COMPARATIVE ANALYSIS OF POPULAR CNN BASED DEEP LEARNING MODELS FOR TREE TRUNK DETECTION IN ORCHARDS

M. Cejnek[*], *J. Vrba*[*], *J. Jura*[*], *P. Trnka*[*], *L. Zeleny*[†]

Abstract: This study compares machine vision deep learning models based on convolutional neural networks to detect tree trunks in orchards from camera images, with a primary focus on apple trees. Two distinct datasets are used, one original with apple trees and another publicly available featuring vineyard trunks. Multiple deep learning models are tested and compared in order to evaluate their efficacy in tree trunk detection. Research not only provides insight into the performance of various models but also serves as a valuable benchmark for assessing achievable results in orchard-based machine vision applications. The findings contribute to the field's understanding of tree trunk detection, facilitating advancements in agricultural automation.

Key words: *machine vision, deep learning, tree trunk, orchard automation, precision agriculture*

Received: September 2, 2024 **DOI:** 10.14311/NNW.2024.34.014
Revised and accepted: October 29, 2024

1. Introduction

The use of automated systems in orchard management has become increasingly popular in the agricultural industry. Automation in agriculture has the potential to increase efficiency, reduce labor costs, and improve overall productivity and sustainability. Automated orchard management systems could perform a variety of tasks, including planting [35], pruning [39], irrigation [27], pest monitoring [4, 9], spraying [32] and harvesting [39]. These systems use advanced algorithms and sensors to identify specific needs and apply the necessary input to optimize their growth and yield. Automating these processes can reduce the need for manual labor and increase the precision of management practices. In addition, automated orchard management systems can help farmers reduce the amount of input they use, which can lead to improved environmental sustainability. With the growing demand for high-quality production and the need for more efficient farming practices,

[*]Matous Cejnek, Jan Vrba, Jakub Jura, Pavel Trnka; Department of Instrumentation and Control Engineering, FME, CTU, Technicka 4, Prague, 16607, Czech Republic E-mail: matous.cejnek@fs.cvut.cz

[†]Lubor Zeleny; Research and Breeding Institute of Pomology Holovousy, Holovousy 129, 50801, Czech Republic

automated orchard management presents an opportunity for farmers to optimize their production processes and meet all needs.

Machine-based real-time object detection plays a vital role in automated orchard management. There are two main directions for machine vision in future orchard management. The first major importance of machine vision is the real-time operation of unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV) for automatic harvesting, data collection, and other field tasks. Both types of vehicles are used in modern orchards [40, 29, 12]. In particular, UGV navigation requires you to avoid collisions with people, trees, and other obstacles [7]. The second main direction of the use of machine vision in agriculture is offline data processing for planning orchard operations. This group of tasks contains operations such as counting leaves, fruits, branches, and other measurements of tree development. Both orchard operations: real-time navigation and data offline processing have specific time and precision requirements. In the case of UGV navigation, machine vision requirements are bound to the area of operation and the available hardware and computational resources. The UGV sensor equipment usually consists of an ultrasound sensor or a laser scanner combined with an RGB camera, possibly combined with a depth camera (RGBD). The common results of this data collection have been well investigated in the general field of UGV navigation [15]. The same holds for computational approaches to data evaluation. A study proposes an approach to navigate the UGV in the orchard with common sensors (lidar and RGB camera) [6]. The authors of the study used a hidden semi-Markov detector to detect tree trunks in trellis-structured apple orchards. Another promising proposed approach is to use a thermal camera instead of an RGB camera to make the process less light-dependent. This approach is presented in study [20], where the authors use a thermal camera to detect pear tree trunks using Faster R-CNN.

However, the modern trend is to reduce dependency on other sensors and use only an RGB camera. With this focus on RGB imaging, convolutional neural networks (CNN) have started to play a key role in object detection since 2012. Due to increasing computational power, the depths and number of parameters of neural networks have increased, and object detection is nowadays completely dominated by the use of deep learning models [43].

This trend of deep neural network dominance is also evident in the field of orchard and/or forest management and maintenance. In the study [14] the tracking and counting of apples and tree trunks is implemented using the YOLOv4 tiny detector and the discriminative correlation filter with channel and spatial reliability (CSR-DCF) detector [26]. In the study [5], authors use YOLOv3, YOLOv5, and faster R-CNN models for the detection of trunks in vineyards. They achieved the best performance and inference time with YOLOv5. In the study [3], authors propose two original models: Single Shot Multibox based on a feed-forward CNN, and MobileNets [17] for the detection of trunks in vineyards. According to the authors, SSD MobileNet-2 slightly outperformed other models in terms of average precision AP. In the study [41], authors use Alexnet, VGG16, and VGG19 to detect shaking points in three apples in orchards for automatic harvest. The results of the study suggest that VGGs should be used over Alexnet. In the study [42], three CNN architectures were employed, namely Deeplab v3+ ResNet 18, VGG 16 and VGG19 for the detection of spots to use the three shakers in the trellis-trained

apple tree. The results presented show that the ResNet-based model outperformed others in terms of accuracy. In study [10], the authors tested many different deep learning models – various versions of YOLO, Single-Shot Detector combined with different versions of MobileNet, and EfficientDet. The study presented its own data set based on data obtained from forest areas. The best results for the given data set are achieved with YOLOv7 Tiny with an image resolution of 640×640. In study [38], authors propose a new Y3TM model based on YOLOv3 for the detection of tree trunks. In the study [33], the authors propose the detection of apple tree trunks with the improved YOLOv5s model. They propose to add the squeeze and excitation module [18] to the last layer of the backbone, which leads to an improvement in the mean average precision mAP by 1.3% to 95.2%.

All the studies mentioned present promising results. However, every study has its own data set and a specific definition of a problem they are trying to solve – different sensor resolutions, trunk distance ranges, sizes, and shapes. Therefore, the results are hardly comparable in all articles.

Another issue arises with the availability of public datasets containing the annotated apple tree trunks in orchards. There is only one publicly available dataset [11] with annotated trunk, but it is related to the problem of finding vineyard trunks.

This study addresses the challenge of detecting apple tree trunks in RGB images to identify and crop the region of interest. To compare various deep learning models, we have created an appropriate publicly available annotated dataset.

2. Materials and Methods

2.1 Datasets

In this study, we evaluated the performance of various object detectors on two datasets. Due to the lack of public datasets with apple tree trunks in orchards, we created a publicly available dataset with annotated tree trunks from vertical system apple trees. The second data set (HUMAIN-Lab Vine-Trunk database [11]) is used to provide some connection to previously published studies in this field. This particular data set is chosen because it presents challenges reasonably similar to the ones explained in this paper. Although this data set is not as exhausting, it is still worthy of being used as a bridge between different studies.

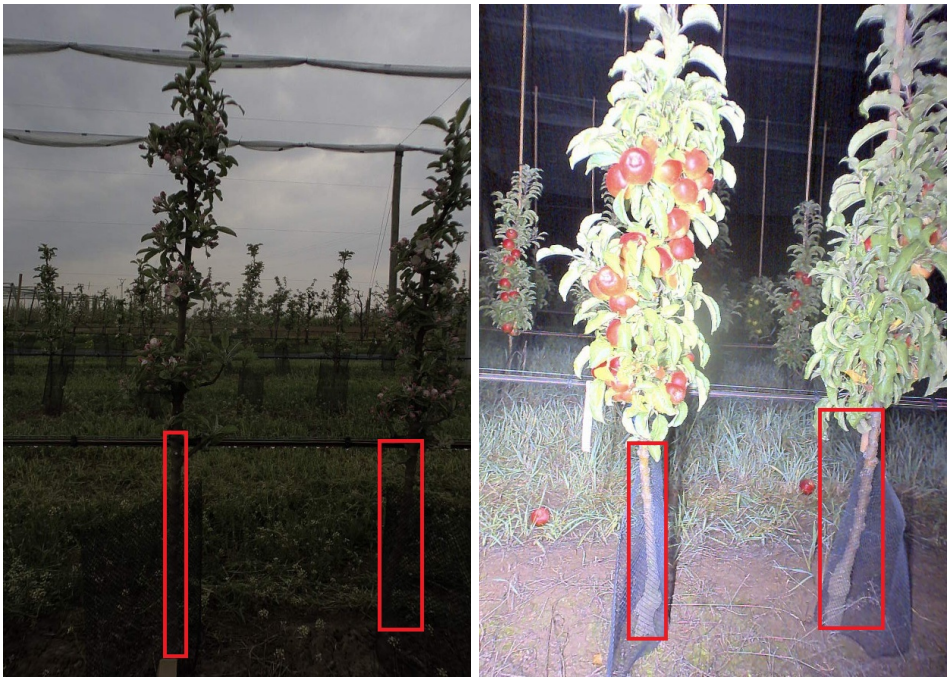
We created a new data set for this study to correspond exactly to the objective of this study.

2.1.1 Apple Tree Trunk Dataset

We created this dataset [8] to fully represent the challenges presented in this study. The images in our data set were obtained using the AV3236DN camera (3MP, WDR, 1/3.2") with MPL3.5 lenses (3.5 mm, 1/2.5", f1.8). Images were taken from two different orchards during multiple data collection sessions. Therefore, some pictures may contain leaves, fruits, and flowers. The entire data set consists of 1580 training images and 176 validation images. Every picture has one, two, or three trunks within. Each orchard contains more than 100 trees consisting of multiple

cultivars of columnar apple trees. The images were taken at a similar distance (100–120 cm) with a focus on the trunk part of the tree. This distance represents half or less of the orchard path width. In practical applications, shorter distance would require much bigger field of view and greater distance is not possible to achieve due to space restrictions. The light conditions vary greatly among the pictures: good visibility during the day, very poor visibility during the late evening, and artificial lighting during the night. Sample images are shown in Fig. 1. Visibility also varies and is also influenced by weather during data collection sessions. Both orchards used during data collection are located in Czechia.

To summarize the high diversity of the images in our data set, images contain different tree cultivars, trees of different age during different year seasons, and various light and weather conditions. Furthermore, one orchard is located within the capital city Prague (50.1210742N, 14.4006475E) while the other is located in a more rural, low-populated area (50.3712739N, 15.5693422E). Therefore, this provided data set represents a complex challenge that entails real-life conditions during the orchard lifecycle.



(a) *Cloudy sky image taken in the early evening.*

(b) *Image with artificial light captured at night.*

Fig. 1 *Sample images from proposed dataset displaying lightning and weather conditions with indicated ground truth bounding boxes.*

2.1.2 HUMAIN-Lab Vine-Trunk Database

The HUMAIN-Lab Vine-Trunk database consists of images of vineyard trunks that are captured by an RGB high-resolution Samsung NX500 Mirrorless camera from three vineyards in North Greek [5]. From this dataset, we used 1883 images for training and 269 images for validation. Training images are augmented, there is applied rotation (between −22 and +22 degrees), brightness (between −55% and +55%), and blur (up to 3.5 pixels). All images have a resolution of 416×416. This dataset is the most similar to the one proposed in this paper to exactly describe the problem that has to be tackled. This dataset contains only images with sufficient sunlight (no artificial light). Sample images are shown in Fig. 2. At least one study [5] was conducted with the HUMAIN-Lab database using neural networks, therefore, this study can be used as a comparative benchmark to evaluate the complexity of our original data set.

Fig. 2 *Images from HUMAIN-Lab Vine-Trunk dataset with indicated ground truth bounding boxes.*

2.2 Evaluated Deep Learning Models

2.2.1 YOLO

You Only Look Once (YOLO) [30] is a state-of-the-art object detection algorithm in computer vision. Unlike traditional object detection algorithms that use a sliding window approach to search for objects in an image, YOLO divides the image into a grid and predicts the bounding boxes and class probabilities for each cell of the grid simultaneously. This approach leads to a much faster object detection process, making it well-suited for real-time applications. The YOLO algorithm uses a deep CNN to predict the bounding boxes and class probabilities for each cell in the grid. The network is trained on a large dataset of labeled images using supervised learning techniques. During training, the network learns to predict the bounding boxes and class probabilities that minimize a predefined loss function.

So far, YOLO has developed multiple versions over time. In this paper, experiments different versions are utilized: YOLOv5 [22], YOLOv6 [23], YOLOv7 [37] and YOLOv8 [21]. Every YOLO version also provides different model sizes (n, s, m, l, tiny, etc.). These sizes represent the number of parameters in the model.

One of the key advantages of YOLO is its ability to handle objects of different sizes and aspect ratios. By predicting the bounding boxes directly, YOLO is able to accurately localize objects of various sizes and shapes. Additionally, YOLO has a high level of accuracy and achieves state-of-the-art results on benchmark object detection datasets. Overall, YOLO is a powerful tool for object detection tasks in computer vision, particularly for real-time applications that require fast and accurate detection.

2.2.2 Single Shot Detector

The Single Shot Detector (SSD) [25] comprises two fundamental components: a backbone model and the SSD head. The backbone model typically employs a pre-trained image classification network, such as ResNet [16], initially trained on ImageNet but with the removal of its final fully connected classification layer. This process results in a deep neural network serving as a feature extractor. The backbone network retains the ability to extract semantic meaning from input images, preserving their spatial structure, although at a reduced resolution. The SSD head consists of one or more additional convolutional layers integrated with the backbone. The outputs of the SSD head are then interpreted as bounding boxes and object classes, spatially located within the final layers' activations.

2.2.3 Faster R-CNN

The Faster R-CNN model [31] consists of two main components: the Region Proposal Network (RPN) and the Fast R-CNN. The RPN acts as a standalone module to generate region proposals. It operates on the final feature maps of a deep CNN, typically a pre-trained network like VGG or ResNet. The RPN generates region proposals by sliding a small network (typically a few convolutional layers) over the feature maps. For each anchor on multiple scales and aspect ratios, the RPN predicts objectness scores and regresses bounding-box coordinates. High-scoring proposals are selected for further processing, providing a more focused set of regions likely to contain objects. The proposals generated by the RPN are fed into Fast R-CNN for object classification and precise bounding-box regression. The Fast R-CNN employs region of interest pooling, allowing it to adaptively resize the proposed regions into a fixed-size feature map. This map is then used for subsequent fully connected layers. The final output includes the class probabilities for each proposed region, along with refined bounding-box coordinates. Faster R-CNN's key innovation lies in its unified architecture, where the RPN and Fast R-CNN are trained jointly, enabling end-to-end training and improving overall accuracy. By introducing a region proposal mechanism within the model, Faster R-CNN significantly outperformed its predecessors, setting new benchmarks in object detection accuracy and efficiency.

2.2.4 EfficientDet

EfficientDet [34], proposed in 2019, introduces a compound scaling method that simultaneously scales the model's depth, width, and resolution. This novel approach allows for better resource utilization and improved performance. The compound scaling coefficients are carefully tuned to maintain a balance between model accuracy and computational efficiency. This ensures that the model can perform well on various tasks, while being suitable for deployment on various devices. EfficientDet employs a Bi-directional Feature Pyramid Network (BiFPN) structure that facilitates information flow not only from high-resolution levels to low-resolution levels but also vice versa, promoting more effective feature utilization. The BiFPN plays a crucial role in connecting different network scales, allowing the model to capture both fine-grained details and high-level context. EfficientDet utilizes separate detection heads for classification and regression tasks. These heads are connected to the feature pyramid network, ensuring that predictions are made on multiple scales. EfficientDet employs focal loss for classification, which helps the model focus on hard examples, and smooth loss for bounding-box regression, contributing to more robust and accurate predictions. Through the compound scaling strategy, EfficientDet achieves state-of-the-art accuracy with fewer parameters compared to previous models. This makes it computationally efficient and more suitable for deployment on resource-constrained devices.

2.2.5 CenterNet

Unlike traditional two-stage detectors, CenterNet [13] pioneers a single-stage approach that directly predicts object centers, bounding boxes, and class probabilities. The main idea of CenterNet lies in treating object detection as a keypoint estimation problem. In its design, CenterNet incorporates a backbone model responsible for feature extraction, often leveraging a pre-trained architecture. The resulting outcome is then obtained through specific CenterNet heads:

- Center Head – Identifies object centers by predicting keypoint locations on feature maps.
- Regression Head: Formulates precise bounding box predictions based on refined feature maps.
- Classification Head: assigns class probabilities to objects.

In other words, CenterNet keypoint detection can be used to detect the center point of the bounding box and regress to all other object properties such as the bounding box size, 3d information, and pose.

2.3 Experiment Setup

All evaluated models were pre-trained in the COCO 2017 dataset [24] and refined in the apple tree trunk dataset and the vine trunk dataset. During training, batch size 8 was used for all experiments, except for the YOLOv7-D6 and YOLOv7-E6E models, where batch size was reduced to 4 due to the size of those models and GPU used. All models were trained for 100 epochs; other hyperparameters were left to

default values. Used evaluation metrics are: precision, accuracy, recall, mAP50 and mAP95.

All experiments were performed on a PC with an AMD Ryzen 9 5900X 12 cores CPU running at 3701 MHz and 128 GB RAM. The operating system was Windows 10 Enterprise LTSC version 10.0.19044 build 19044. The code for the experiments is written in Python 3.9 [36]. YOLO models use Pytorch 1.13.1 [28] while the rest of the models are based on Tensorflow 2.10.1 [2, 1], and these models are part of TensorFlow 2 Detection Model Zoo [19].

3. Results

To compare the performance of models, we evaluated the mean average precision mAP for the confidence threshold 0.5, mean average precision for multiple confidence thresholds $mAP95$, precision and recall. All results for tree trunk detection are summarized in Tab. III (Tensorflow2 Object Detection Zoo models), Tab. IV (YOLO family models with resolution 640×640), and Tab. V (YOLO family results with resolution 1280×1280). Evaluation of training time for a single epoch for both datasets is shown in Tab. I (for the YOLO family models) and Tab. II (for the Tensorflow2 Object Detection Zoo models).

Model	Resolution	TT_a	TT_v	Model	Resolution	TT_a	TT_v
YOLOv5n	640	14	15	YOLOv5n6	1280	16	19
YOLOv5s	640	15	16	YOLOv5s6	1280	17	20
YOLOv5m	640	18	21	YOLOv5m6	1280	23	26
YOLOv5l	640	24	28	YOLOv5l6	1280	27	33
YOLOv5x	640	36	43	YOLOv5x6	1280	39	48
YOLOv6-N	640	23	24	YOLOv6-N6	1280	32	29
YOLOv6-S	640	25	26	YOLOv6-S6	1280	47	31
YOLOv6-M	640	36	41	YOLOv6-M6	1280	99	48
YOLOv6-L	640	32	39	YOLOv6-L6	1280	105	51
YOLOv7-tiny	640	22	27	YOLOv7-W6	1280	101	113
YOLOv7	640	37	44	YOLOv7-E6	1280	132	156
YOLOv7-X	640	47	59	YOLOv7-D6*	1280	159	199
YOLOv8n	640	10	13	YOLOv7-E6E*	1280	189	235
YOLOv8s	640	11	14				
YOLOv8m	640	19	22				
YOLOv8l	640	27	33				
YOLOv8x	640	40	47				

Tab. I *Table with YOLO models resolution in pixels, and training times for single epoch for apple trunk dataset (TT_a) and vine trunk dataset (TT_v) in seconds.*

Model	Resolution	TT_a	TT_v
CenterNet HourGlass104	512	182	223
CenterNet Resnet50 V1 FPN	512	37	42
CenterNet Resnet50 V2	512	30	38
Center Net Resnet101 V1 FPN	512	54	63
CenterNet MobileNet V2 FPN	512	25	28
EfficientDet D0	640	46	59
EfficientDet D1	640	96	110
Faster R-CNN ResNet50 V1	640	79	110
Faster R-CNN ResNet101 V1	640	103	119
Faster R-CNN ResNet152 V1	640	159	190
SSD MobileNet V1 FPN	640	69	70
SSD MobileNet V2 FPNLite	640	46	53
SSD Resnet50 V1 FPN	640	79	91
SSD Resnet101 V1 FPN	640	103	120
SSD Resnet152 V1 FPN	640	179	220

Tab. II *Table with selected Tensorflow object detection ZOO models resolution in pixels, and training times for single epoch for apple trunk dataset (TT_a) and vine trunk dataset (TT_v) in seconds.*

4. Discusion

This study evaluated the models on two distinct datasets, apple and vine trunks, broadening the applicability of the findings. Better results were obtained with the apple trunks dataset. However, this improvement may not necessarily be due to the type of tree trunks but could be influenced by other variables related to the overall scene. In particular, variations in lighting and weather could affect the visibility and texture of the tree trunks, which may in turn influence the model's performance. For example, images taken under overcast or cloudy conditions may offer more consistent lighting, which could enhance detection accuracy, whereas bright sunlight or shadows might obscure important features, making detection more difficult.

This demonstrates the potential for the models to be used in diverse orchard environments, where tree trunk types and visual characteristics might differ. Further exploration of these environmental factors (e.g., different weather conditions or times of day) is crucial to better understand how such variations impact detection performance. However, there is a trade-off between accuracy and training speed. Models with higher mAP scores, such as YOLOv6-L6, often require more training time per epoch.

Among the models tested, YOLOv5x and YOLOv5m6 emerged as the top performers, suggesting that the YOLOv5 architecture is well suited for tree trunk detection tasks. However, the optimal image resolution appears to depend on model and tree trunk type. While higher resolution might be beneficial in some scenarios, lighting and weather variations could also affect how well the model generalizes to different conditions, and this study does not provide a definitive answer. This

Apple trunks				
Model	*mAP*	*mAP*95	R	P
CenterNet HourGlass104	0.895	0.375	0.971	0.921
CenterNet Resnet50 V1 FPN	**0.959**	0.464	0.991	0.985
CenterNet Resnet50 V2	0.936	**0.474**	0.994	0.973
CenterNet Resnet101 V1 FPN	0.944	0.469	0.988	0.973
CenterNet MobileNet V2 FPN	0.931	0.447	0.987	**0.984**
EfficientDet D0	0.895	0.375	0.971	0.921
EfficientDet D1	0.938	0.413	0.994	0.952
Faster R-CNN ResNet50 V1	0.932	0.469	0.986	0.97
Faster R-CNN ResNet101 V1	0.946	0.481	**1**	0.975
Faster R-CNN ResNet152 V1	0.751	0.273	0.790	0.915
SSD MobileNet V1 FPN	0.944	0.446	0.982	0.958
SSD MobileNet V2 FPNLite	0.929	0.437	0.971	0.955
SSD Resnet50 V1 FPN	0.872	0.364	0.973	0.877
SSD Resnet101 V1 FPN	0.938	0.451	0.976	0.969
SSD Resnet152 V1 FPN	0.751	0.273	0.790	0.917
Vine trunks				
Model	*mAP*	*mAP*95	R	P
CenterNet HourGlass104	0.720	**0.275**	0.891	0.923
CenterNet Resnet50 V1 FPN	0.658	0.225	0.902	0.883
CenterNet Resnet50 V2	0.657	0.232	0.916	0.875
CenterNet Resnet101 V1 FPN	0.688	0.233	0.916	0.910
CenterNet MobileNet V2 FPN	0.506	0.153	0.872	0.822
EfficientDet D0	0.719	0.266	0.911	**0.955**
EfficientDet D1	**0.734**	0.263	0.944	0.911
Faster R-CNN ResNet50 V1	0.598	0.191	**0.951**	0.754
Faster R-CNN ResNet101 V1	0.624	0.212	0.943	0.783
Faster R-CNN ResNet152 V1	0.630	0.207	0.946	0.787
SSD MobileNet V1 FPN	0.637	0.197	0.911	0.878
SSD MobileNet V2 FPNLite	0.623	0.190	0.882	0.869
SSD Resnet50 V1 FPN	0.572	0.175	0.878	0.797
SSD Resnet101 V1 FPN	0.536	0.161	0.870	0.738
SSD Resnet152 V1 FPN	0.554	0.17	0.886	0.760

Tab. III *Results of Tensorflow ZOO object detection models – R stands for recall, P stands for precision.*

highlights the importance of exploring different image resolutions during model fine-tuning for specific applications.

Using the COCO17 dataset for pretraining proved to be a viable approach. It provided a strong foundation for the models, allowing them to learn general object detection capabilities before fine-tuning on the specific task. Although this study does not compare different pre-training strategies, future research could explore this avenue for potential performance improvements.

Model	Apple trunks				Vine trunks			
	mAP	mAP95	R	P	mAP	mAP95	R	P
YOLOv5n	0.977	0.527	0.963	0.964	0.731	0.256	0.706	0.769
YOLOv5s	0.971	0.530	0.95	0.972	0.747	0.274	**0.728**	0.771
YOLOv5m	0.985	0.539	0.961	0.975	0.734	0.292	0.707	0.777
YOLOv5l	0.978	0.533	0.964	0.975	0.755	0.299	0.708	0.791
YOLOv5x	**0.988**	0.527	**0.972**	**0.982**	0.746	0.299	0.718	0.776
YOLOv6-N	0.957	0.466	0.934	0.968	0.722	0.275	0.644	0.765
YOLOv6-S	0.97	0.545	0.945	0.974	0.759	0.3	0.722	0.776
YOLOv6-M	0.964	0.48	0.944	0.966	0.76	**0.307**	0.706	0.768
YOLOv6-L	0.963	0.484	0.95	0.961	**0.762**	0.299	0.705	0.784
YOLOv7-tiny	0.963	0.474	0.939	0.95	0.69	0.231	0.69	0.726
YOLOv7	0.914	0.452	0.853	0.945	0.717	0.258	0.7	0.779
YOLOv7-X	0.925	0.466	0.861	0.954	0.747	0.273	0.726	0.779
YOLOv8n	0.978	0.538	0.953	0.972	0.721	0.281	0.653	0.771
YOLOv8s	0.973	0.54	0.958	0.964	0.732	0.279	0.672	**0.799**
YOLOv8m	0.974	**0.544**	0.939	0.974	0.736	0.281	0.706	0.761
YOLOv8l	0.971	0.534	0.956	0.953	0.71	0.272	0.657	0.766
YOLOv8x	0.974	0.535	0.958	0.961	0.731	0.271	0.648	.0753

Tab. IV *Results of YOLO* 640×640 *family models – R stands for recall, P stands for precision.*

Model	Apple trunks				Vine trunks			
	mAP	*mAP*95	R	P	*mAP*	*mAP*95	R	P
YOLOv5n6	0.961	0.510	0.936	0.960	0.775	0.319	0.695	0.805
YOLOv5s6	**0.985**	0.523	0.956	0.982	0.773	0.319	0.741	0.778
YOLOv5m6	0.975	0.524	**0.974**	0.952	**0.779**	**0.327**	0.717	0.790
YOLOv5l6	0.97	0.52	0.942	0.966	0.772	0.326	0.698	**0.814**
YOLOv5x6	0.975	0.52	0.965	0.964	0.755	**0.327**	0.712	0.769
YOLOv6-N6	0.958	0.513	0.942	0.966	0.69	0.233	0.637	0.693
YOLOv6-S6	0.97	0.52	0.947	0.98	0.673	0.238	0.658	0.684
YOLOv6-M6	0.98	0.537	0.972	0.956	0.756	0.279	0.725	0.744
YOLOv6-L6	0.967	**0.546**	0.947	0.969	0.734	0.279	0.679	0.777
YOLOv7-W6	0.98	0.512	0.956	**0.986**	0.76	0.293	0.741	0.748
YOLOv7-E6	0.974	0.513	0.961	0.975	0.774	0.292	0.707	0.802
YOLOv7-D6*	0.975	0.513	0.961	0.964	0.77	0.287	**0.746**	0.778
YOLOv7-E6E*	0.973	0.506	0.956	0.964	0.765	0.279	0.742	0.743

Tab. V *Results of YOLO* 1280×1280 *family models – R stands for recall, P stands for precision.*

5. Conclusion

This comparative investigation presents a performance analysis of contemporary deep CNN designed for object detection. All networks were pretrained on the COCO17 dataset before starting a consistent baseline for further fine-tuning and comprehensive evaluation. Fine-tuning procedures were executed on two relatively small datasets, namely the newly acquired apple tree trunk dataset and the publicly accessible vine trunk dataset. Throughout the training process, we also scrutinized the training time for a single epoch, a technical metric that is seldom reported but essential for the evaluation of development. The optimal model for the detection of vine trunks and apple tree trunks, determined by the mean Average Precision (mAP), is YOLOv5x with $mAP = 0.988$, where the input image resolution is 640×640. Similarly, for the detection of vine trunks, the superior model is YOLOv5m6 with $mAP = 0.779$ where the input image size is 1280×1280. We also evaluated models based on $mAP95$, where the superior model for apple trunk detection was YOLOv6-L6 with $mAP95 = 0.546$ with an input image resolution of 1280×1280. For the detection of vine trunks, the best models were YOLOv5m6 and YOLOv5x6, both of which achieved $mAP95 = 0.327$. Both models have an input image size resolution of 1280×1280. The findings of this study provide many new insights for the task of UGV navigation. Furthermore, the findings can help develop solutions for other essential tasks necessary for effective orchard management, such as counting blossoms and apples, mapping the tree crown structure, and measuring long-term tree development. Lastly, the data set created for this study has value beyond the scope of this study. The availability of this specific data set serves as a valuable resource for researchers and practitioners working on agricultural automation and precision agriculture.

Acknowledgement

This research has been supported by the QK21010170 Research Grant, New Orchard concept using 4.0 technology.

References

[1] ABADI M., AGARWAL A., BARHAM P., BREVDO E., CHEN Z., CITRO C., CORRADO G.S., DAVIS A., DEAN J., DEVIN M., GHEMAWAT S., GOODFELLOW I., HARP A., IRVING G., ISARD M., JIA Y., JOZEFOWICZ R., KAISER L., KUDLUR M., LEVENBERG J., MANÉ D., MONGA R., MOORE S., MURRAY D., OLAH C., SCHUSTER M., SHLENS J., STEINER B., SUTSKEVER I., TALWAR K., TUCKER P., VANHOUCKE V., VASUDEVAN V., VIÉGAS F., VINYALS O., WARDEN P., WATTENBERG M., WICKE M., YU Y., ZHENG X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Available also from: https://www.tensorflow.org/.

[2] ABADI M., BARHAM P., CHEN J., CHEN Z., DAVIS A., DEAN J., DEVIN M., GHEMAWAT S., IRVING G., ISARD M., et al. Tensorflow: A system for large-scale machine learning. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[3] AGUIAR A.S., MONTEIRO N.N., SANTOS F.N.d., SOLTEIRO PIRES E.J., SILVA D., SOUSA A.J., BOAVENTURA-CUNHA J. Bringing semantics to the vineyard: An approach on deep learning-based vine trunk detection. *Agriculture*. 2021, 11(2), pp. 131.

[4] ALBANESE A., NARDELLO M., BRUNELLI D. Automated pest detection with DNN on the edge for precision agriculture. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 2021, 11(3), pp. 458–467.

[5] BADEKA E., KALAMPOKAS T., VROCHIDOU E., TZIRIDIS K., PAPAKOSTAS G., PACHIDIS T., KABURLASOS V. Real-time vineyard trunk detection for a grapes harvesting robot via deep learning. In: *Thirteenth International Conference on Machine Vision*, 2021, pp. 394–400.

[6] BARGOTI S., UNDERWOOD J.P., NIETO J.I., SUKKARIEH S. A pipeline for trunk detection in trellis structured apple orchards. *Journal of field robotics*. 2015, 32(8), pp. 1075–1094.

[7] BERGERMAN M., MAETA S.M., ZHANG J., FREITAS G.M., HAMNER B., SINGH S., KANTOR G. Robot farmers: Autonomous orchard vehicles help tree fruit production. *IEEE Robotics & Automation Magazine*. 2015, 22(1), pp. 54–63.

[8] CEJNEK M., VRBA J., JURA J., TRNKA P. *Apple tree images for trunk detection experiments (YOLOv8 format)*. 2023. Available also from: https://figshare.com/articles/dataset/Apple_tree_images_for_trunk_detection_experiments_YOLOv8_format_/24849711.

[9] CHEN C.-J., HUANG Y.-Y., LI Y.-S., CHEN Y.-C., CHANG C.-Y., HUANG Y.-M. Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access*. 2021, 9, pp. 21986–21997.

[10] Da SILVA D.Q., dos SANTOS F.N., FILIPE V., SOUSA A.J., OLIVEIRA P.M. Edge AI-Based Tree Trunk Detection for Forestry Monitoring Robotics. *Robotics*. 2022, 11(6), pp. 136.

[11] De AGUIAR A.S.P., dos SANTOS F.B.N., dos SANTOS L.C.F., de JESUS FILIPE V.M., de SOUSA A.J.M. Vineyard trunk detection using deep learning–An experimental device benchmark. *Computers and Electronics in Agriculture*. 2020, 175, pp. 105535.

[12] DROUKAS L., DOULGERI Z., TSAKIRIDIS N.L., TRIANTAFYLLOU D., KLEITSIOTIS I., MARIOLIS I., GIAKOUMIS D., TZOVARAS D., KATERIS D., BOCHTIS D. A Survey of Robotic Harvesting Systems and Enabling Technologies. *Journal of Intelligent & Robotic Systems*. 2023, 107(2), pp. 21.

[13] DUAN K., BAI S., XIE L., QI H., HUANG Q., TIAN Q. Centernet: Keypoint triplets for object detection. In: *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6569–6578.

[14] GAO F., FANG W., SUN X., WU Z., ZHAO G., LI G., LI R., FU L., ZHANG Q. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Computers and Electronics in Agriculture*. 2022, 197, pp. 107000.

[15] GAO X., LI J., FAN L., ZHOU Q., YIN K., WANG J., SONG C., HUANG L., WANG Z. Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *Ieee Access*. 2018, 6, pp. 49248–49268.

[16] HE K., ZHANG X., REN S., SUN J. Deep Residual Learning for Image Recognition. *CoRR*. 2015, abs/1512.03385.

[17] HOWARD A.G., ZHU M., CHEN B., KALENICHENKO D., WANG W., WEYAND T., ANDREETTO M., ADAM H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*. 2017, abs/1704.04861.

[18] HU J., SHEN L., SUN G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[19] HUANG J., RATHOD V., SUN C., ZHU M., KORATTIKARA A., FATHI A., FISCHER I., WOJNA Z., SONG Y., GUADARRAMA S., et al. Speed/accuracy trade-offs for modern convolutional object detectors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[20] JIANG A., NOGUCHI R., AHAMED T. Tree trunk recognition in orchard autonomous operations under different light conditions using a thermal camera and faster R-CNN. *Sensors*. 2022, 22(5), pp. 2065.

[21] JOCHER G., CHAURASIA A., QIU J. *YOLO by Ultralytics*. 2023. Available also from: https://github.com/ultralytics/ultralytics.

[22] JOCHER G., STOKEN A., BOROVEC J., NANOCODE012, CHRISTOPHERSTAN, CHANGYU L., LAUGHING, TKIANAI, HOGAN A., LORENZOMAMMANA, YXNONG, ALEXWANG1900, DIACONU L., MARC, ML5AH, DOUG, WANGHAOYANG0106, INGHAM F., FREDERIK, GUILHEN, HATOVIX, POZNANSKI J., FANG J., YU L., CHANGYU98, WANG M., GUPTA N., AKHTAR O., PETRDVORACEK, RAI P. *ultralytics/yolov5: v3.1 – Bug Fixes and Performance Improvements*. 2020. Available also from: https://doi.org/10.5281/zenodo.4154370.

[23] LI C., LI L., JIANG H., WENG K., GENG Y., LI L., KE Z., LI Q., CHENG M., NIE W., et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*. 2022.

[24] LIN T., MAIRE M., BELONGIE S.J., BOURDEV L.D., GIRSHICK R.B., HAYS J., PERONA P., RAMANAN D., DOLL'A R P., ZITNICK C.L. Microsoft COCO: Common Objects in Context. *CoRR*. 2014, abs/1405.0312.

[25] LIU W., ANGUELOV D., ERHAN D., SZEGEDY C., REED S., FU C.-Y., BERG A.C. Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016, pp. 21–37.

[26] LUNEŽIČ A, VOJÍŘ T., ZAJC L.Č., MATAS J., KRISTAN M. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*. 2018, 126(7), pp. 671–688.

[27] OSROOSH Y., PETERS R.T., CAMPBELL C.S., ZHANG Q. Automatic irrigation scheduling of apple trees using theoretical crop water stress index with an innovative dynamic threshold. *Computers and Electronics in Agriculture*. 2015, 118, pp. 193–203.

[28] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 8024–8035. Available also from: \url{http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf}.

[29] RADCLIFFE J., COX J., BULANON D.M. Machine vision for orchard navigation. *Computers in Industry*. 2018, 98, pp. 165–171.

[30] REDMON J., DIVVALA S., GIRSHICK R., FARHADI A. You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[31] REN S., HE K., GIRSHICK R., SUN J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015, 28.

[32] SEOL J., KIM J., SON H.I. Field evaluations of a deep learning-based intelligent spraying robot with flow control for pear orchards. *Precision Agriculture*. 2022, 23(2), pp. 712–732.

[33] SU F., ZHAO Y., SHI Y., ZHAO D., WANG G., YAN Y., ZU L., CHANG S. Tree Trunk and Obstacle Detection in Apple Orchard Based on Improved YOLOv5s Model. *Agronomy*. 2022, 12(10), pp. 2427.

[34] TAN M., PANG R., LE Q.V. Efficientdet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781–10790.

[35] TATAR A.B., TANYILDIZI A.K., TAŞAR B. A Conceptual Design of Two DoF Crawler Tree Planting Robot with Helical Digging Arm. In: *2023 14th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, 2023, pp. 8–11.

[36] VAN ROSSUM G., DRAKE F.L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

[37] WANG C.-Y., BOCHKOVSKIY A., LIAO H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[38] YANG T., ZHOU S., XU A. Rapid image detection of tree trunks using a convolutional neural network and transfer learning. *measurement*. 2021, 5, pp. 6.

[39] ZAHID A., MAHMUD M.S., HE L., HEINEMANN P., CHOI D., SCHUPP J. Technological advancements towards developing a robotic pruner for apple trees: A review. *Computers and Electronics in Agriculture*. 2021, 189, pp. 106383.

[40] ZHANG C., VALENTE J., KOOISTRA L., GUO L., WANG W. Orchard management with small unmanned aerial vehicles: A survey of sensing and analysis approaches. *Precision Agriculture*. 2021, 22(6), pp. 2007–2052.

[41] ZHANG J., KARKEE M., ZHANG Q., ZHANG X., YAQOOB M., FU L., WANG S. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Computers and Electronics in Agriculture*. 2020, 173, pp. 105384.

[42] ZHANG X., KARKEE M., ZHANG Q., WHITING M.D. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. *Journal of Field Robotics*. 2021, 38(3), pp. 476–493.

[43] ZOU Z., CHEN K., SHI Z., GUO Y., YE J. Object detection in 20 years: A survey. *Proceedings of the IEEE*. 2023.