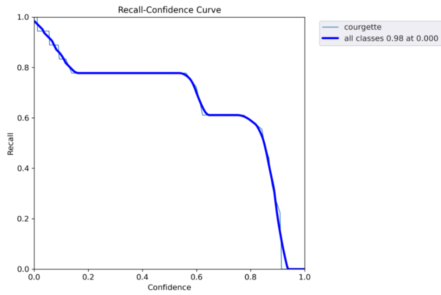
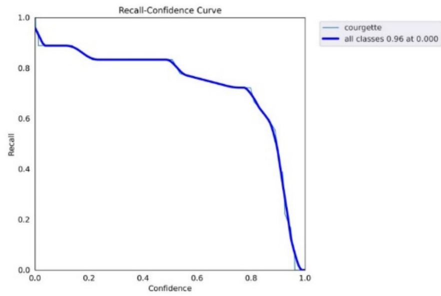
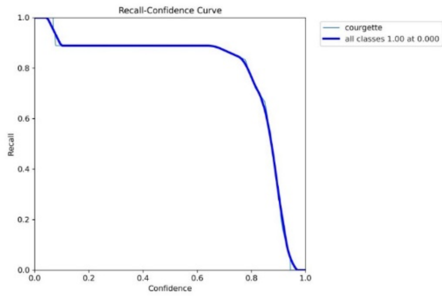
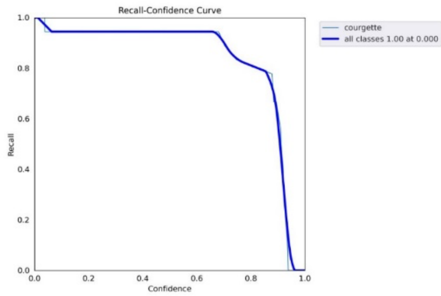
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l	

Figure 5. Analysis graphs of the recall values obtained in the detection of objects in the YOLOv5 models.

The analysis graphs representing the recall values of the object recognition of the YOLOv5 models are shown in Figure 5. The recall value is defined as the ratio between true positive recognition and the sum of true positive and false negative recognition and is used to evaluate the performance accuracy of the model. When the Recall graph of the YOLOv5 medium model was examined, it was seen the Recall score of the model generally increased over time. The recall metric expresses how well the model detects true positive examples. The model’s increasing Recall score indicates that it can detect positive examples more accurately over time. However, the fluctuations seen during the training process indicate that there are changes in the performance of the model on certain stages or subsets of data. It is understood that the ability of the YOLOv5m model to better detect positive examples generally increased and the model started to make more correct positive detections over time. The overall increase in Recall values indicated that the model's inclusion performance improved.

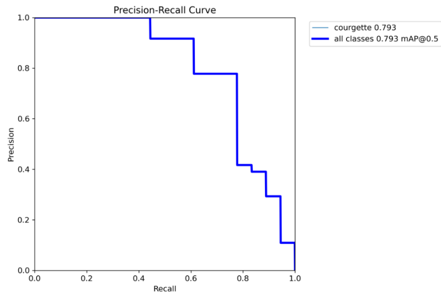
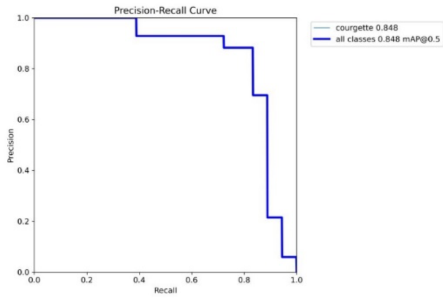
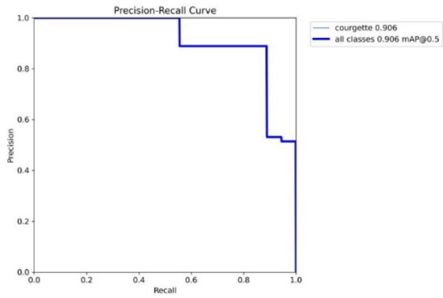
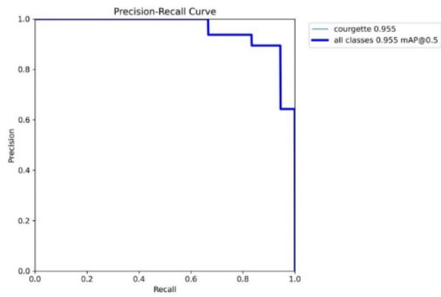
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m	
Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l	

Figure 6. Analysis graphs of precision and recall values obtained in object detection of the YOLOv5 models.

Figure 6 shows a graphical analysis of the precision and recall values associated with the YOLOv5 models in the context of object recognition. Precision and recall are important metrics that evaluate the performance of a model in terms of accuracy and inclusivity. The "Precision-Recall" graph for the YOLOv5m model shows that the model generally reaches a favorable balance. This observation suggests that the model has both a strong ability to produce accurate positive predictions and an effective ability to identify a significant proportion of positive instances. The high values of both metrics imply that the overall performance of the YOLOv5m model is commendable. The analysis concludes that the majority of objects recognized by the model were correct and that it successfully identified a significant number of objects.

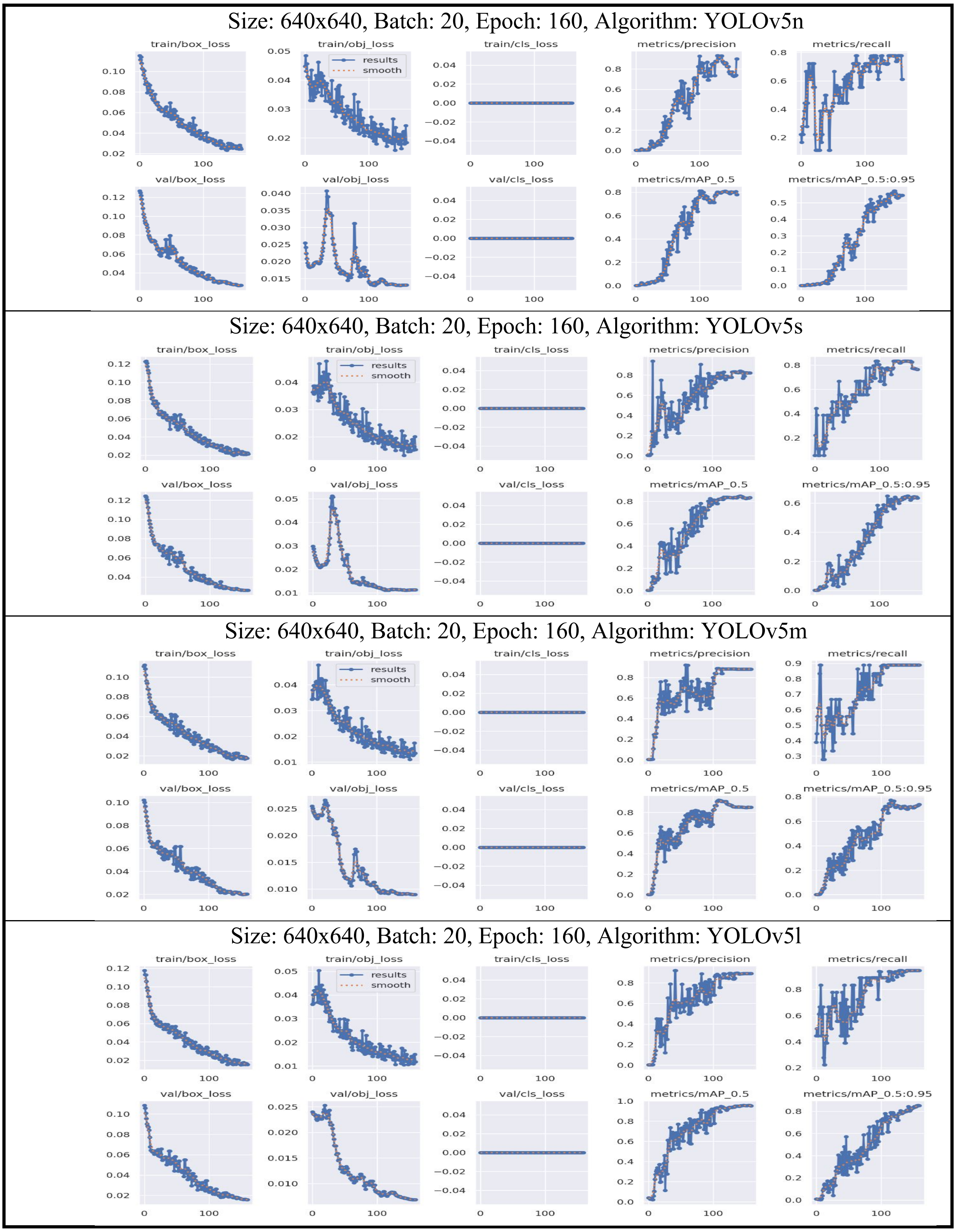


Figure 7. Graphs of error rates and performance values of YOLOv5 models.

Figure 7 shows the graphical representation of the error rates and performance metrics for the YOLOv5 models. In the context of deep learning, the loss function, which is determined in the last layer of a neural network, quantifies the discrepancy between the model's predictions and the actual values. The observations indicate that the error rates of the YOLOv5m model generally decrease over time, suggesting that the model improves its performance during the training process, resulting in predictions that are closer to the actual values. In addition, a decrease in the values of val/box_loss, val/dfl_loss, and val/cls_loss was observed, implying that the YOLOv5m model has strong generalization capabilities in the validation dataset. The results show that the YOLOv5m model consistently evolved in a positive direction, such that errors decreased as training progressed. Detailed statistics obtained by adding a column at the end of each training epoch are shown in Tables 1, 2, 3, and 4 (for clarity, only the first three and last three training epochs are included for each model).

Table 1. Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n

Epoch	**0**	**1**	**2**	**…**	**159**
train/box_loss	0.11534	0.11127	0.11388	...	0.02448
train/obj_loss	0.04471	0.04842	0.04304	…	0.01839
train/cls_loss	0	0	0	…	0
metrics/precision	0.00133	0.001	0.00166	…	0.89887
metrics/recall	0.22222	0.16667	0.27778	…	0.61111
metrics/mAP_0.5	0.00113	0.00079	0.00138	…	0.77906
metrics/mAP_0.5:0.95	0.00023	0.00017	0.00032	…	0.54423
val/box_loss	0.12682	0.12475	0.12191	…	0.02695
val/obj_loss	0.02545	0.02413	0.02240	…	0.01312
val/cls_loss	0	0	0	…	0
x/lr0	0.0991	0.09729	0.09549	…	0.00022
x/lr1	0.0001	0.00029	0.00049	…	0.00022
x/lr2	0.0001	0.00029	0.00049	…	0.00022

Table 2. Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s

Epoch	**0**	**1**	**2**	**…**	**159**
train/box_loss	0.12277	0.12329	0.12112	…	0.021183
train/obj_loss	0.036358	0.038643	0.035678	…	0.015321
train/cls_loss	0	0	0	…	0
metrics/precision	0.0048613	0.003927	0.0026778	…	0.82107
metrics/recall	0.055556	0.22222	0.44444	…	0.76541
metrics/mAP_0.5	0.0037031	0.0036202	0.0039904	…	0.83393
metrics/mAP_0.5:0.95	0.0008226	0.0009349	0.0007792	…	0.63445
val/box_loss	0.12434	0.12307	0.12045	…	0.026205
val/obj_loss	0.029808	0.028709	0.027439	…	0.011324
val/cls_loss	0	0	0	…	0
x/lr0	0.0991	0.097298	0.095494	…	0.0002238
x/lr1	0.0001	0.0002981	0.0004938	…	0.0002238
x/lr2	0.0001	0.0002981	0.0004938	…	0.0002238

Table 3. Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m

Epoch	**0**	**1**	**2**	**…**	**159**
train/box_loss	0.1108	0.1123	0.1082	…	0.0175
train/obj_loss	0.0343	0.0379	0.0346	…	0.0134
train/cls_loss	0	0	0	…	0
metrics/precision	0.0023	0.0026	0.0033	…	0.8813
metrics/recall	0.3888	0.4444	0.5555	…	0.8888
metrics/mAP_0.5	0.0021	0.0023	0.0030	…	0.8493
metrics/mAP_0.5:0.95	0.0004	0.0004	0.0006	…	0.7365
val/box_loss	0.1020	0.1000	0.0966	…	0.0201
val/obj_loss	0.0255	0.0250	0.0246	…	0.0089
val/cls_loss	0	0	0	…	0
x/lr0	0.0991	0.0972	0.0954	…	0.0002
x/lr1	0.0001	0.0003	0.0005	…	0.0002
x/lr2	0.0001	0.0003	0.0005	…	0.0002

Table 4. Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l

Epoch	0	1	2	…	159
train/box_loss	0.1169	0.11307	0.11292	…	0.015369
train/obj_loss	0.036053	0.040098	0.036604	…	0.011766
train/cls_loss	0	0	0	…	0
metrics/precision	0.004063	0.003	0.0026667	…	0.89094
metrics/recall	0.44444	0.5	0.44444	…	0.94444
metrics/mAP_0.5	0.039699	0.039337	0.036866	…	0.9528
metrics/mAP_0.5:0.95	0.009723	0.010574	0.0087236	…	0.85202
val/box_loss	0.10852	0.10577	0.10071	…	0.015674
val/obj_loss	0.024017	0.023684	0.023542	…	0.0068746
val/cls_loss	0	0	0	…	0
x/lr0	0.0991	0.097298	0.095494	…	0.0002238
x/lr1	0.0001	0.0002981	0.0004938	…	0.0002238
x/lr2	0.0001	0.0002981	0.0004938	…	0.0002238

The explanations of the parameters and metrics in the columns in the statistics table are as follows:

Train/box_loss: It is the loss in the box estimation in the training data of the model. Box estimation refers to the ability to predict the positions of objects and their information (such as x and y coordinates, width, and height of area) in an object detection model.

*Train/obj_los*s: It is the loss of object recognition in the model training data. Object recognition refers to the ability of the corresponding object detection model to correctly classify objects in an image.

Train/cls_loss: It is the loss of classification in the model training data. Classification refers to the ability of the object detection model to accurately predict the classes of objects in images. For single-class models, this loss is expected to be 0.

Metrics/precision: It is the accuracy rate that expresses the rate at which the model correctly predicts the prediction of objects in an image.

Metrics/recall: It is the rate that expresses the success of the model in detecting all objects in an image.

Metrics/mAP_0.5: It is the mean precision of the model. Evaluate the performance of the model by comparing the correct and incorrect predictions. This metric has a threshold value of 0,5, which indicates that the accuracy of the model predictions should be greater than half.

Metrics/mAP_0.5:0.95: It is the mean average precision of the model calculated according to a certain accuracy limit (between 0.5 and 0)

Val/box_loss: It is the box prediction loss in the model validation data.

Val/obj_loss: It is the loss of object recognition in the model validation data.

Val_cls_loss: It is the classification loss in the model validation data. This loss is expected to be 0 in models that are validated over a single class.

X/lr0: It is the first learning rate as the optimization parameter of the model. The learning rate determines how drastically the model's weight values updated during training will change. A low learning rate causes the model to learn more slowly, whereas a high learning rate causes the model to learn faster.

X/lr1 and x/lr2: These parameters are the model optimization parameters, similar to the x / lr0 parameter, and are used to update the weight values during training.

The primary differences among these models are rooted in their size and complexity, which in turn influence their speed and overall performance. Below is a comparative analysis based on the training outcomes.

3.1. Yolov5 NANO

The mean precision (mAP@0.5) was determined as 0.77906 in the last epoch, indicating the accuracy of the model in terms of object detection. The precision and recall were 0.89887 and 0.61111, respectively, indicating a relatively high number of true positives but potentially more false negatives. Losses (train/box_loss, train/object_loss, val/box_loss, val/object_loss) decreased significantly throughout the epochs, indicating that the model was learning and improving. In the YOLOv5n model, it is observed that the values of lr0, lr1, and lr2 provide a higher learning rate at the beginning of the learning process, while the learning rate decreases as the epochs progress. Especially, low learning rate in the last epochs will enable the model to produce more accurate results without overfitting.

3.2. Yolov5 SMALL

The mean precision value (mAP @ 0.5) was 0.83393 in the last epoch. It performed better than the Nano model. The precision of the last epoch and the recall of the last epoch were 0.82107 and 0.76541, respectively. This indicated a balanced performance between true positives and false negatives. The model showed a decrease in loss values over time. This indicated that the learning phase of the model improved. In the YOLOv5s model, the decrease in the values of lr0, lr1, and lr2 as the epochs progress, while providing fast learning at the beginning, allows the model to make more fine and careful adjustments later. It plays an important role in improving the overall accuracy and performance of the model.

3.3. Yolov5 MEDIUM

The mean precision (mAP @ 0,5) was 0.84938 in the last epoch. This indicated that it performed better than both the Nano and Small models. The precision and recall of the last epoch were 0,88139 and 0.88889, respectively. The model showed superior performance in identifying true positives and minimizing false positives and negatives. According to these values, it is understood that the loss values will decrease more significantly over time and there will be efficient learning and possibly better generalization. It is seen that the YOLOv5m model is more successful than lr0, lr1, and lr2, especially in the identification of complex and large objects. It is understood that the low learning rate in the last epochs contributes to the high accuracy of the model.

3.4. Yolov5 LARGE

The mean precision (mAP@0,5) was 0.015674 in the last epoch. This value was lower than the other three values of the model epoch. The last epoch precision and recall were 0.89904 and 0.94444 respectively. Given the performance metrics of the other models, it was expected that this model (mAP@0.5) would have higher precision and higher recall. It is acknowledged that the training process may be slower and require more computational resources. In the YOLOv5s model, a decrease in the values of lr0, lr1, and lr2 was observed over the epochs. This initial decrease in values facilitates fast learning and is crucial for improving the overall accuracy and performance of the model. When evaluating the four models, the YOLOv5m model was identified as the model with the highest performance in terms of mAP @ 0.5, lr0, lr1, and lr2 as well as precision and recall among the models with complete datasets. However, the balance between speed and accuracy must be considered when choosing the right model for a particular task.

3.5. Comparison of model algorithms

In the four trained models; the YOLOv5n algorithm, the YOLOv5s algorithm, the YOLOv5m, and the YOLOv5l algorithm are used. The comparison of these algorithms is indicated in Table 5. The metric data of YOLOv5m and the differences of other models from these data are shown in Table 6.

Table 5. Comparison of algorithm values

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.	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	24	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1

Table 6. The metric data of YOLOv5m and the difference of other models from these data

Model	metrics/precision	Difference (Model 3)	Model	metrics/ recall	Difference (Model 3)
YOLOv5m	0.88139		YOLOv5m	0.88889	
YOLOv5n	0.89887	-0.01748	YOLOv5n	0.61111	-0.522221
YOLOv5s	0.82107	0.06032	YOLOv5s	0.76741	0.812349
YOLOv5l	0.89094	-0.00955	YOLOv5l	0.94444	-0.05555
Model	**metrics/mAP_0.5**	**Difference (Model 3)**	**Model**	**metrics/ mAP_0.5:0.95**	**Difference (Model 3)**
YOLOv5m	0.84938		YOLOv5m	0.7365	
YOLOv5n	0.77906	0.07032	YOLOv5n	0.54423	0.19227
YOLOv5s	0.83393	0.01545	YOLOv5s	0.63445	0.01205
YOLOv5l	0.9528	-0.10342	YOLOv5l	0.85202	-0.11552

While the accuracy of correct predictions and the average performance of the models in object recognition are crucial for evaluating their effectiveness, these metrics alone are not sufficient. The number of missed predictions in both the training and validation datasets are also important factors in evaluating model performance. The parameters train/cls_loss and val/cls_loss, which represent the classification losses during training and validation, are particularly important for models that need to recognize a large number of object classes. The analysis of the values of the four parameter columns (train/box_loss, train/obj_loss, val/box_loss, val/obj_loss) in the table with the loss values of the models shows that "YOLOv5m" has the lowest loss values in the training dataset. It is also evident that "YOLOv5m" demonstrates the lowest loss values for both box prediction and object recognition within the validation dataset. A comparison of the training data of the different models is shown in Table 7.

Table 7. Comparative analysis of the models based on training data

Model	train/box_loss	Difference (YOLOv5m)	Model	train/obj_loss	Difference (YOLOv5m)
YOLOv5m	0.17526		YOLOv5m	0.013463	
YOLOv5n	0.024484	0.006958	YOLOv5n	0.018398	-0.004935
YOLOv5s	0.021183	-0.003923	YOLOv5s	0.015321	-0.001858
YOLOv5l	0.15369	0.002157	YOLOv5l	0.011766	0.001697
Model	**val/box_loss**	**Difference (YOLOv5m)**	**Model**	**val/obj_loss**	**Difference (YOLOv5m)**
YOLOv5m	0.020153		YOLOv5m	0.0089622	
YOLOv5n	0.026951	-0.006798	YOLOv5n	0.013128	-0.0041658
YOLOv5s	0.026205	-0.006052	YOLOv5s	0.011324	-0.0023618
YOLOv5l	0.015674	0.004479	YOLOv5l	0.0068746	0.0020876

In conclusion, the optimization parameters of the models (x/lr0-1-2) were analyzed. All models exhibit identical values for these parameters. The specific parameter values are presented in Table 8. The YOLOv5m model, which had the highest number of parameters among the four models, showed the best performance compared to the other models. Starting with high accuracy, recall, and mAP values, YOLOv5m further improved them during training (Table 8). Reduced losses will enable the YOLOv5m model to capture more complex patterns and detect objects with greater accuracy. The YOLOv5m model gives the best result according to higher validation and positive sample capture rate.

Table 8. Optimization parameters of the models

Model	x/lr0-1-2	Difference (YOLOv5m)
YOLOv5m	0.00022375	
YOLOv5n	0.00022375	0
YOLOv5s	0.00022375	0
YOLOv5l	0.00022375	0

3.6. Training result

Screenshots of the training results are given separately for each model in Figures 8, 9, 10, and 11.

Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5n

Figure 8. Validation Batch 'prediction marks' resulting from model training (YOLOv5n) (Original).

Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5s

Figure 9. Validation Batch 'prediction ındicators' derived from the training of the models (YOLOv5s) (Original).

Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5m

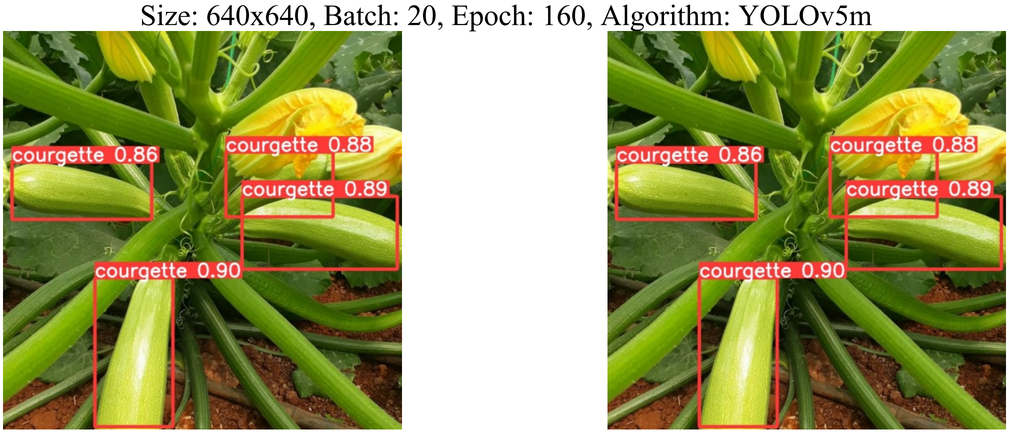


Figure 10. Validation Batch 'prediction marks' resulting from model training (YOLOv5m) (Original).

Size: 640x640, Batch: 20, Epoch: 160, Algorithm: YOLOv5l

Figure 11. Validation Batch ‘prediction indicators’ derived from the training of the models (YOLOv5l) (Original).

4. Discussion

The integration of deep learning algorithms into agricultural harvesting systems is expected to bring significant benefits to agricultural productivity. By using these algorithms, greater accuracy and efficiency can be achieved in the detection and harvesting of agricultural products. In addition, the use of these technologies has the potential to improve occupational health and safety for agricultural workers. The effectiveness of these algorithms in harvesting systems is also underpinned by other studies in the field of deep learning. For example, Fountas et al. (2020) demonstrated the effectiveness of deep learning algorithms in their research focused on the use of agricultural robots. These studies demonstrated that the proficiency of agricultural robots in detecting and harvesting crops was enhanced through the implementation of deep learning algorithms.

The YOLOv5 model was successfully used in other studies. Soeb et al. (2023) used the YOLOv7 deep learning model to detect leaf diseases of green tea. The detection and identification results for the YOLOv7 approach were validated with significant statistical metrics such as detection precision, precision, recall, mAP value, and F1 score, resulting in 97.3%, 96.7%, 96.4%, 98.2%, and 0.965, respectively. Hong et al. (2023) used the YOLOv5 model to allow harvesting robots to recognize fruits under different conditions for asparagus. With this study, they showed that the YOLOv5 model was effective in fruit recognition and harvesting applications. Roshanianfard et al. (2022) evaluated the performance of a robotic end effector for courgette harvesting. This study assessed the end-effector parameters, including precision, repeatability, damage rate, and the area suitable for harvesting.Such performance evaluations are important for the development and optimization of robotic harvesting systems. Darwin et al. (2021) stated that plant diseases can affect agricultural yields and deep learning classifier models can be used in disease diagnosis. Elavarasan and Vincent (2020) introduced a deep reinforcement learning model aimed at predicting crop yields. In this model, they used previous climate data, soil information, and crop management practices as input characteristics. Shin et al. (2020) proposed a deep reinforcement learning model for crop yield prediction in sustainable agricultural practices. The model uses a combination of deep learning and reinforcement learning techniques to analyze various environmental factors and accurately predict crop yield. The proposed model was concluded to perform better than traditional methods and provided valuable information to optimize agricultural practices. This study aimed to identify the most effective detection model suitable for harvesting courgette fruit. Rivera Zarate (2023) focused on the use of LiDAR technology in agriculture and metric estimations. This technology is used to make predictions on topics such as plant health, tree height, tree inventory, LAI estimation, soil properties, and crop yield. Rai et al. (2023) created an open-source dataset in multiple formats for real-time weed detection in agriculture. The dataset consists of 3,975 images of five different weed species commonly found in North Dakota. The dataset is intended to have more efficient and sustainable weed detection and herbicide application in agricultural practices. Luo et al. (2023) have investigated the application of computer vision technologies in agriculture in urban and controlled environment agriculture. They also emphasized that artificial intelligence technologies, deep learning models such as ResNet and MobileNet, and object detection models such as Faster R-CNN, Mask R-CNN, and YOLO can have great potential in the management of crops, animals, and plants. (Zualkernan et al., 2023) stated that traditional machine learning techniques, CNNs, and transformers can be used for simple classification problems, and emphasized that CNNs are the best choice. They achieved 82.10% accuracy in sugar cane plant counting Lu et al. (2023) integrated YOLOv8 technology into the field of phytology. They aimed to improve the detection of small objects with simple and effective optimizations. They achieved distinguishing small objects in UAV images using their YOLOv8-UAV approach. Palacios et al. (2023) developed a new algorithm using computer vision and machine learning for early yield prediction for different grape varieties. As a result of their study, they were able to develop an automatic method for grape yield prediction using computer vision and machine learning and obtained successful results in different grape varieties. Sapkota et al. (2023) proposed an image processing algorithm for corn row identification and classified the remaining vegetation as weeds after identifying the corn rows. Using their image processing algorithm, they created a weed map and sprayed accordingly. As a result, they achieved a 26.2% savings in pesticides compared to existing methods. Ubaid and Javaid (2024) performed feature extraction using VGG-16 and Inception-v3 modules using a model trained on a dataset obtained from a sugarcane field. Jaramillo-Hernández et al. (2024) have addressed the use of machine vision techniques in low-cost devices and how these techniques can be used to increase productivity in precision agriculture (Jaramillo-Hernández et al., 2024). The researchers set out to design, implement, and execute a methodology that integrates computer vision and artificial intelligence and incorporates object detection and integrated depth estimation methods. In this research, the Depth Object Detector (DOD) was trained with the Microsoft Common Objects in Context dataset and the MinneApple dataset. Similarly, Punithavathi et al. (2023) aimed to develop a model for crop and weed identification in precision agriculture using methods based on computer vision and deep learning. Their research led to the development of a model for weed detection and classification using multiscale Faster R-CNN for object detection and optimal extreme learning machine (ELM) processes for weed classification (Punithavathi et al., 2023).The current study seeks to identify the most effective detection model suitable for courgette harvesting.The YOLOv5 model was successfully used in other studies. It also showed the potential of using robotic systems in courgette harvesting. A range of metrics was utilized to evaluate the performance of robotic systems employed in courgette harvesting. These performance assessments are essential for the enhancement and optimization of robotic harvesting technologies.

5. Conclusions

The accuracy of object recognition in training and validation images was evaluated using a dataset created with YOLOv5 models. An examination of the metric data and accuracy prediction rates, which indicate the model's success in object recognition, validated the effectiveness of the training outcomes. By examining the discrepancies between the training and validation datasets concerning loss metrics, along with assessing the loss values in the validation set for a more comprehensive understanding, it was determined that the model designated as ‘YOLOv5m’ exhibited the following parameters:

python train.py --img 640 --batch 20 --epochs 160 --data dataset.yaml --weights yolov5m.pt.

A deeper analysis of the metric data and prediction accuracy revealed that the training outcomes for the “YOLOv5 medium” model surpassed those of the other models. The enhanced accuracy of the YOLOv5m model, in comparison to its counterparts, played a crucial role in its overall effectiveness. In addition, the YOLOv5m model showed a balanced performance and achieved commendable values for mean average precision (mAP) and recall. The initial learning rates lr0, lr1, and lr2 were the same for all models and were gradually reduced over the epochs. This approach allowed the models to learn quickly in the initial phase and then refine their learning through more accurate adjustments. Based on the analysis of these values, the YOLOv5m model was found to be the most effective choice for robotic systems, especially when identifying complex and large objects.

Ethical Statement

Ethical approval is not required for this study because given that only plant photops were involved.

Conflict of Interest

The author declares that there are no conflicts of interest.

Funding Statement

This research received no external funding.

Author Contributions

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

Acknowledgements

The author acknowledges the scientific and technical supports of the University of Tekirdag Namık Kemal University, Türkiye for this research work.

References

Alam, M. S., Alam, M., Tufail, M., Khan, M. U., Güneş, A., Salah, B., Nasir, F. E., Saleem, W., & Khan, M. T. (2022). TobSet: A new tobacco crop and weeds image dataset and its utilization for vision-based spraying by agricultural robots. *Applied Sciences*, *12*(3), 1308. https://doi.org/10.3390/app12031308

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., & Tielen, T. (2020). Development of a sweet pepper harvesting robot. *Journal of Field Robotics*, *37*(6), 1027-1039. https://doi.org/10.1002/rob.21937

Atalay, M., & Çelik, E. (2017). Artificial intelligence and machine learning applications in big data analysis. *Mehmet Akif Ersoy University Social Sciences Institute Journal*, *9*(22), 155-172. https://doi.org/https://doi.org/10.20875/makusobed.309727

Altınbilek, H. F., & Kızıl, Ü. (2022). Identification of Paddy Rice Diseases Using Deep Convolutional Neural Networks. *Yuzuncu Yıl University Journal of Agricultural Sciences, 32*(4), 705-713. https://doi.org/10.29133/yyutbd.1140911

Bai, W., Zhao, J., Dai, C., Zhang, H., Zhao, L., Ji, Z., & Ganchev, I. (2023). Two novel models for traffic sign detection based on YOLOv5s. *Axioms*, *12*(2), 160. https://doi.org/10.3390/axioms12020160

Barman, U., Das, D., Sonowal, G., Dutta, M. (2024). Innovative Approaches to Rice (Oryza sativa) Crop Health: A Comprehensive Analysis of Deep Transfer Learning for Early Disease Detection. *Yuzuncu Yıl University Journal of Agricultural Sciences, 34*(2), 314-322. https://doi.org/10.29133/yyutbd.1402821

Bati, C. T., & Ser, G. (2023). Effects of Data Augmentation Methods on YOLO v5s: Application of Deep Learning with Pytorch for Individual Cattle Identification. *Yuzuncu Yıl University Journal of Agricultural Sciences, 33*(3), 363-376. https://doi.org/10.29133/yyutbd.1246901

Chen, W., Lu, S., Liu, B., Chen, M., Li, G., & Qian, T. (2022). CitrusYOLO: A algorithm for citrus detection under orchard environment based on YOLOv4. *Multimedia Tools and Applications*, *81*, 31363–31389.https://doi.org/10.1007/s11042-022-12687-5

Darwin, B., Dharmaraj, P., Prince, S., Popescu, D. E., & Hemanth, D. J. (2021). Recognition of bloom/yield in crop images using deep learning models for smart agriculture: A review. *Agronomy*, *11*(4), 646. https://doi.org/10.3390/agronomy11040646

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197-387.https://doi.org/10.1561/2000000039

Droukas, L., Doulgeri, Z., Tsakiridis, N. L., Triantafyllou, D., Kleitsiotis, I., Mariolis, I., Giakoumis, D., Tzovaras, D., Kateris, D., & Bochtis, D. (2023). A Survey of Robotic Harvesting Systems and Enabling Technologies. *Journal of Intelligent & Robotic Systems*, *107*(2), 21.https://doi.org/10.1007/s10846-022-01793-z

Du, F. J., & Jiao, S. J. (2022). Improvement of lightweight convolutional neural network model based on YOLO algorithm and its research in pavement defect detection. *Sensors*, *22*(9), 3537. https://doi.org/10.3390/s22093537

Elavarasan, D., & Vincent, P. D. (2020). Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. *IEEE Access*, *8*, 86886-86901. https://doi.org/10.1109/access.2020.2992480

Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Hellmann Santos, C., & Pekkeriet, E. (2020). Agricultural robotics for field operations. *Sensors*, *20*(9), 2672. https://doi.org/10.3390/s20092672

Gholipoor, M., & Fathollah, N. (2019). Fruit yield prediction of pepper using artificial neural network. *Scientia Horticulturae*, *250*, 249-253.https://doi.org/10.1016/j.scienta.2019.02.040

Hong, W., Ma, Z., Ye, B., Yu, G., Tang, T., & Zheng, M. (2023). Detection of green asparagus in complex environments based on the improved YOLOv5 algorithm. *Sensors*, *23*(3), 1562. https://doi.org/10.3390/s23031562

İmak, A., Doğan, G., Şengür, A., & Ergen, B. (2023). A new method based on extracting, combining and selecting deep features from natural and synthetic data for classification of grapevine leaf species. *Int. J. Pure App. Sci.*, *9*(1), 46-55. https://doi.org/10.29132/ijpas.1144615

Jaramillo-Hernández, J. F., Julian, V., Marco-Detchart, C., & Rincón, J. A. (2024). Application of machine vision techniques in low-cost devices to improve efficiency in precision farming. *Sensors*, *24*(3), 937. https://doi.org/10.3390/s24030937

Kaldarova, M., Akanova, A., Nazyrova, A., Mukanova, A., & Tynykulova, A. (2023). Identification Of Weeds In Fields Based On Computer Vision Technology. *Eastern-European Journal of Enterprise Technologies*, *124*(2). https://doi.org/10.15587/1729-4061.2023.284600

Karahanlı, G., & Taşkın, C. (2024). Determining the growth stages of sunflower plants using deep learning methods. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *39*(3), 1455-1472. https://doi.org/10.17341/gazimmfd.1200615

Kini, A. S., Reddy, P. K., & Pai, S. N. (2023). Techniques of deep learning and image processing in plant leaf disease detection: A review. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(3), 3029-3040. https://doi.org/10.11591/ijece.v13i3.pp3029-3040

Lu, D., Ye, J., Wang, Y., & Yu, Z. (2023). Plant detection and counting: Enhancing precision agriculture in UAV and general scenes. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3325747

Luo, J., Li, B., & Leung, C. (2023). A survey of computer vision technologies in urban and controlled-environment agriculture. *ACM Computing Surveys*, *56*(5), 1-39. https://doi.org/10.1145/1122445.1122456

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*,105254. https://doi.org/1016/j.compag.2020.105254

Nath, S. (2024). A vision of precision agriculture: Balance between agricultural sustainability and environmental stewardship. *Agronomy Journal*, *116*(3), 1126-1143. https://doi.org/10.1002/agj2.21405

Palacios, F., Diago, M. P., Melo-Pinto, P., & Tardaguila, J. (2023). Early yield prediction in different grapevine varieties using computer vision and machine learning. *Precision Agriculture*, *24*(2), 407-435.https://doi.org/10.1007/s11119-022-09950-y

Punithavathi, R., Rani, A. D. C., Sughashini, K., Kurangi, C., Nirmala, M., Ahmed, H. F. T., & Balamurugan, S. (2023). Computer Vision and Deep Learning-enabled Weed Detection Model for Precision Agriculture. *Comput. Syst. Sci. Eng.*, *44*(3), 2759-2774. https://doi.org/10.32604/csse.2023.027647

Rai, N., Mahecha, M. V., Christensen, A., Quanbeck, J., Zhang, Y., Howatt, K., Ostlie, M., & Sun, X. (2023). Multi-format open-source weed image dataset for real-time weed identification in precision agriculture. *Data in Brief*, *51*, 109691. https://doi.org/10.1016/j.dib.2023.109691

Redmon, J., & Farhadi, A. (2017). *YOLO9000: better, faster, stronger.* Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. In computer vision and pattern recognition. *arXiv preprint arXiv:1804.02767*. https://doi.org/10.48550/arXiv.1804.02767

Rivera Zarate, G. (2023). LiDAR applications in precision agriculture for cultivating crops: A review of recent advances. *Instituto de Ingeniería y Tecnología* (207), 107737. https://doi.org/10.1016/j.compag.2023.107737

Roshanianfard, A., Noguchi, N., Ardabili, S., Mako, C., & Mosavi, A. (2022). Autonomous robotic system for pumpkin harvesting. *Agronomy*, *12*(7), 1594. https://doi.org/10.3390/agronomy12071594

Rudenko, M., Plugatar, Y., Korzin, V., Kazak, A., Gallini, N., & Gorbunova, N. (2023). The use of computer vision to improve the affinity of rootstock-graft combinations and identify diseases of grape seedlings. *Inventions*, *8*(4), 92. https://doi.org/10.3390/inventions8040092

Sapkota, R., Stenger, J., Ostlie, M., & Flores, P. (2023). Towards reducing chemical usage for weed control in agriculture using UAS imagery analysis and computer vision techniques. *Scientific Reports*, *13*(1), 6548. https://doi.org/10.1038/s41598-023-33042-0

Shin, Y. H., Park, M. J., Lee, O. Y., & Kim, J. O. (2020). Deep orthogonal transform feature for image denoising. *IEEE Access*, *8*, 66898-66909. https://doi.org/10.1109/ACCESS.2020.2986827

Soeb, M. J. A., Jubayer, M. F., Tarin, T. A., Al Mamun, M. R., Ruhad, F. M., Parven, A., Mubarak, N. M., Karri, S. L., & Meftaul, I. M. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Scientific Reports*, *13*(1), 6078. https://doi.org/10.1038/s41598-023-33270-4

Štaka, Z., & Mišić, M. (2023). Leaf counting in the presence of occlusion in Arabidopsis thaliana plant using convolutional neural networks. *Journal of Electronic Imaging*, *32*(5), 052407-052407. https://doi.org/10.1117/1.jei.32.5.052407

Ubaid, M. T., & Javaid, S. (2024). Precision agriculture: Computer vision-enabled sugarcane plant counting in the tillering phase. *Journal of Imaging*, *10*(5), 102. https://doi.org/10.3390/jimaging10050102

Wang, H., Ji, C., Gu, B., & Tian, G. (2013). Cucumber image segmentation based on weighted connection coefficient pulse coupled neural network. *Nongye Jixie Xuebao= Transactions of the Chinese Society for Agricultural Machinery*, *44*(3), 204-208. https://doi.org/10.6041/j.issn.1000-1298.2013.03.037

Wang, Y., Wang, Y., & Zhao, J. (2022). MGA-YOLO: A lightweight one-stage network for apple leaf disease detection. *Frontiers in Plant Science*, *13*, 927424. https://doi.org/10.3389/fpls.2022.927424

Xiao, F., Wang, H., Li, Y., Cao, Y., Lv, X., & Xu, G. (2023). Object detection and recognition techniques based on digital image processing and traditional machine learning for fruit and vegetable harvesting robots: an overview and review. *Agronomy*, *13*(3), 639. https://doi.org/10.3390/agronomy13030639

Xu, J., & Lu, Y. (2023). *Openweedgui: an open-source graphical user interface for weed imaging and detection.* Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping VIII. https://doi.org/10.1117/12.2664131

Zhu, L., Li, Z., Li, C., Wu, J., & Yue, J. (2018). High performance vegetable classification from images based on alexnet deep learning model. *International Journal of Agricultural and Biological Engineering*, *11(4)*,217-223. https://doi.org/10.25165/j.ijabe.20181104.2690

Zualkernan, I., Abuhani, D. A., Hussain, M. H., Khan, J., & El-Mohandes, M. (2023). Machine learning for precision agriculture using imagery from unmanned aerial vehicles (uavs): A survey. *Drones*, *7(6)*, 382. https://doi.org/10.3390/drones7060382