machines

MDPI

Article

Autonomous Agricultural Robot Using YOLOv8 and ByteTrack for Weed Detection and Destruction

Ardin Bajraktari and Hayrettin Toylan *

Mechatronics Engineering Department, Faculty of Technology, University of Kirklareli, Kirklareli 39100, Türkiye; ardinbajraktari4@gmail.com

* Correspondence: hayrettintoylan@klu.edu.tr

Abstract: Automating agricultural machinery presents a significant opportunity to lower costs and enhance efficiency in both current and future field operations. The detection and destruction of weeds in agricultural areas via robots can be given as an example of this process. Deep learning algorithms can accurately detect weeds in agricultural fields. Additionally, robotic systems can effectively eliminate these weeds. However, the high computational demands of deep learning-based weed detection algorithms pose challenges for their use in real-time applications. This study proposes a vision-based autonomous agricultural robot that leverages the YOLOv8 model in combination with ByteTrack to achieve effective real-time weed detection. A dataset of 4126 images was used to create YOLO models, with 80% of the images designated for training, 10% for validation, and 10% for testing. Six different YOLO object detectors were trained and tested for weed detection. Among these models, YOLOv8 stands out, achieving a precision of 93.8%, a recall of 86.5%, and a mAP@0.5 detection accuracy of 92.1%. With an object detection speed of 18 FPS and the advantages of the ByteTrack integrated object tracking algorithm, YOLOv8 was selected as the most suitable model. Additionally, the YOLOv8-ByteTrack model, developed for weed detection, was deployed on an agricultural robot with autonomous driving capabilities integrated with ROS. This system facilitates real-time weed detection and destruction, enhancing the efficiency of weed management in agricultural practices.

Keywords: deep learning; weed detection; agricultural robot

check for updates

Academic Editor: Dan Zhang

Received: 17 January 2025
Revised: 17 February 2025
Accepted: 22 February 2025
Published: 7 March 2025

Citation: Bajraktari, A.; Toylan, H. Autonomous Agricultural Robot Using YOLOv8 and ByteTrack for Weed Detection and Destruction. *Machines* **2025**, *13*, 219. https://doi.org/10.3390/machines13030219

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The world's population is growing very fast, and with population growth comes food demand. Therefore, weed control in production areas is becoming increasingly important to increase crop yields and food production [1]. In the past years, the most effective method of weed control was known to be the use of herbicides. However, the use of chemical herbicides has negative effects on the environment and human health [2]. Therefore, the goal of reaching healthy and safe agricultural products has led researchers to develop different weed control methods.

Many researchers are using machine vision-based sensors to collect images and applying artificial intelligence algorithms to perform the necessary image processing and modeling to achieve weed identification with minimal human intervention. Traditional methods use characteristics such as color, shape, texture, and distribution to distinguish between crops and weeds. Hamuda et al. proposed a new algorithm based on color features, morphological erosion, and expansion. The proposed algorithm uses HSV color space to distinguish between crops, weeds, and soil. As a result of their study, a sensitivity of 98.91% and an accuracy of

99.04% were obtained [3]. Cho et al. (2002) used an artificial neural network (ANN) model with eight different shape features for weed detection. This model achieved a detection accuracy of 93.3% for crops and 93.8% for weeds [4]. Bakhshipour et al. identified 14 out of 52 tissue features for weed detection in sugar beet using a feature selection method. These were used as input to the ANN algorithm. The average accuracy of the weed detection algorithm was calculated for 40 test images and was 95.3%. Sugar beet detection accuracy was found to be 91.2% [5]. Traditional methods that use the distributive property are effective when crops are planted at regular intervals along the row. Lottes et al. employed a Bayesian approach, a probabilistic model, to differentiate between sugar beets and weeds. They reported achieving classification success with over 95% accuracy in beet fields [6].

The development of machine learning and deep learning methods in the last decade has increased the tendency toward these methods in solving the crop/weed discrimination problem [7]. Review studies conducted in this field also support this finding [8,9]. Support Vector Machine (SVM) [10], K-Nearest Neighbor (KNN) [11], and Random Forest (RF) classifier [12] are the most prominent machine learning models in solving this problem. Islam et al. compared the performance of machine learning algorithms such as Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) to detect weeds using UAV images obtained from a pepper field. According to their results, the accuracy of RF in weed detection was 96%, SVM 94%, and KNN 63% [11]. Despite the high accuracy of machine learning (ML) methods, significant experimental efforts are needed to extract meaningful features [13]. In contrast, deep learning (DL) eliminates the need for hand-crafted features. Instead, it generates deep representations of input images [14]. Supported by large-scale image data, deep learning algorithms can effectively solve the challenge of precise weed detection. Rahman et al. developed DL-based weed detection models in their study. These models include YOLOv5, Retina Net, Efficient Det, Fast RCNN, and Faster RCNN. Retina Net achieved 79.98% detection accuracy despite having a longer inference time, while YOLOv5, which has the fastest inference time, achieved 76.58% detection accuracy [13]. Hu et al. proposed an innovative graph-based deep learning architecture called Graph Weeds Net (GWN), which aims to recognize multiple weed species using RGB images collected from different areas. This architecture achieved an accuracy rate of 98.1% [15]. Yu et al. compared the performance of deep convolutional neural network (DCNN) models—VGGNet, GoogLeNet, and DetectNet—in detecting emerging weeds in Bermuda grass. The study found that while the VGGNet model achieved impressive results across different mowing heights and surface conditions, DetectNet emerged as the most effective DCNN architecture for detecting a variety of broadleaf weeds alongside dormant Bermuda grass [16]. CNN models employ a two-stage process, making them slightly slower compared to single-stage YOLO models. Hussain et al. collected approximately 24,000 images from potato fields under varying weather conditions, including sunny, cloudy, and partly cloudy environments. These images were tested using YOLOv3 and Tiny-YOLOv3 models for the detection of lamb's quarters weed and potato plants infected with early blight, as well as healthy potato plants. For the weed dataset, the mAP values of the Tiny-YOLOv3 and YOLOv3 models were determined to be 78.2% and 93.2%, respectively [17]. Ruigrok et al. reported that they effectively controlled 96% of weeds and identified potato plants with an accuracy of 84% using the YOLOv3 algorithm. Their study focused on recognizing crops rather than identifying weeds [18]. Junior and Ulson proposed a real-time weed detection system based on the YOLOv5 architecture. Their model was evaluated on a custom dataset consisting of five weed species, both with and without transfer learning. The results demonstrated that the system is functional, achieving a 77% accuracy rate while detecting weeds at 62 FPS [19]. Rehman et al. developed a novel detection model based on the YOLOv5 architecture,

providing a method for distinguishing between soybean plants and weeds. They compared the model's performance against different YOLO versions and the transformer-based RT-DETR, reporting superior results with a mAP of 73.9 [20]. Li et al. developed a weed detection algorithm called YOLOv10n-FCDS, which identified Sagittaria trifolia, a common weed in rice fields, with an accuracy of 87.4%. Based on the obtained results, the rice fields were divided into sections, and customized spray prescriptions were formulated for each section at varying application rates [21].

Improving both detection accuracy and speed simultaneously in the models used by the researchers will positively affect the success of robotic systems. In addition, the development of weeding robot prototypes has progressed significantly, particularly with the use of machine vision for weed detection. Astrand and Baerveldt developed an autonomous agricultural mobile robot that discriminates between sugar beet and weeds with K-Nearest Neighbor classifiers using three features. Although the accuracy rate of the system seems to be satisfactory at 96%, the process of extracting individual plants out of a scene has been conducted manually [22]. Bawden et al. developed a modular agricultural robot platform with a mechanical and chemical weeding system based on machine vision. The performance of different color spaces in weed detection was investigated, and plant species were classified using both local binary patterns and covariance features. They preferred to use covariance features on the robot platform. During the training and testing phases, which were conducted using only 40 images, it was reported that the weed classification system was able to classify broad-leaved and grass plants with 96% accuracy and distinguish individual weed species with 92.3% accuracy [23].

In addition to academic research, company R&D departments are developing robotic agricultural platforms for weed control. One such example is BoniRob (Bosch, Baden-Württemberg, Germany) (Figure 1a) created by Deepfield Robotics (Germany), a Bosch-owned startup [24]. BoniRob can effectively eliminate weeds by burying them deep in the soil, achieving an accuracy rate of 80%. Ecorobotix has developed two different models in this field. The ARA model (Figure 1b) is designed as a high-precision sprayer used behind a tractor [25]. The AVO model (Figure 1c) is an autonomous system powered by solar energy and has the capacity to spray up to 10 hectares per day [26]. Carbon Robotic (Figure 1d) has developed an autonomous laser-powered weed management solution with its agricultural robot platform, arguing that conventional chemicals negatively affect soil health [27].

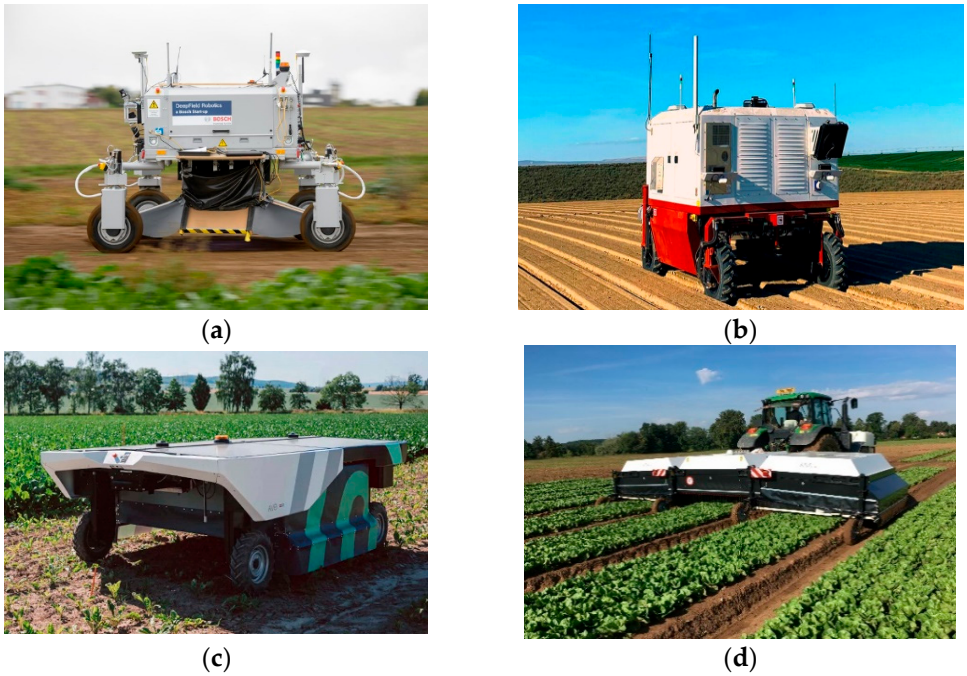
Agricultural research has shown that the use of autonomous and semi-autonomous robots can enhance productivity while also promoting healthy production and environmental sustainability. In this context, the integration of image processing-based deep learning methods into agricultural robots is expected to provide significant contributions.

State-of-the-art weed detection and removal robots are optimized for industrial use with advanced multispectral and hyperspectral cameras, LiDAR systems, high-performance processors, and durable mechanical components. Due to these features, they tend to be highly expensive. Academic research plays a crucial role in the development of commercial robots and paves the way for more efficient, cost-effective, and environmentally friendly solutions in the future.

Despite numerous academic studies on weed detection and destruction robots, ensuring that these systems operate with high accuracy and speed remains a challenging research area that requires further development. This study aims to contribute to this gap by developing a prototype of a low-cost weed detection and removal robot with an accuracy/speed balance by integrating an object-tracking algorithm into a new generation deep learning method.

The main objective of this study is to develop an autonomous agricultural robot capable of real-time differentiation between crops and weeds using the YOLOv8-ByteTrack

deep learning (DL) algorithm without feature extraction. The low-cost prototype system successfully distinguished weeds in real time. Additionally, various YOLO deep learning models were compared, and the results were presented. One of the objectives of this study is to minimize the negative impact of chemical herbicides on soil and water resources by replacing chemical spraying with laser-based weed control. This approach contributes significantly to sustainable agricultural practices by targeting only the necessary areas for intervention.



(**a**) (**b**) (**c**) (**d**)

Figure 1. Machine vision-based weeding robots: (**a**) the Bonirob, (**b**) the ARA, (**c**) the AVO, (**d**) the Laserweeder.

2. Materials and Methods

The proposed autonomous agricultural robot is composed of two main parts: an object detection/tracking part and an autonomous navigation part. An overview of the autonomous agricultural robot for weed detection and its block diagram are presented in Figure 2 and Figure 3, respectively.

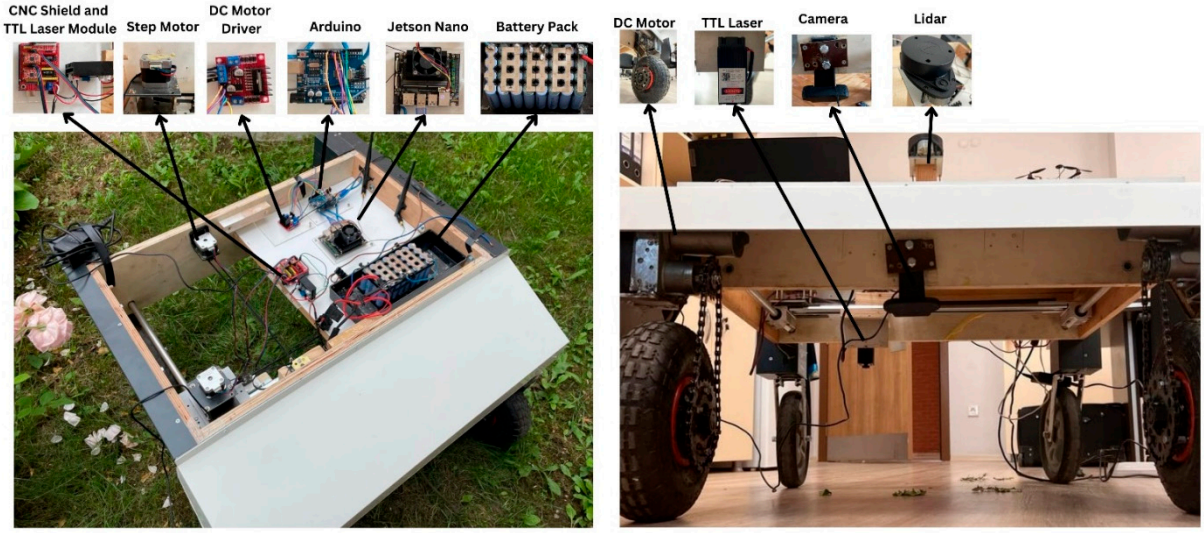


Figure 2. Overview of the autonomous agricultural robot.

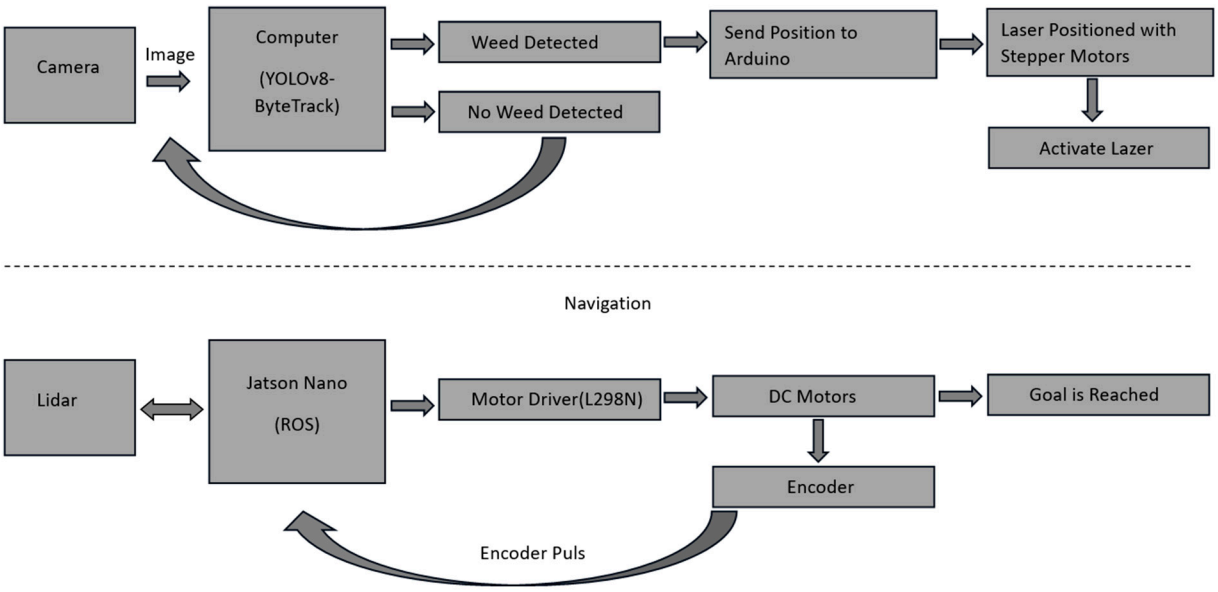


Figure 3. Block diagram of the autonomous agricultural robot.

2.1. Autonomous Navigation

2.1.1. Differential Drive

The agricultural robot utilizes a differential drive system, which enables movement and rotation by varying the speed of its two motors. It is equipped with 36 V DC motors and a gearbox, along with an additional 1:2 ratio gearbox connected by a chain. The differential drive can be characterized by the robot's X and Y positions on a plane and angular position, θ, as illustrated in Figure 4.

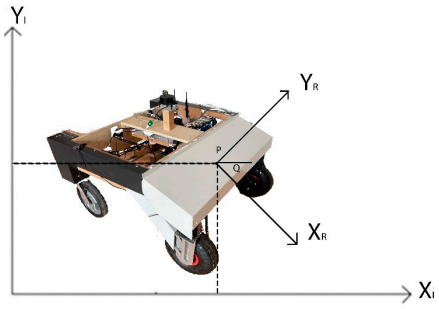


Figure 4. Position of the autonomous agricultural robot.

The position of a mobile robot can be described by two systems, inertial and robot coordinate. The inertial coordinate system is a global frame fixed in the environment in which the robot moves. The robot coordinate system is a local frame that is attached to the autonomous agricultural robot and moves with it. {XI, YI} Inertial fixed reference frame and {XR, YR} the relationship between the Robot Frame is represented by the following matrix. The expression P [x y θ] shows the position of the robot in the cartesian coordinate system [28]:

$$\varepsilon_R = R(\theta) \in_1 \quad (1)$$

$$= R(\theta). \begin{bmatrix} . & . & . \\ x & y & \theta \end{bmatrix}^T \quad (2)$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where (x, y) denotes the coordinates of the center point of the two wheels. Q denotes the orientation of the rear two movable wheels. v represents the linear velocity of the robot; ω represents the angular velocity of the robot and represents the traction wheel radius [29].

$$\left\{ \begin{array}{l} v = \frac{r_R\omega_R + r_L\omega_L}{2} \\ \omega = \frac{r_R\omega_R - r_L\omega_L}{b} \end{array} \right\} \quad (4)$$

where ω_R and ω_L are the speeds of the right and left wheel, r_R and r_L are the radii of the right and left wheel, respectively, and b is the distance between both drive wheels and the axis of symmetry. Equation (5) can be rewritten as follows [29]:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r_R}{2} & \frac{r_R}{2} \\ \frac{r_R}{b} & \frac{r_L}{b} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (5)$$

2.1.2. Localization and Mapping

Localization and mapping of the agricultural robot are based on the distributed framework of the ROS (Robot Operating System). As an open-source software library and framework, ROS supports the development of robot applications and can operate across multiple machines. In ROS, files compatible with the system are referred to as packages, and a collection of these packages is known as a stack; stacks can be shared and distributed among users. The ROS framework can be programmed using popular programming languages such as C++ 14 and Python 3.9. Firstly, the ROS compatible with the Linux operating system was installed on the system computer. Then, localization and mapping operations of the autonomous agricultural robot using 2D lidar were performed with the SLAM (Simultaneous Localization and Mapping) application via ROS. In this robot for autonomous navigation, we used packages such as Ros Serial (https://github.com/ros-drivers/rosserial), RF2O Package (https://github.com/MAPIRlab/rf2o_laser_odometry), Hector SLAM (https://github.com/tu-darmstadt-ros-pkg/hector_slam), Gmapping (https://github.com/ros-perception/slam_gmapping), and Navigation Stack (https://github.com/ros-planning/navigation, all accessed on 17 January 2025).

Lidar plays a critical role in the process of acquiring depth information by providing high measurement accuracy and fine angular resolution. Various studies have been conducted to enhance the 3D imaging performance of Lidar technology. For example, since obtaining a depth map requires multiple reconstruction calculations, CS-based dual-frequency laser 3D imaging methods, which work with only two calculations, have been preferred by researchers [30]. Additionally, Lidar and camera fusion-based systems have demonstrated effective performance in extracting 3D contours [31]. In this study, depth information between the agricultural robot and obstacles was obtained using Lidar, and this information proved to be sufficient for the robot to avoid obstacles and create a map.

Many SLAM algorithms can be utilized on the ROS platform. In the experiments conducted by Santos et al. [32], the Gmapping algorithm demonstrated a particularly low error rate and CPU overhead. This algorithm employs the Particle Filtering (PF) probabilistic approach, which is grounded in the Sequential Monte Carlo Technique for sampling particle distributions. The Sequential Monte Carlo Technique estimates the robot's position using data gathered from sensors [33]. Additionally, the Gmapping algorithm integrates both scan matching and odometry information to reduce the number of particles. Figure 5 presents the flowchart of autonomous navigation.

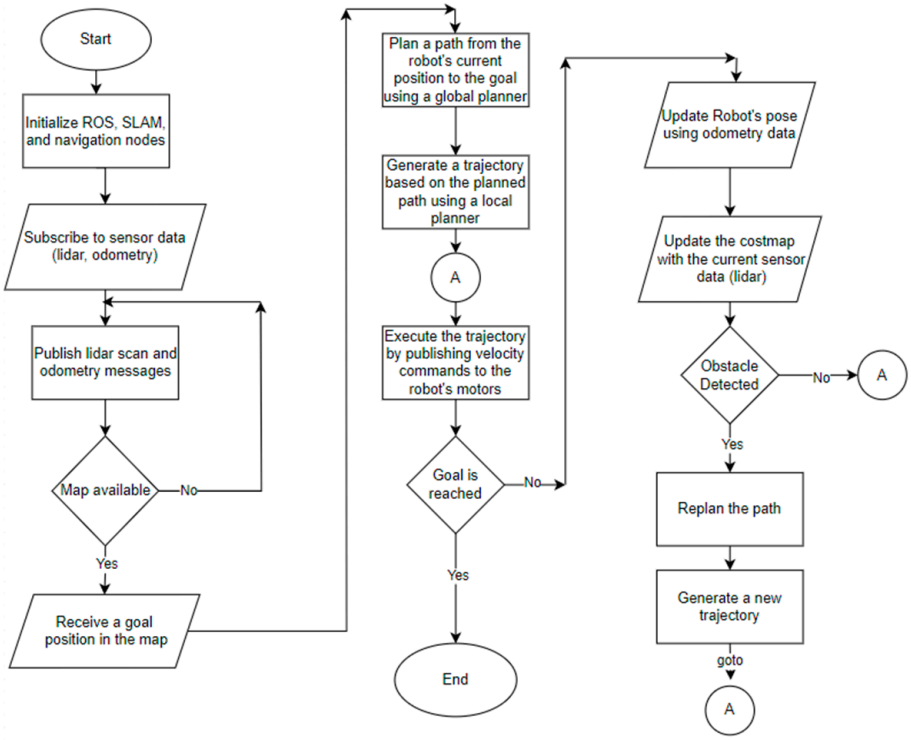


Figure 5. Flowchart of autonomous navigation part.

2.2. Weed Detection and Tracking

2.2.1. YOLO (You Only Look Once)

YOLO (You Only Look Once) emerged in the field of computer vision with a paper published in 2015 by Joseph Redmon et al. [34]. YOLO is a one-stage deep learning algorithm using a convolutional neural network with a high success rate in accurately detecting objects in image and video data [35,36]. Its ability to perform detection with a single propagation makes it ideal for real-time applications requiring high detection speeds [37]. The YOLO algorithm decomposes the image and passes it through a convolutional neural network in a single pass, generating a result vector about the image. Initially, the entire image is divided into cells of size m × n. Each cell is analyzed by the convolutional neural network to detect objects within it and draw bounding boxes around them. During this process, the algorithm checks whether the object's center point falls within the cell. If the midpoint of the object is inside the cell, the center, height, width, and confidence score (x, y, w, h, and a confidence score) of the detected object are calculated. To determine the best bounding box, the Intersection Over Union (IOU) metric is applied between the cells.

Following the release of the first YOLO version, YOLOv2 [38], YOLOv3 [39], and YOLOv4 [40], nine models were released in 2016, 2018, and 2020, respectively. In 2020, YOLOv5 [41] introduced by a company called Ultralytics, was released on a GitHub (https://github.com/) repository by the company's founder and CEO, Glenn Jocher. In 2022, YOLOv6 [42] was developed by Meituan researchers, followed by YOLOv7 [43]. Finally, in 2023, YOLOv8 [44] was released by Ultralytics. After the end of the experimental tests of this study, YOLOv9 [45], YOLOv10 [46], and YOLOv11 [47] were published, respectively.

The loss function in the YOLO algorithm can be analyzed under three main categories: location loss, confidence loss, and classification loss. The sum of these losses constitutes the overall loss function. The formula is as follows [48]:

$$\begin{aligned}
&\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\|_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]\\
&+\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\|_{ij}^{obj}\left[\left(\sqrt{\omega_i}-\sqrt{\hat{\omega}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]\\
&+\sum_{i=0}^{s^2}\sum_{j=0}^{B}\|_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2\\
&+\lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\|_{ij}^{nobj}\left(C_i-\hat{C}_i\right)^2\\
&+\sum_{i=0}^{s^2}\|_{i}^{obj}\sum_{c\in classes}\left(p_i(c)-p_i\hat{(c)}\right)^2
\end{aligned} \tag{6}$$

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{J=0}^{B} \tag{7}$$

The symbols x, y, w, h, c, and p define the center coordinates, width, height, confidence level, and category probability of the bounding box, respectively. The hat (ˆ) symbol in the formulae refers to the estimated values. λcoord is the highest in order to have more importance in the first term.

2.2.2. YOLO Models

This study is based on YOLOv5 and YOLOv8 algorithms for weed detection. YOLOv5 was inspired by YOLOv3 and YOLOv4 and the arithmetic set feature was added to provide a more effective detection. YOLOv5 uses the Auto Anchor algorithm instead of manually determined anchor boxes to increase the speed of the R-CNN algorithm. In addition, K-means are applied to the size of the bounding boxes to obtain a better priority value. The YOLOv5 architecture consists of three main components: Backbone, Neck, and Head. The architecture of YOLOv5 is shown in Figure 6.

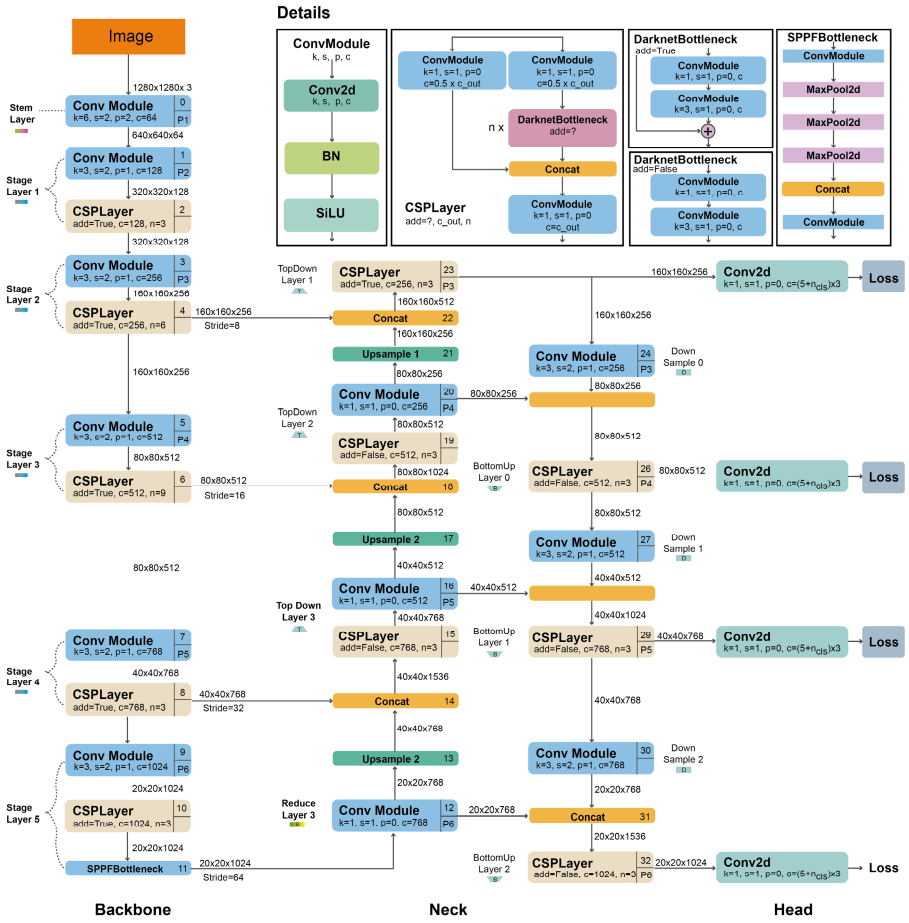


Figure 6. YOLOv5 architecture [49].

YOLOv8 features a backbone akin to the C2f module found in YOLOv5, with some modifications to the CSPLayer. This model employs an anchor-free detector that processes box prediction, classification, and regression tasks independently. The output layer of YOLOv8 utilizes a sigmoid activation function to generate the objectivity score, indicating the probability of an object being present within the bounding box [29,41]. Additionally, it adopts CioU for bounding box loss and binary cross-entropy for classification loss, enhancing the detection of small objects [49]. Figure 7 illustrates the architecture of YOLOv8.

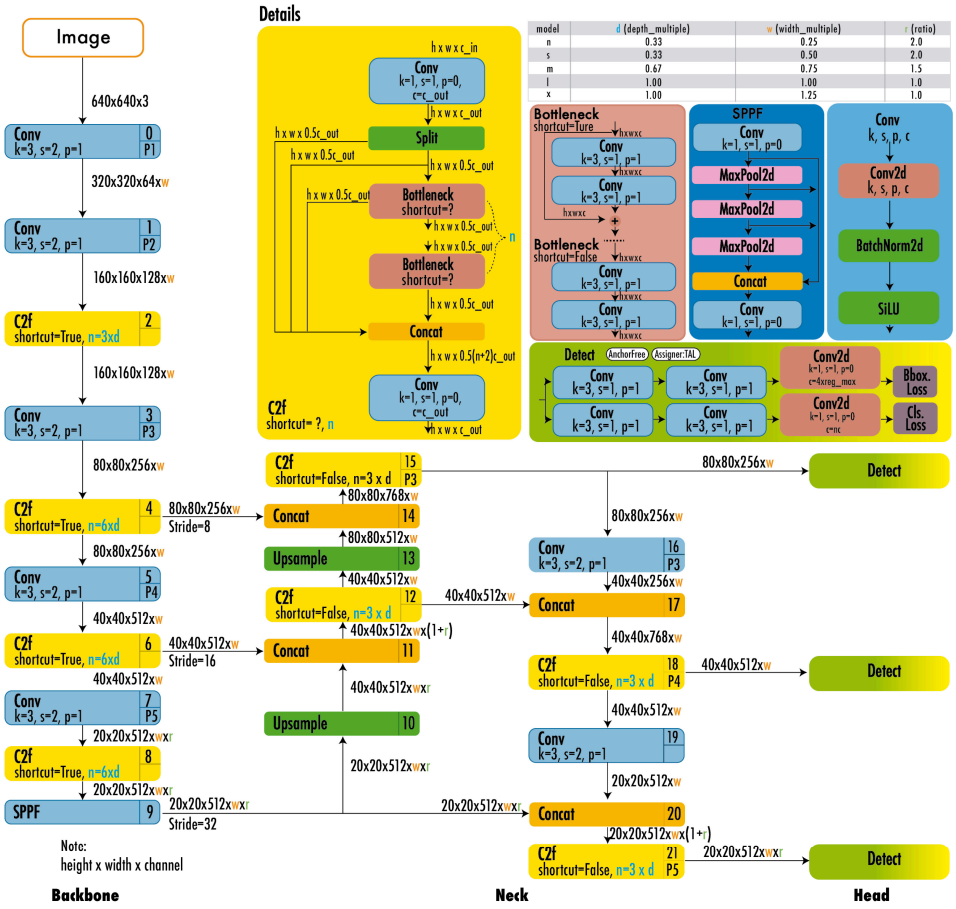


Figure 7. YOLOv8 architecture [49].

2.2.3. Pruning and Quantization

DeepSparse is a CPU inference tool that reduces the computation time required to increase the inference speed of the neural network. Due to this advantage, DeepSparse has been stated to offer a 5.8 times speedup for YOLOv5s running on the same system [47]. Pruning reduces the number of parameters in the model by removing less significant weights, while quantization compresses the model by reducing the bit precision of weights and activations (e.g., from 32-bit floating-point to 8-bit integer). These modifications balance model size, processing speed, and detection accuracy. While a slight reduction in accuracy is observed, the model continues to maintain a high detection rate. This indicates that the effectiveness of pruning and quantization is not significantly compromised. One of the greatest advantages of pruning and quantization is the notable increase in inference speed. This improvement in processing speed enhances the system's real-time capabilities, allowing the robot to process more frames per second and make faster decisions an essential factor for dynamic agricultural environments. Pruning and quantization techniques can be applied in YOLOv5 versions to increase the extraction speed without loss of accuracy. In this study, quantization and pooling operations are performed on the lean YOLOv5 model

trained on our dataset using the SparseML library (https://github.com/neuralmagic/sparseml, accessed on 17 January 2025). In addition, the performance of the different sparse models (quantization and pooling) obtained through transfer learning is also compared.

2.2.4. Weed Tracking with ByteTrack

Multi-object tracking algorithms are essential for detecting and effectively tracking the movements of multiple objects. They play a crucial role in machine vision, particularly in monitoring several objects in recorded images or live streams. Commonly used methods in object tracking algorithms include ByteTrack, SORT, DeepSORT, and StrongSORT [50].

The ByteTrack object tracking algorithm has low hardware requirements due to its simple structure, which enables fast processing. By incorporating objects with low confidence scores into its evaluation, it enhances overall tracking accuracy. This capability allows ByteTrack to achieve an effective balance between tracking speed and accuracy [51]. For example, ByteTrack consistently outperformed SORT and DeepSORT in detecting vehicles and people in highway timelapse videos [52]. Moreover, object tracking algorithms such as DeepSORT, StrongSORT, DeepOCSORT, and BOT-SORT employ the Re-Identification (ReID) method, which significantly reduces the network's processing speed [51].

As a result of the literature review, the ByteTrack algorithm, proposed by Zhang et al. [53]. In 2021, was selected as the object-tracking algorithm for this study. ByteTrack differentiates itself in the field of object tracking, treating each detection as a byte. This algorithm allows objects with low detection scores to be recovered through a secondary association pass, without being limited to high-scoring bins, thus providing a more robust association process. ByteTrack's simple structure is a key advantage; it relies on precise detections and byte-level association, avoiding complex appearance and motion models [54]. ByteTrack workflow is shown in Figure 8.

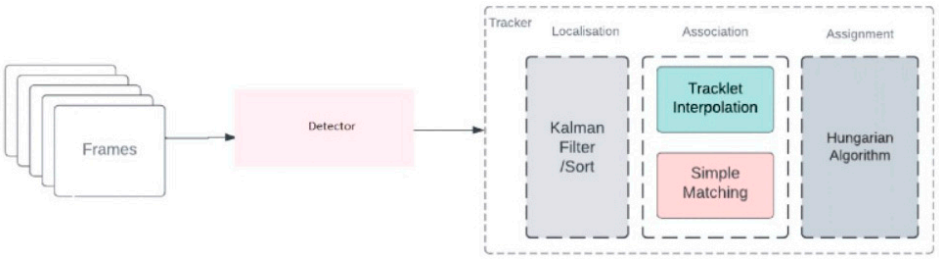


Figure 8. ByteTrack workflow [55].

2.3. Metrics

In order to evaluate the accuracy of the YOLO models used in this study, several important metrics were determined: recall, precision, and mean average precision (mAP). In addition, in real-time systems, the frame rate (FPS—Frames Per Second) is a factor that directly affects system performance. While increasing the model performance, it is also important that the FPS value ensures the sustainability of the real-time system.

Sensitivity measures the model's ability to accurately predict true positive values, while precision refers to the rate at which all classes are correctly predicted. The mAP indicates the accuracy score between the true bounding box and the detected box. A high score indicates that the model provides high accuracy in its detections.

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$F1score = 2 \times \frac{P \times R}{P + R} \tag{10}$$

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{\sum_{i=1}^{n} AP(i)}{n} \tag{12}$$

True positives (TP) refer to instances where both the predicted results and the ground truth values are positive samples. False positives (FP) occur when the predicted result is positive, but the ground truth value is negative. False negatives (FN) indicate cases where the predictions are negative, while the ground truth is positive. AP(i) refers to the average accuracy for the current species, and n signifies the number of categories [56].

3. Results and Discussion

The dataset used in this study was obtained from the Roboflow Weeds Dataset, a publicly available dataset that includes images of different weed species captured in diverse field conditions. The dataset contains 3926 images of weeds commonly found in agricultural settings [57]. An additional 200 images of weeds were incorporated into the dataset to enhance the model's performance in its intended environment. In all datasets, Dandelion, Heliotropium indicum, Young Field Thistle (Cirsium arvense), Plantago lanceolata, Eclipta, and Urtica dioica were identified as significant weed species, and examples of images are depicted in Figure 9.

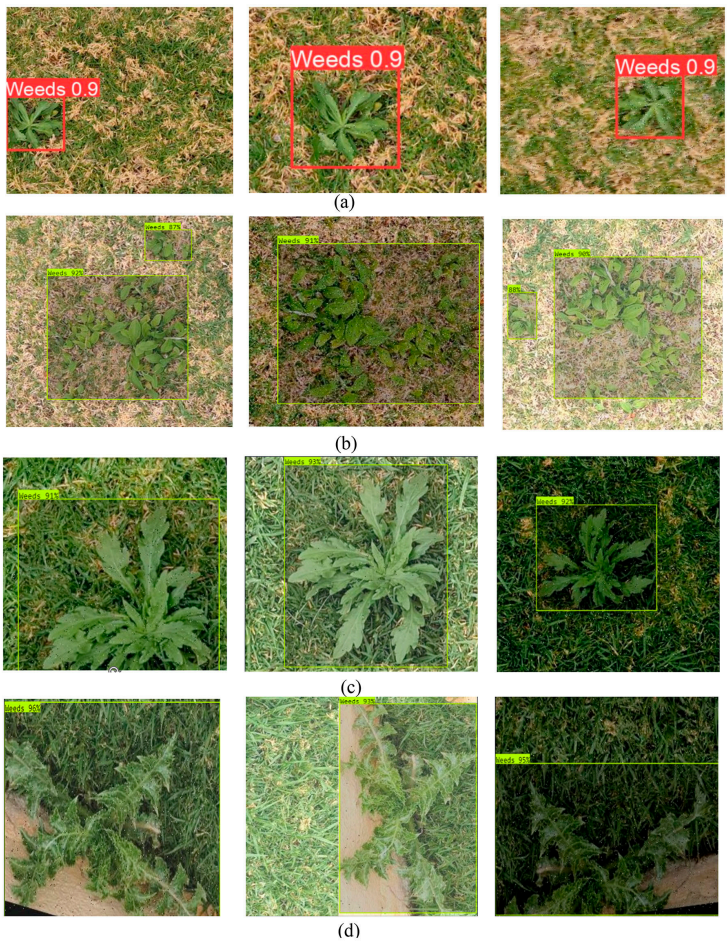


Figure 9. *Cont.*

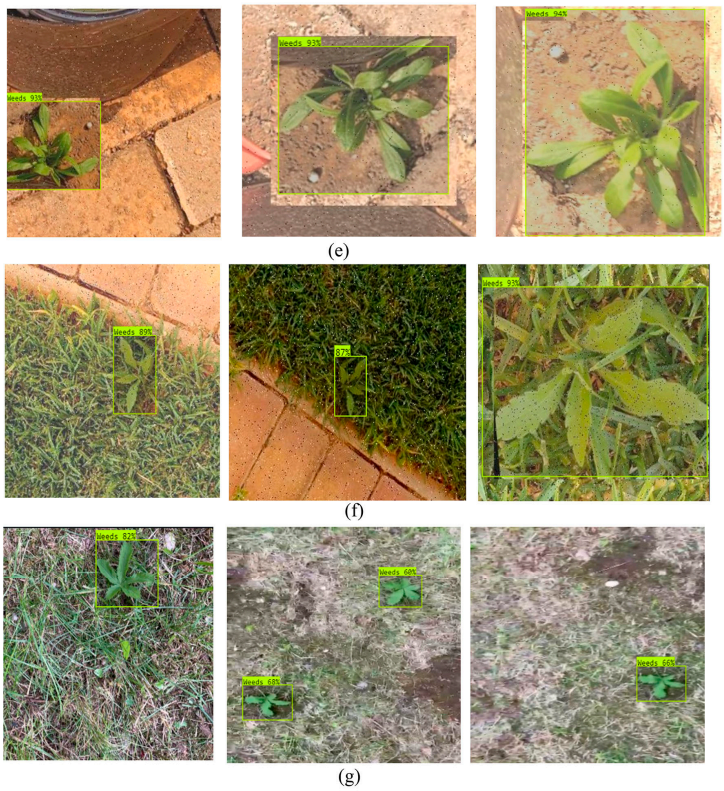


Figure 9. Types of weeds: (**a**) Dandelion Weeds, (**b**) Heliotropium indicum, (**c**) Young field Thistle Cirsium arvense, (**d**) Cirsium arvense, (**e**) Plantago lanceolata, (**f**) Eclipta, (**g**) Urtica Diocia.

To ensure the robustness of the deep learning model, images were collected under various conditions. The dataset includes images taken in bright sunlight, partial shade, and low-light conditions, simulating real-world agricultural environments. Some images were captured under dry, wet, and cloudy conditions, which affect the contrast and appearance of weeds. Images include different angles, soil textures, and crop backgrounds to improve model generalization.

Augmentation techniques were used to further enhance the dataset's variability and improve the model's ability to generalize, including brightness and contrast adjustment, rotation and flipping, Gaussian noise injection, scaling and cropping, and color jittering. These augmentations were performed using Roboflow's built-in augmentation pipeline and additional processing in Python using the albumentations library.

The implementation of these techniques will positively impact the three key performance aspects of deep learning algorithms. The use of diverse lighting conditions and augmentations helps the model recognize weeds in different scenarios, reducing overfitting. By simulating real-world field conditions, the model can effectively differentiate weeds from crops, even under challenging circumstances. Data augmentation improves mAP scores by exposing the model to multiple variations, making it more resilient to unseen images.

In this experimental study, six different YOLO models (the YOLOv5, the YOLOv5 Pruned and Quantized, the YOLOv5 Pruned and Quantized with Transfer Learning, the YOLOv8, the YOLOv9, and the YOLOv11) were trained and tested for the detection of seven different weed species (Dandelion weeds (Figure 9a), Heliotropium indicum (Figure 9b), Young field thistle cirsium arvense (Figure 9c), Cirsium arvense (Figure 9d), Plantago lanceolata (Figure 9e), Eclipta (Figure 9f), and Urtica diocia (Figure 9g)). The training of the models was performed over 100 epochs.

Results of YOLO Models

When the results of the YOLOv5 model are analyzed, the mAP@0.5 value is 0.930, the recall value is 0.884, and the precision value is 0.938. When the model was run on the CPU, it reached approximately 15 fps., which can be seen in Figure 10.

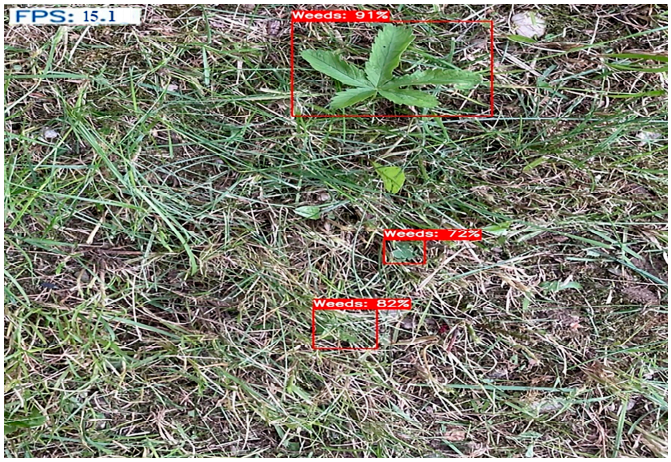


Figure 10. Results for the YOLOv5 model on image.

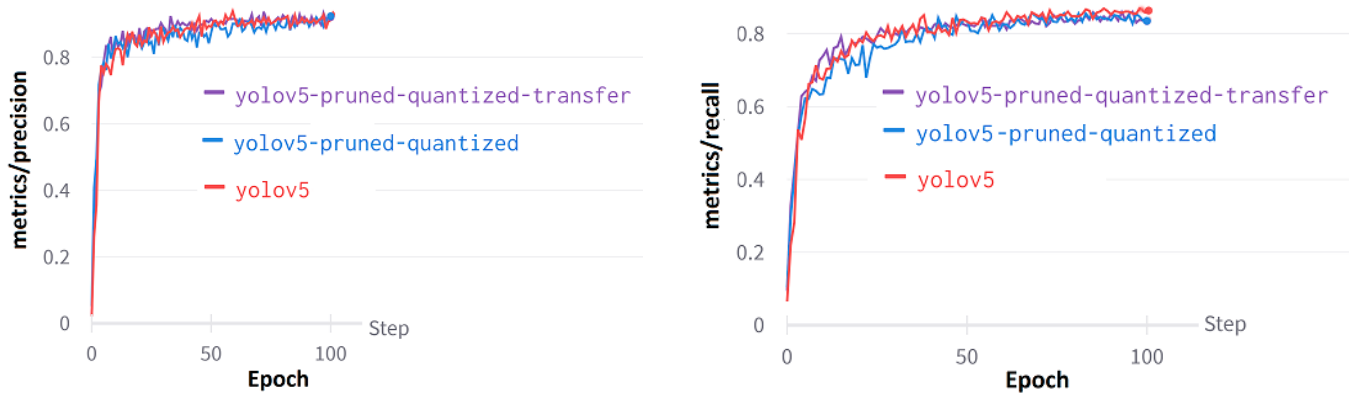
The model can be optimized and pre-diluted (pruned and quantified) by adapting it to the dataset via SparseML. This process is called Sparse Transfer Learning. The YOLOv5 Pruned and Quantized with Deepsparse achieved an mAP@0.5 value of 0.909, a recall value of 0.834, and a precision value of 0.921. Figure 11a shows that the fps reached 47.1 on the CPU. The YOLOv5 model was pruned and quantized to improve fps. As a result, the fps increased to 116 (Figure 11b) on the CPU, while mAP, precision, and recall values remained largely unchanged. The mAP@0.5 value for this model is 0.909, the recall value is 0.835, and the precision value is 0.924.



(**a**) (**b**)

Figure 11. (**a**) Results of the YOLOv5 Pruned and Quantized with Transfer Learning, (**b**) Result of the YOLOv5 Pruned and Quantized.

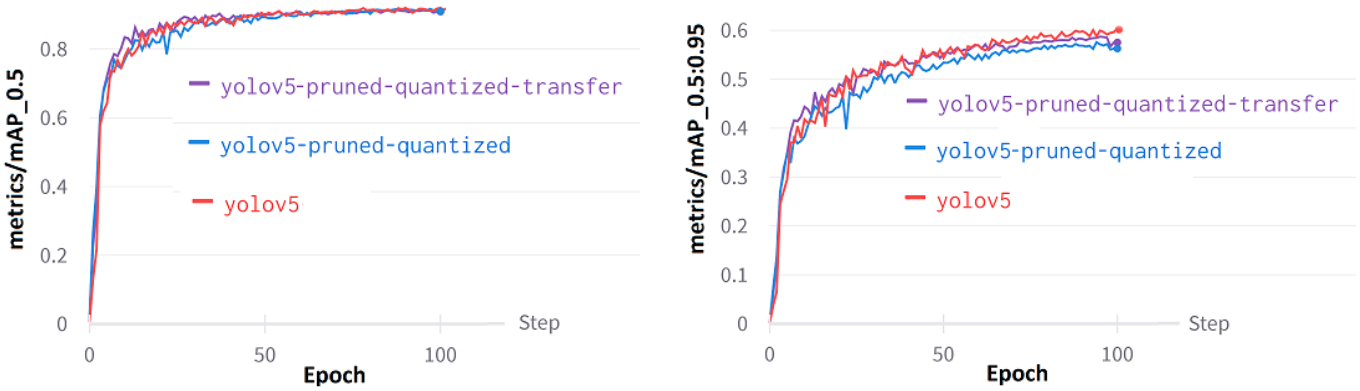
Figure 12a shows the precision curve for YOLO models: the YOLOv5, the YOLOv5 Pruned and Quantized, and the YOLOv5 Pruned and Quantized with Transfer Learning models. When pruning and quantification were applied to the YOLOv5 model, a decrease in the positive predictive value was observed. The recall value, defined as the probability of detection, is higher in the YOLOv5 model compared to the other two models. Change in recall values in epochs can be seen in Figure 12b.



(**a**) (**b**)

Figure 12. Performance curves of YOLOv5: (**a**) Metrics/precision curves, (**b**) Metrics/recall curves.

Mean Average Precision (mAP) is essential for model evaluation because it offers a comprehensive metric that encompasses both precision and recall across various weed classes. The mAP@0.5 (Figure 13a) evaluates the ability of models to accurately detect weeds by measuring sensitivity at a threshold of 0.5 IoU. On the other hand, mAP@0.5:0.95 (Figure 13b) comprehensively evaluates the detection of weeds by averaging the sensitivities obtained at various IoU thresholds. The mAP@0.5:0.95 value for the YOLOv5 without pruning and quantization is 0.6473. In contrast, the pruned and quantized model with transfer learning has an mAP@0.5:0.95 value of 0.587. Finally, the pruned and quantized model has an mAP@0.5:0.95 value of 0.571.



(**a**) (**b**)

Figure 13. Performance curves of YOLOv5: (**a**) Metrics/mAP@0.5, (**b**) metrics/mAP@0.5:0.95.

Despite parameter reduction, it was observed that pruned and quantized models did not have a significant negative impact on robustness. Overall, the models maintained high detection accuracy despite variations in field conditions, lighting, and weed species.

The developed agricultural autonomous robot uses a model trained with YOLOv8 for object tracking. The purpose of the tracking algorithms is not to reevaluate the weed image that has completed the detection process. It is seen as an important process to burn the areas identified as weeds with laser and to avoid wasting time by burning this area again. SORT, DeepSORT, ByteTrack, and BotSort applications were tested for object tracking in this model. As a result of the comparisons made by the trial and error method, the ByteTrack application showed the best performance. Sort and BotSort tracking algorithms were not preferred because they are outdated and increase the processing time. For this reason, YOLOv8 ByteTrack, the tracking algorithm of the YOLO model, was used.

The performance of YOLOv8 is shown in Figure 14. The mAP@0.5 value of this model is 0.921, the recall rate is 0.865, and the precision value is 0.938. This model, running on a GPU, performed at an average speed of 18 fps.

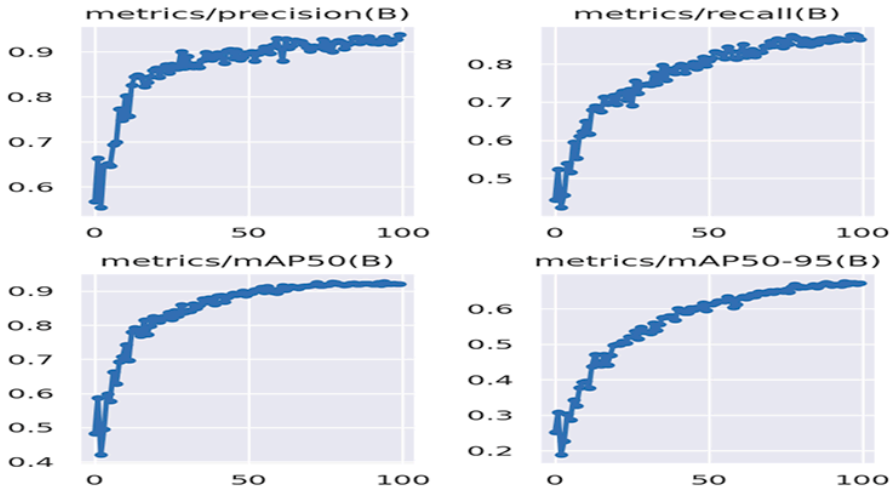


Figure 14. Performance results of YOLOv8.

YOLOv9 introduces modifications to the network architecture, particularly in feature extraction and optimization strategies, aiming to improve detection speed while maintaining high accuracy. The mAP@0.5 value of this model is 0.893, the recall rate is 0.803, and the precision value is 0.905. The model performed well in detecting weeds but exhibited a slightly lower recall, indicating a higher rate of false negatives compared to other models. Despite this, it maintained a strong balance between precision and detection speed.

YOLOv11 builds on advancements from previous YOLO versions, incorporating improved object localization techniques and an enhanced detection backbone. The mAP@0.5 value of this model is 0.926, the recall rate is 0.870, and the precision value is 0.915. The model demonstrated strong performance in weed detection in terms of both precision and detection speed. However, compared to other models, except for YOLOv9, it exhibited slightly lower precision due to a higher false positive rate. The YOLO models were tested with the generated data set. The performance comparisons of these models are given in Table 1.

Table 1. Results of the comparative performance analysis of YOLO models.

Model	mAP	Precision	Recall	Fps
YOLOv5	0.930	0.938	0.884	15
YOLOv5 Pruned and Quan. with Tf.	0.909	0.921	0.834	47
YOLOv5 Pruned and Quan.	0.909	0.924	0.835	116
YOLOv8 ByteTrack	0.921	0.938	0.865	18
YOLOv9	0.893	0.905	0.803	21
YOLOv11	0.926	0.915	0.870	24

The models presented in Table 1 were tested on a laptop equipped with an NVIDIA GTX 1650 (4GB VRAM) (NVIDIA Corporation, Location: Santa Clara, CA, USA) and an Intel i5 processor with 16GB RAM (Intel Corporation, Location: Santa Clara, CA, USA). Comparing these models based only on performance metrics can lead to errors in real-time systems. Therefore, the detection speed (fps) of the model is also important for system performance. To enhance the detection accuracy, pruned and quantized processes were

implemented, resulting in a substantial increase in fps from 15 to 116. However, this improvement came at the cost of a reduction in the model's detection accuracy. As a result, considering both accuracy and speed performance, YOLOv8 was preferred for use in the autonomous agricultural robot system. Additionally, the integration of YOLOv8 with the ByteTrack object tracking feature further supports this decision. YOLOv8-ByteTrack achieved a frame rate of 18 FPS, meaning that processing a new frame took an average of 55 ms. When head tracking was performed using ByteTrack, an additional latency of approximately 5 ms per frame was introduced, resulting in a total delay of around 60 ms. In tests conducted solely on the CPU (Intel i5 without GPU acceleration), the FPS dropped to a range of 4–6, significantly limiting the practicality of real-time deployment on a CPU. In terms of computational costs, running YOLOv8-ByteTrack on the GTX 1650 utilized approximately 75–85% of the GPU resources. Memory consumption ranged between 3 GB VRAM and 5–7 GB RAM. Under load, the system's power consumption was estimated to be around 40 W–50 W.

Some recent studies have been carried out on the detection weed performance of models such as Support Vector Machine (SVM) [10], YOLO [35], Faster R-CNN [58], and RetinaNet [59]. While considerable efforts have been made on weed detection, very few studies were focused on the detection and speed performance together. This study sets itself apart from previous research by not only comparing model performances based on existing or acquired images but also by implementing the proposed model within a machine vision hardware prototype.

The deep learning model based on ByteTrack and YOLOv8 is designed for generalization across multiple environments. However, its adaptation to new agricultural settings can be further improved through several approaches. First, the model can be retrained using novel datasets of weeds and crops specific to a particular region or agricultural application. Alternatively, transfer learning can be utilized to leverage the pre-trained model without the need for full retraining, enabling the identification of new weed species. Transfer learning allows the model to efficiently adapt to new weed types or different crop species without requiring training from scratch. For instance, if the model was initially trained to detect seven different weed species, it can be fine-tuned using new images to accommodate fields with different weed compositions while preserving learned representations of plant structures. If a new weed species emerges in the field, a small, labeled dataset containing images of this weed can be used for incremental learning to enhance recognition. Instead of retraining the entire network, transfer learning enables fine-tuning of only the last few layers, making adaptation more efficient.

4. Conclusions

Weeds can negatively impact crop development, so it is important to keep them under control. However, the presence of various weed species and the resemblance of some species to crops complicate the identification process. This study integrates ByteTrack, an effective object-tracking system for the detection and eradication of weeds in agricultural areas by considering seven different weed species under a single category with the deep learning algorithm YOLOv8 in a hardware prototype. The experimental results indicate that the ByteTrack-YOLOv8 model achieved a detection speed of 18 FPS and a mAP@0.5 value of 92.1%. This model was compared with six different YOLO models using the same dataset. The proposed model achieved a detection accuracy slightly lower than YOLOv5. However, for this study, the YOLOv8 model was favored due to detection speed and object tracking capabilities. Although modified YOLOv5 models did improve detection speed, this enhancement came at the expense of detection accuracy, making them less desirable.

As a result, the ByteTrack-YOLOv8 model provides satisfactory results in weed detection and tracking, along with strong hardware compatibility.

One of the primary limitations of this study is the high computational power required by the YOLO model due to its operation with large-scale datasets. This challenge has made it difficult to run the model on the embedded system (Jetson Nano). Additionally, the model used in the agricultural robot was developed to recognize seven common weed species. However, the dominant weed species may vary across different geographical regions. Therefore, to ensure the model adapts to local weed species, it must be retrained using an appropriate dataset that accounts for variations in weather and lighting conditions.

Agricultural robots can detect not only weeds but also harmful insects, diseases, and the ripeness level of crops. This enables the development of a more efficient and sustainable agricultural system. Furthermore, the integration of artificial intelligence and agricultural robot technologies, as demonstrated in this study, facilitates the development of environmentally friendly applications.

In future studies, it is recommended to use BLDC motors instead of DC motors to enhance the autonomous driving performance of agricultural robots. Additionally, integrating the IMU with wheel encoders and the Extended Kalman Filter will allow for more accurate odometry readings. Along with these hardware upgrades, several approaches can be applied to improve the performance of YOLO, such as enhancing the model architecture and optimizing the training processes. These improvements could make YOLO more effective, especially in real-time and large-scale applications.

Author Contributions: Conceptualization, A.B. and H.T.; data curation, A.B.; writing—original draft preparation, A.B. and H.T.; writing—review and editing, A.B. and H.T.; Supervision, A.B. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by TUBITAK (The Scientific and Technological Research Council of Turkey) with 2209-A—Research Project Support Program.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. FAO. NSP-Weeds. Available online: http://www.fao.org/agriculture/crops/thematic-sitemap/theme/biodiversity/weeds/en/ (accessed on 8 August 2023).
2. Babalola, O.O.; Truter, J.C.; Van Wyk, J.H. Lethal and teratogenic impacts of imazapyr, diquat dibromide, and glufosinate ammonium herbicide formulations using frog embryo teratogenesis assay-xenopus (FETAX). *Arch. Environ. Contam. Toxicol.* **2021**, *80*, 708–716. [CrossRef] [PubMed]
3. Hamuda, E.; Mc Ginley, B.; Glavin, M.; Jones, E. Automatic crop detection under field conditions using the HSV colour space and morphological operations. *Comput. Electron. Agric.* **2017**, *133*, 97–107. [CrossRef]
4. Cho, S.; Lee, D.S.; Jeong, J.Y. AE—Automation and emerging technologies: Weed–plant discrimination by machine vision and artificial neural network. *Biosyst. Eng.* **2002**, *83*, 275–280. [CrossRef]
5. Bakhshipour, A.; Jafari, A.; Nassiri, S.M.; Zare, D. Weed segmentation using texture features extracted from wavelet sub-images. *Biosyst. Eng.* **2017**, *157*, 1–12. [CrossRef]
6. Lottes, P.; Stachniss, C. Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 5155–5161. [CrossRef]
7. Li, N.; Zhang, X.; Zhang, C.; Ge, L.; He, Y.; Wu, X. Review of machine-vision-based plant detection technologies for robotic weeding. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 2370–2377. [CrossRef]
8. Hasan, A.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [CrossRef]

9. Rai, N.; Zhang, Y.; Ram, B.G.; Schumacher, L.; Yellavajjala, R.K.; Bajwa, S.; Sun, X. Applications of deep learning in precision weed management: A review. *Comput. Electron. Agric.* **2023**, *206*, 107698. [CrossRef]
10. Bakhshipour, A.; Jafari, A. Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Comput. Electron. Agric.* **2018**, *145*, 153–160. [CrossRef]
11. Islam, N.; Rashid, M.M.; Wibowo, S.; Xu, C.Y.; Morshed, A.; Wasimi, S.A.; Rahman, S.M. Early weed detection using image processing and machine learning techniques in an Australian chilli farm. *Agriculture* **2021**, *11*, 387. [CrossRef]
12. Haug, S.; Michaels, A.; Biber, P.; Ostermann, J. Plant classification system for crop/weed discrimination without segmentation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March 2014; pp. 1142–1149. [CrossRef]
13. Rahman, A.; Lu, Y.; Wang, H. Performance evaluation of deep learning object detectors for weed detection for cotton. *Smart Agric. Technol.* **2023**, *3*, 100126. [CrossRef]
14. Hu, K.; Wang, Z.; Coleman, G.; Bender, A.; Yao, T.; Zeng, S.; Walsh, M. Deep Learning Techniques for In-Crop Weed Identification: A Review. *arXiv* **2021**, arXiv:2103.14872. Available online: http://xxx.lanl.gov/abs/2103.14872 (accessed on 12 March 2023).
15. Hu, K.; Coleman, G.; Zeng, S.; Wang, Z.; Walsh, M. Graph weeds net: A graph-based deep learning method for weed recognition. *Comput. Electron. Agric.* **2020**, *174*, 105520. [CrossRef]
16. Yu, J.; Sharpe, S.M.; Schumann, A.W.; Boyd, N.S. Deep learning for image-based weed detection in turfgrass. *Eur. J. Agron.* **2019**, *104*, 78–84. [CrossRef]
17. Hussain, N.; Farooque, A.A.; Schumann, A.W.; McKenzie-Gopsill, A.; Esau, T.; Abbas, F.; Zaman, Q. Design and development of a smart variable rate sprayer using deep learning. *Remote Sens.* **2020**, *12*, 4091. [CrossRef]
18. Ruigrok, T.; van Henten, E.; Booij, J.; van Boheemen, K.; Kootstra, G. Application-specific evaluation of a weed-detection algorithm for plant-specific spraying. *Sensors* **2020**, *20*, 7262. [CrossRef]
19. Junior, L.C.M.; Ulson, C.A.J. Real time weed detection using computer vision and deep learning. In Proceedings of the 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 15–18 August 2021; pp. 1131–1137. [CrossRef]
20. Rehman, M.U.; Eesaar, H.; Abbas, Z.; Seneviratne, L.; Hussain, I.; Chong, K.T. Advanced drone-based weed detection using feature-enriched deep learning approach. *Knowl.-Based Syst.* **2024**, *305*, 112655. [CrossRef]
21. Li, Y.; Guo, Z.; Sun, Y.; Chen, X.; Cao, Y. Weed Detection Algorithms in Rice Fields Based on Improved YOLOv10n. *Agriculture* **2024**, *14*, 2066. [CrossRef]
22. Åstrand, B.; Baerveldt, A.J. An agricultural mobile robot with vision-based perception for mechanical weed control. *Auton. Robot.* **2002**, *13*, 21–35. [CrossRef]
23. Bawden, O.; Kulk, J.; Russell, R.; McCool, C.; English, A.; Dayoub, F.; Perez, T. Robot for weed species plant-specific management. *J. Field Robot.* **2017**, *34*, 1179–1199. [CrossRef]
24. Deepfield Robotics-Bosch. Available online: https://spectrum.ieee.org/bosch-deepfield-robotics-weed-control (accessed on 21 December 2024).
25. Ecorobitix-ARA. Available online: https://ecorobotix.com/en/ara/ (accessed on 21 December 2024).
26. Ecorobitix-AVO. Available online: https://ecorobotix.com/en/avo/ (accessed on 21 December 2024).
27. Carbon Robotics. Available online: https://carbonrobotics.com/ (accessed on 21 December 2024).
28. Malu, S.K.; Majumdar, J. Kinematics, localization and control of differential drive mobile robot. *Glob. J. Res. Eng.* **2014**, *14*, 1–9.
29. De Giorgi, C.; De Palma, D.; Parlangeli, G. Online Odometry Calibration for Differential Drive Mobile Robots in Low Traction Conditions with Slippage. *Robotics* **2023**, *13*, 7. [CrossRef]
30. Li, X.; Hu, Y.; Jie, Y.; Zhao, C.; Zhang, Z. Dual-Frequency Lidar for Compressed Sensing 3D Imaging Based on All-Phase Fast Fourier Transform. *J. Opt. Photonics Res.* **2024**, *1*, 74–81. [CrossRef]
31. Hu, K.; Chen, Z.; Kang, H.; Tang, Y. 3D vision technologies for a self-developed structural external crack damage recognition robot. *Autom. Constr.* **2024**, *159*, 105262. [CrossRef]
32. Santos, J.M.; Portugal, D.; Rocha, R.P. An evaluation of 2D SLAM techniques available in robot operating system. In Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linköping, Sweden, 21–26 October 2013; pp. 1–6. [CrossRef]
33. Adanur, F.; Kabataş, M.; Benli, E. Mapping and Navigation on Rough Surface with LIDAR and IMU. In Proceedings of the 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, Turkey, 7–9 September 2022; pp. 1–5. [CrossRef]
34. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
35. Dang, F.; Chen, D.; Lu, Y.; Li, Z. YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. *Comput. Electron. Agric.* **2023**, *205*, 107655. [CrossRef]

36. Ying, B.; Xu, Y.; Zhang, S.; Shi, Y.; Liu, L. Weed Detection in Images of Carrot Fields Based on Improved YOLO V4. *Trait. Du Signal* **2021**, *38*, 341–348. [CrossRef]
37. Sportelli, M.; Apolo-Apolo, O.E.; Fontanelli, M.; Frasconi, C.; Raffaelli, M.; Peruzzi, A.; Perez-Ruiz, M. Evaluation of YOLO Object Detectors for Weed Detection in Different Turfgrass Scenarios. *Appl. Sci.* **2023**, *13*, 8502. [CrossRef]
38. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271. [CrossRef]
39. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
40. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
41. Jocher, G. YOLOv5 by Ultralytics. Available online: https://github.com/ultralytics/yolov5 (accessed on 22 September 2023).
42. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. Yolov6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
43. 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. *arXiv* **2022**, arXiv:2207.02696.
44. YOLOv8 by Ultralytics. Available online: https://docs.ultralytics.com/ (accessed on 20 December 2023).
45. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
46. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
47. YOLOv11 by Ultralytics. Available online: https://docs.ultralytics.com/tr/models/yolo11/ (accessed on 1 December 2024).
48. Tao, J.; Wang, H.; Zhang, X.; Li, X.; Yang, H. An object detection system based on YOLO in traffic scene. In Proceedings of the 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 21–22 October 2017; pp. 315–319. [CrossRef]
49. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
50. Wang, Y.; Mariano, V.Y. A Multi Object Tracking Framework Based on YOLOv8s and Bytetrack Algorithm. *IEEE Access* **2024**, *12*, 120711–120719. [CrossRef]
51. You, L.; Chen, Y.; Xiao, C.; Sun, C.; Li, R. Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack. *Electronics* **2024**, *13*, 3033. [CrossRef]
52. Abouelyazid, M. Comparative Evaluation of SORT, DeepSORT, and ByteTrack for Multiple Object Tracking in Highway Videos. *Int. J. Sustain. Infrastruct. Cities Soc.* **2023**, *8*, 42–52.
53. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 1–21. [CrossRef]
54. Yang, X.; Bist, R.; Paneru, B.; Chai, L. Monitoring activity index and behaviors of cage-free hens with advanced deep learning technologies. *Poult. Sci.* **2024**, *103*, 104193. [CrossRef]
55. Aadi, F.Z.A.H.; Sadiq, A.; Labd, Z. Comparing Object Tracking Algorithms for Real-Time Applications: Performance Analysis and Implementation Study. In Proceedings of the 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), Istanbul, Turkiye, 26–28 October 2023; pp. 1–5. [CrossRef]
56. Chen, J.; Wang, H.; Zhang, H.; Luo, T.; Wei, D.; Long, T.; Wang, Z. Weed detection in sesame fields using a YOLO model with an enhanced attention mechanism and feature fusion. *Comput. Electron. Agric.* **2022**, *202*, 107412. [CrossRef]
57. Roboflow Weeds Dataset. Available online: https://universe.roboflow.com/roboflow-demo-projects/agriculture-328r8/dataset/6 (accessed on 2 May 2024).
58. Saleem, M.H.; Potgieter, J.; Arif, K.M. Weed detection by faster RCNN model: An enhanced anchor box approach. *Agronomy* **2022**, *12*, 1580. [CrossRef]
59. Peng, H.; Li, Z.; Zhou, Z.; Shao, Y. Weed detection in paddy field using an improved RetinaNet network. *Comput. Electron. Agric.* **2022**, *199*, 107179. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.