Paradigm
reference-global.com
DOI: 10.2478/agriceng-2026-0001

ISSN 2083-1587; e-ISSN 2449-5999
2026,Vol. 30,No.1, pp. 1-17
Agricultural Engineering
https://agriceng.org

DEVELOPMENT AND EVALUATION OF A YOLO ALGORITHM-BASED ROBOTIC SPRAYER FOR REAL-TIME WEED DETECTION

Ameer H. Al-Ahmadi[a*], Alaa Subr[a], Stanisław Parafiniuk[b], Marek Milanowski[c]

[a] Department of Agricultural Machines and Equipment, College of Agricultural Engineering Sciences, University of Baghdad, Baghdad 10071, Iraq; e-mail: ameer.alahmadi@coagri.uobaghdad.edu.iq, ORCID 0000-0002-9046-9990; e-mail: alaa.kamel@coagri.uobaghdad.edu.iq, ORCID 0000-0001-5529-9773

[b] Department of Machinery Exploitation and Management of Production Processes, University of Life Sciences in Lublin, Głęboka 28, 20-612 Lublin, Poland; e-mail: stanislaw.parafiniuk@up.edu.pl, ORCID 0000-0002-8566-6527

[c] Department of Food Process Engineering, University of Life Sciences in Lublin, Głęboka 31, 20-612 Lublin, Poland; e-mail: marek.milanowski@up.edu.pl, ORCID 0000-0003-3367-7942

Corresponding author: e-mail: ameer.alahmadi@coagri.uobaghdad.edu.iq

ARTICLE INFO

Article history:
Received: August 2025
Received in the revised form: November 2025
Accepted: December 2025

Keywords:
YOLO,
Robotic Sprayer,
Weed Detection,
Artificial Intelligence,
Machine Learning

ABSTRACT

Weed control with chemicals is a challenging process that should be performed in a rational way to reduce their negative impact on the surrounding environment. The growth of artificial intelligence algorithms encourages researchers to develop smart spraying robots that detect and spray weeds and distinguish them from the main crop which leads to sustainable use of these chemicals and achieves some of the sustainable development goals. However, few studies are available to comprehensively compare different versions of YOLO algorithm to detect weed. In this research, seven versions of YOLO algorithms were evaluated for their performance to detect and spray four types of weeds, namely, Cultivated licorice (*Glycyrrhiza glabra* L.), Dyer's Croton (*Chrozophora verbascifolia*), Lambsquarters (*Chenopodium album* L.), and Puncturevine (*Tribulus terrestris* L.) using a locally manufactured remotely controlled spraying robot. The results showed that YOLOv6n surpassed other algorithms which achieved the highest precision (0.89), recall (0.80), F1-score (0.84), mAp@0.50 (0.86), inference speed (18.83 fps), in addition to the field indicators including true positive rate (0.83), false negative rate (0.17), false positive rate (0.19), true negative rate (0.81).

Introduction

Weeds are a serious threat tothe farm productivity since they compete with the main crop for nutrients, water, and light (Al-Khafaji et al., 2023; Al-ziady et al., 2021). There are many different species of weeds which vary in size, shape and colour. Moreover, they grow at different times even within the same species, which results in their distinct sizes. There are separate procedures to control weeds, but the most popular is chemical control using sprayers. The conventional spray method consists in broadcast spraying of herbicides which is usually performed evenly over the entire field regardless of weeds distribution in the field. This procedure leads to herbicide over-application that in turn causes undesired problems including increasing costs, pollution, health issues, and weed resistance to herbicides. Therefore, it is necessary to address these problems to perform the procedure reasonably as needed in the field.

Site-specific weed management, especially smart spraying of herbicides, is a promising technology that has been investigated recently by many researchers, especially under the development of processors, microcontrollers, and machine learning algorithms. Smart spraying of agrochemicals can achieve the first, second, third, sixth, and twelfth United Nations' Sustainable Development goals including combating poverty and hunger, enhancing health, and water cleanness, and ensuring sustainable consumption of agrochemicals, respectively (Nasir et al., 2023). Spot spraying procedures require nozzles provided with solenoid valves. A droplet size also needs to be controlled using different methods (adding adjuvants for example) to not affect the application process (Milanowski et al., 2022). Robotic sprayers have become popular in recent years; they are usually accompanied by weed detection technology. Different studies were conducted to evaluate the performance of these robotic sprayers (Parafiniuk et al., 2024).

Nasir et al., (2023) concluded that the selective and variable application rate methods reduced the applied agrochemicals amount to 60% (compared with conventional broadcast spraying). Leise et al., (2025) compared spot spray with broadcast spray application. They indicated a saving of herbicide ranging from 1% to 94%. These results were influenced by the weed infestation during herbicide applications (dated 2022 and 2023). In their study on sugarcane farms, Azghadi et al. (2025) showed that spot spraying was 97% more effective compared with broadcast spraying. They also found that the applied herbicide reduced by 35% when performing spot spraying (depending on the weed density). This reduction increased to 65% when the weed density was lower in some specific trial plots. He et al., (2024) found that using the variable rate spraying method resulted in a saving of 46.3% of pesticide consumption compared with constant spraying. Moreover, they found that the coverage percentage was reduced (by 13.98%) when spraying areas with scattered weed density compared with areas with dense weeds.

Weed detection is a prerequisite step that is required for spot spraying and site-specific methods. However, weed detection experiences many challenges such as the similarity in shape and color between weeds and crops, and the complexity when the weed density is high. Moreover, the size of weeds, especially when they are small, constitutes a challenge with regard to to their detection.

Many studies have been conducted to compare and evaluate diverse types and versions of weed detection algorithms in various conditions. Rahman et al., (2022) compared different weed detection models including YOLOv5, RetinaNet, EfficientDet, and Faster RCNN. They found that RetinaNet (R101-FPN) registered the highest overall detection accuracy (mean average precision (mAP@0.50) of 79.98%). While YOLOv5n model registered promising results concerning a real-time deployment in resource-constraint devices. Nasir et al. (2023) compared six YOLO-based models for the detection of tobacco crops. Their results showed that YOLOv5n model was more convincing than other versions, giving a result of 87% and 67 for F1-score and FPS rate, respectively. Kanade et al. (2025) evaluated two versions of YOLO weed detectors (YOLOv5 and YOLOv8) in their study on a cotton field. They found that the detection accuracy (mAP@0.50) range was from 69.88 % to 76.50 % for YOLOv8n and YOLOv5s6, respectively. While YOLOv5n was the fastest model with 3.07 ms inference time and had a lower number of model parameters (1.7 million) and GFLOPs (4.1) which is suitable for real-field conditions. Gómez et al. (2025) evaluated three models for deep learning in their study of weed detection in maize fields, the models were Faster R-CNN, RT-DETR, and YOLOv11. Their results indicated that YOLOv11 had an outstanding performance compared with other models, with the value of mean average precision (mAP) of 97.5% and operating in real-time at 34 frames per second (FPS). The hardware assessments showed that YOLOv11m was the most viable for field work, it achieved high precision (mAP of 94.4%) and lower energy consumption.

Zhang et al. (2022) used a deep learning-based method to detect grape downy mildew; it consisted of coordinate attention (CA) mechanism integrated into YOLOv5. The proposed approach (YOLOv5-CA) resulted in a detection precision of 85.59%, a recall of 83.70%, and a mAP@0.50 of 89.55%. Wang et al. (2023) found that using an improved yolov5s model resulted in both a real-time performance and accuracy compared to other models (yolov5l, yolov5m, and yolov5x), making it simpler to implement the model on edge devices. The recognition accuracy of target (weeds) using this model was 83%. Sulzbach et al. (2025) proposed and evaluated four new optimized modifications of YOLOv8 (YOLOv8 (femto), YOLOv8 (atto), YOLOv8 (pico), and YOLOv8 (zepto)) for weed detection in soybeans. The results showed that YOLOv8 models (especially YOLOVv8 nano and femto) registered a significant and encouraging potential for real-time weed detection and distinguishing between various weed species in soybean fields. They concluded that the mAP@0.50 values increased from 0.70 to 0.73 when using a semi-supervised technique. Moreover, they indicated that the YOLOv8 model accuracy was not affected by reduction of the model parameters, allowing the decrease of the complexity and processing time. Chen et al. (2025) proposed an improved version of the YOLOv8 baseline model namely, GE-YOLO model to detect three types of weeds that frequently appear in rice fields. Their results indicated that the proposed model (GE-YOLO) registered better performance and detection capability (93.1% mAP, 90.3% F1 Score, and 85.9 FPS) than other object detection algorithms such as YOLOv8, YOLOv10, and YOLOv11. Deng et al. (2025) have proposed the model (HAD-YOLO) for weed detection, the aim of which is to enhance the extracted features and to decrease the model complexity. They tested the model under greenhouse conditions, and the result of the mAP was 94.2%; while the test results under field conditions were 96.2% for the mAP, and 30.6 for the detection FPS. X. Wang et al. (2024) developed an improved YOLO v5s model for weed detection in straw-mulched maize field. Their findings showed that although the proposed model registered a 0.9% accuracy reduction in average weed detection, it enhanced the detection speed by 8.46%, and the memory and computational load of the model decreased by 50.36% and 53.16%, respectively. Their findings in the spraying test in field using the proposed model were 90% of weed detection accuracy, and the target localization error was within 4%, the effective application rate was 96.3%, a missed application rate of 13.3%, and an erroneous application rate of 3.7%. Vijayakumar et al. (2025) developed and evaluated a machine vision robotic smart sprayer to identify and control weeds in pepper and tomato crops. Their results indicated effective real-time weed targeting where the precision values were 89% and 88% (with missed targets below 5%), while recall values were 86% and 84% for pepper and tomato crops, respectively.

Different studies were conducted in Iraq to evaluate the pesticide application techniques and to deal with the issues that accompanied it (Jawad & Subr, 2024; A. Subr et al., 2020; A. K. Subr et al., 2019). However, less research was conducted to study the effectiveness of using robotic sprayers and spot spraying in weed control in addition to the comparison among different versions of YOLO object detection algorithm. The aim of this research is to compare the performance of seven versions of YOLO convolutional neural network algorithms in relation to some training, validation, and field indicators using a locally manufactured and remotely controlled spraying robot.

Materials and methods

The robot design

A four-wheel drive robot (Figure 1) was designed and manufactured. The robot composed of chases made of square galvanized steel that was designed to carry all other parts. Four motors were used to move the robot. Each motor was connected to 24 VDC power source (battery collection) through high power motor driver (BTS7960 IBT2) to control the speed and direction of the motor movement. Each motor was coupled with an agricultural air-filled wheel.

A 100-liter tank was used to store the liquid to be sprayed which had an upper filling hole, and two side holes: one at the upper part of the tank which was connected to an adapter for the return line, and the other one at the lower part of the tank that was connected with ball valve as a discharge port.

Two positive-displacement diaphragm pumps were connected in parallel to duplicate the discharge. Three normally closed solenoid valves were used to control the flow of the liquid, and each of them was attached to a spraying nozzle (TeeJet Stainless steel Disc-core full cone D1) with a flow rate of 0.49 $\mathrm{l\,min^{-1}}$ at 2.0 bar. Flexible reinforced plastic hoses that are designed for high pressure associated with proper adapters were adopted to connect the parts of the spraying system.

The control unit was divided into two parts: one to control robot steering, and the other to control the smart spraying system. The first part consisted of an Arduino Mega 2560 R3 board that controls the four motor drivers via a Microzone MC6C remote controller. The other part consisted of a Raspberry Pi 4 model B (8GB). Since the Raspberry Pi showed a relatively limited computational capability which is not suitable for real-time applications, especially with YOLO algorithms, which was observed from preliminary experiments, it was attached to an edge TPU coprocessor (Coral USB Accelerator) to increase the speed of the inference for the object detection algorithm running on. A webcam with (52°) field of view was connected to the Raspberry Pi 4 board. The camera was fixed at the height of 50 cm above the ground to cover the operational width of the robot (65 cm).

Figure 1. The manufactured robot that was used in the study

Two 12V batteries were connected in series to double the voltage to obtain the maximum power for the 24V motors and provide the other components with power by using DC-DC step down power converter LM2596S to lower the voltage to 5V to provide the Arduino and Raspberry Pi boards.

Image acquisition and dataset creation

The dataset is composed of 2063 images as JPG files with their annotation files as text files for two eggplants at two growth stages, namely, one week and three weeks after germination, respectively, and four weed species (Figure 2), namely, Cultivated licorice (*Glycyrrhiza glabra* L.), Dyer's Croton (*Chrozophora verbascifolia*), Lambsquarters (*Chenopodium album* L.), and Puncturevine (*Tribulus terrestris* L.) resulting in six classes.

The images were taken in different posture and illumination conditions using the mobile camera (8.9 MP) that was set in a square mode (2992×2992 pixels). The Roboflow platform was used to annotate these images by assigning a bounding box and name for each class of interest (Table 1). After the annotation processes, the images were resized into 416×416 pixels, and some augmentation techniques that are available in the platform were adopted resulting in 5305 images. The applied augmentation techniques included flipping (horizontally and vertically), 90° rotation (clockwise, counterclockwise, upside down), rotation (-15°, 15°), saturation (-30%, 30%), brightness (-15%, 15%), exposure (-10%, 10%), and noise (up to 1.5%). The dataset was divided into 92% as training set (4863 images) and 8% as validation set (442 images).

Table 1.
Number of annotations for each class before the augmentation

Class	Number of annotations
Eggplant 1	704
Eggplant 2	679
Weed 1	668
Weed 2	520
Weed 3	443
Weed 4	530

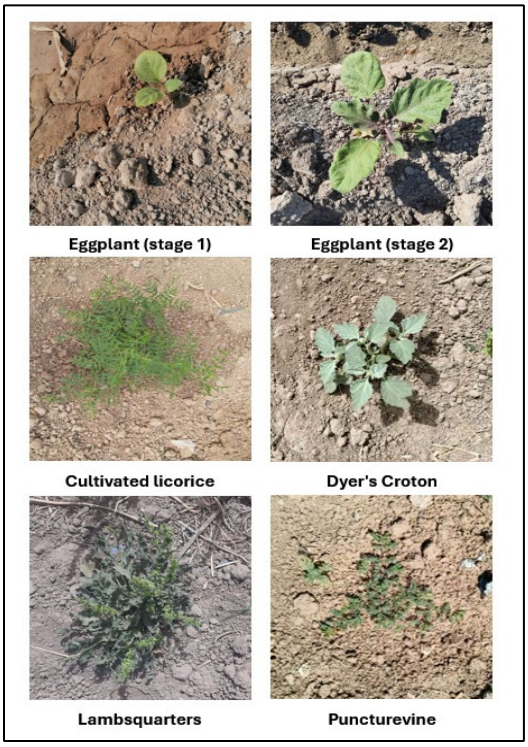


Figure 2. Eggplant at two growth stages and the types of the studied weeds

Object detection algorithm

In this research, different versions of pretrained Ultralytics YOLO algorithm were studied and compared for their performance in detection six classes including two growth stages of eggplant and four types of weeds. YOLO algorithms are convolutional neural networks that were built as a robust single-shot object detection algorithms that can be used for machine vision applications (Nasir et al., 2023). The versions that were studied in this research included YOLOv5n, YOLOv6n, YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n, and YOLOv12n since they are trained using the same procedure and dataset style. Each algorithm was trained for 1000 epochs and a batch size of 8 with a patience of 100 that the training was forced to early stop after 100 epoch if there was no improvement in the performance of the algorithm (Table 2). It should be noted that there is a tradeoff in relation to the image input size that increasing the image input size increases the performance metrics but decreases the inference speed (Zhang et al., 2022). Therefore, the image size of 416×416 pixels was adopted in this study to balance the performance and inference speed of the detection.

The training process was performed using a Lenovo laptop with an 11th Gen Intel(R) Core (TM) i7-11370H@3.30GHz processor, 16-GB RAM, and NVIDIA GeForce RTX 3050 Laptop GPU (4096MiB). The used operating system was Windows 11 (64-bit). Python-3.9.2 with Torch-2.1.1, and CUDA 11.8 was used for the training process.

The algorithms were supposed to be used with the robot smart spraying system that run on Raspberry Pi 4 Model B associated with the Coral USB accelerator. Therefore, the models obtained from the training process were converted from Pytorch format to TensorFlow Lite Edge TPU format which is compatible with Coral USB accelerator and provide faster inferecne speed except YOLOv10 that its converting to the TensorFlow Lite Edge TPU format was failed for unknown reasons.

Table 2.
Training settings and hyperparameters

Parameter	Value
Epochs	1000
Batch size	8
Image size	416
Patience (Early stopping)	100
Optimizer	auto
Learning rate	0.01
Momentum	0.937

A python code was created to receive the video frames from a web camera, process and analyze them using one of the studied algorithms, then to save a video file (mp4 format) for each replicate to be studied and analyzed in order to get the results of the type-specific weed spray rate. The code was designed to display a graphical user interface (GUI) that can be displayed on the smartphone screen via RealVNC Veiwer software to enable the user to name the process, and select the algorithm tested easily. Moreover, the GUI enables the user to select the mode among variable rate mode, constant rate mode, open and close each valve separately for the purpose of checking or calibrating the spraying flow rate.

Evaluation metrics

Training and validation metrics for YOLO algorithms

The performance of the studied algorithms was compared based on the mean average precision at an intersection over union (IoU) threshold of 0.50 (mAP@0.50) ((1), the precision ((2) which represents the ratio of true positives relative to all positive predictions and measures the capability of the model to avoid misclassification of objects under wrong classes (false positives), the recall ((3) which represents the ratio of true positives relative to all actual positives and measure the capability of the model to detect as many instances of the class as possible with less missed predictions, and the F1 score ((4) which represents the harmonic mean of precision and recall, providing a balanced assessment of the model's performance while considering both false positives and false negatives (Ajayi et al., 2023; Ghadi & Salman, 2022; Zhang et al., 2022). Additionally, the training duration and the inference speed was also investigated as additional performance indicators.

$$mAP = \frac{\sum_{m=1}^{M} AP(m)}{M} \tag{1}$$

where:
AP – is the average precision,
M – is the number of classes,
m – the specific class number.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

where:
TP – is true positive,
FP – is false positive,
FN – is false negative.

Inference speed (FPS)

The inference speed as frame per second (FPS) for each model after converting them to TensorFlow Lite Edge TPU format was investigated by running each one for five minutes and recording the perpocees (T_{pre}), inference (T_i), and postprocess (T_{post}) times for each inference (Al-Hababi et al., 2024), then the inference speed was calculated by (5:

$$FPS = \frac{1000}{T_{pre}\,(ms) + T_i\,(ms) + T_{post}\,(ms)} \tag{5}$$

The preprocess time includes the time required for resizing the input image, normalization of the pixel values, preparing the image details to be compatible for YOLO model such as the ordering of the channels and layouts. While the inference time represents the time required for the model to generate the predictions of the input image. Finally, the postprocess time represents the time for refining the raw predictions obtained from the inference process excluding the low confidence predictions by what is called non-maximum suppression (NMS) to provide the final prediction (Portnov et al., 2024; Yang et al., 2025).

Field experiment

This spraying robot was designed to spray one row in one pass including intra-row weeds that are detected within the crop row, and inter-row weeds that are detected on the right and left sides of the crop row. Therefore, the image analyzed by the YOLO algorithm was divided into three identical columns, each covering one of the rows and relating to one of the solenoid valves. Moreover, a horizontal threshold line divided the image into two unidentical rows which can be adjusted to prevent pre- or post-spraying (Figure 3). Each solenoid valve will be activated when a detected weed localizes in its related column of the camera and passes the horizontal threshold line.

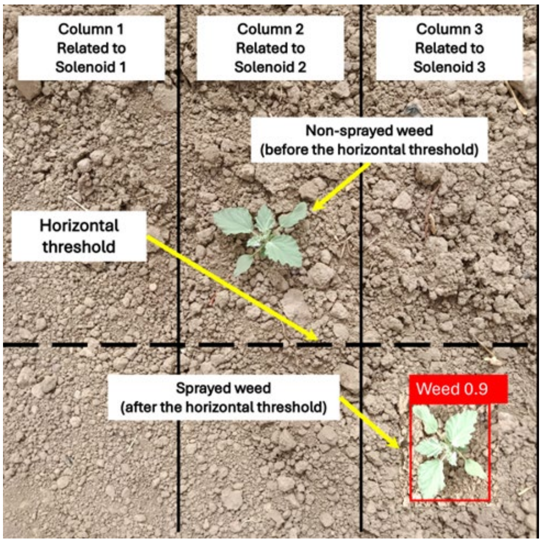


Figure 3. The layout of the image fed to the algorithm

A preliminary experiment was conducted to determine the best robot forward speed from 1 to 3 km.h^{-1}. It was shown that the speed of 2 km·h^{-1} was the best choice since 3 km.h^{-1} was not suitable for the inference speed available from the detection system. A field experiment was conducted with three replicates to evaluate the robot's performance in field conditions. Each replicate contained 10 plants from each type of the four studied weed types arranged randomly in three lines according to the three nozzles of the robot. White inkjet photo paper was cut in 7.5×2.5 cm to be used as a sample collector. A food dye namely, Brilliant blue sivi karisimi was mixed with tap water – 5g dye to 100 ml water – as a spraying solution (Oun & Subr, 2023). The pieces of cut paper were put next to each weed and some of them were randomly distributed on the naked ground to facilitate counting of the sprayed weed and the weed - free patches of ground that that were mistakenly sprayed.

Four parameters were calculated – using the (6, (7, (8, (9 namely, true positive rate (sprayed weed rate), false negative rate (missed weed rate), false positive rate (non-weed sprayed rate), true negative rate (non-weed avoidance rate) for each of the four algorithms tested.

Moreover, the results for the type-specific weed spray rate were calculated by manually reviewing the video recorded by the python code for each replicate and counting the total number of each weed type and the number of each weed type that were correctly sprayed, and the type-specific weed spray rate for each type was calculated according to (10.

$$True\ positive\ rate = \frac{Weed\ sprayed}{Total\ number\ of\ weed} \tag{6}$$

$$False\ negative\ rate = 1 - True\ positive\ rate \tag{7}$$

$$False\ positive\ rate = \frac{Non{-}weed\ sprayed}{Total\ number\ of\ Non{-}weed} \tag{8}$$

$$True\ negative\ rate = 1 - False\ positive\ rate \tag{9}$$

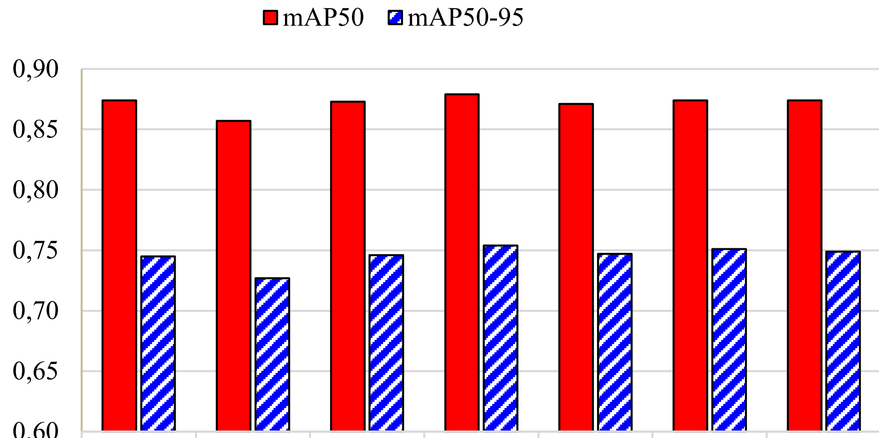
$$TSWSR_i = \frac{N_i}{T_i} \tag{10}$$

where:
$TSWSR$ – is the type-specific weed spray rate,
N_i – is the number of weeds of type (i),
T_i – is the total number of weeds type (i).

Results and discussion

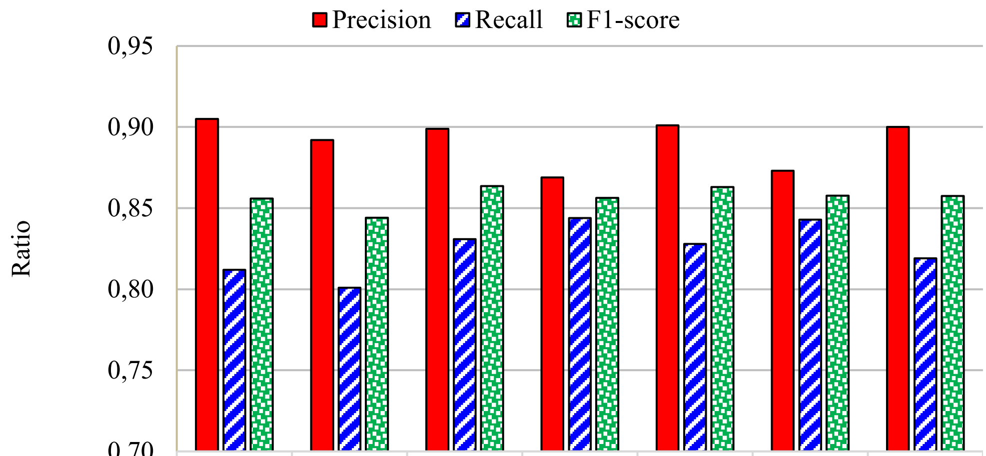
Training and validation metrics

The mean average precision obtained after the training process revealed that all the algorithms achieved approximately similar mAP@0.50 ranged from 0.86 for YOLOv6n to 0.88 for YOLOv9t (Figure 4). The training results showed that all the trained algorithms achieved similar precision and recall values. The precision of the algorithms ranged from 0.87 for YOLOv11n to 0.91 for YOLOv5n. However, the recalls were less than the precisions for all the algorithms that ranged from 0.80 for YOLOv6n to 0.84 for YOLOv11n. The F1-score values were identical for all the algorithms (0.86) except for YOLOv6n which was slightly lower (0.84) (Figure 5).



	YOLOv05n	YOLOv06n	YOLOv08n	YOLOv09t	YOLOv10n	YOLOv11n	YOLOv12n
mAP50	0,87	0,86	0,87	0,88	0,87	0,87	0,87
mAP50-95	0,75	0,73	0,75	0,75	0,75	0,75	0,75

Figure 4. Mean average precision of each algorithm at intersection over union (IoU) threshold of 0.5 (mAP50) and the range of threshold from 0.5-0.95 (mAP50-95)



	YOLOv05n	YOLOv06n	YOLOv08n	YOLOv09t	YOLOv10n	YOLOv11n	YOLOv12n
Precision	0,91	0,89	0,90	0,87	0,90	0,87	0,90
Recall	0,81	0,80	0,83	0,84	0,83	0,84	0,82
F1-score	0,86	0,84	0,86	0,86	0,86	0,86	0,86

Figure 5. The precision, recall, and F1-score of each algorithm

Inference speed

The results of the inference speed showed a significant variation among the algorithms studied. The highest average inference speed was 18.83 FPS that was obtained by YOLOv6n, while the lowest was 2.94 FPS that was obtained by YOLOv12n. Figure 6 showed that YOLOv6n, YOLOv5n, and YOLOv8n showed promising inference speeds for real-time applications and vice versa for YOLOvasYOLOv11n, and YOLOv9t. YOLOv10n was excluded from this test since it cannot be converted into TensorFlow Lite Edge TPU format as it was mentioned above.

Therefore, regarding the result of the inference time and the similarities among the studied algorithms in relation to other performance indicators including the precision, recall, F1-score, and mean average precision, YOLOv6n was adopted for our field experiment.

Training time

The time of training the algorithm can be considered as a comparative indicator when adopting an appropriate algorithm especially when other indicators are similar. The training time of these algorithms ranges from 3.75 h for YOLOv8n to 11.45 for YOLOv9t. The results showed that YOLOv5n, YOLOv8n can be the best option regarding the training time, and they both achieved relatively similar mAP, precision, recall, and F1-score to other algorithms (Figure 7).

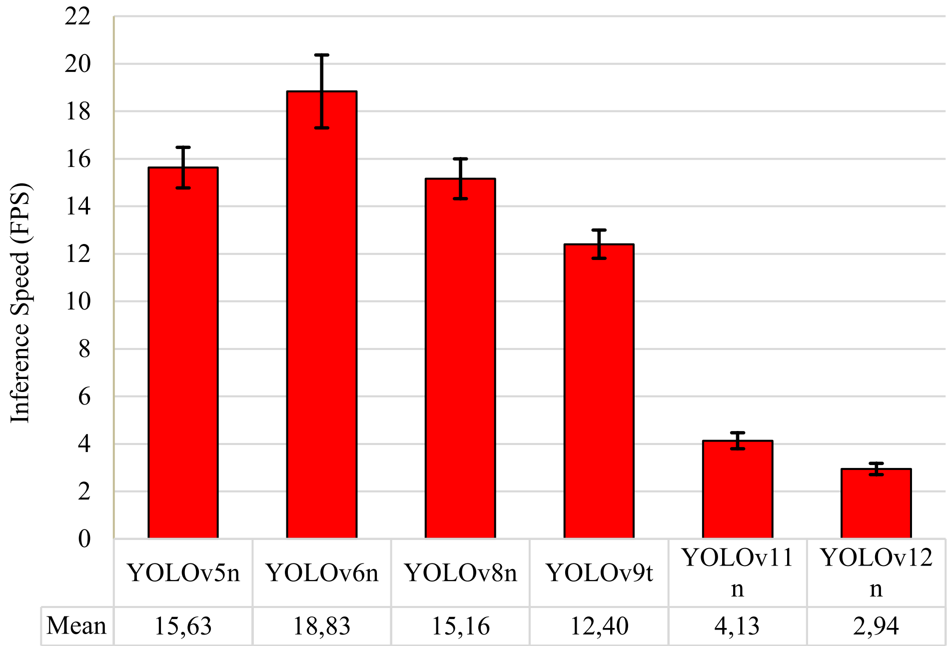
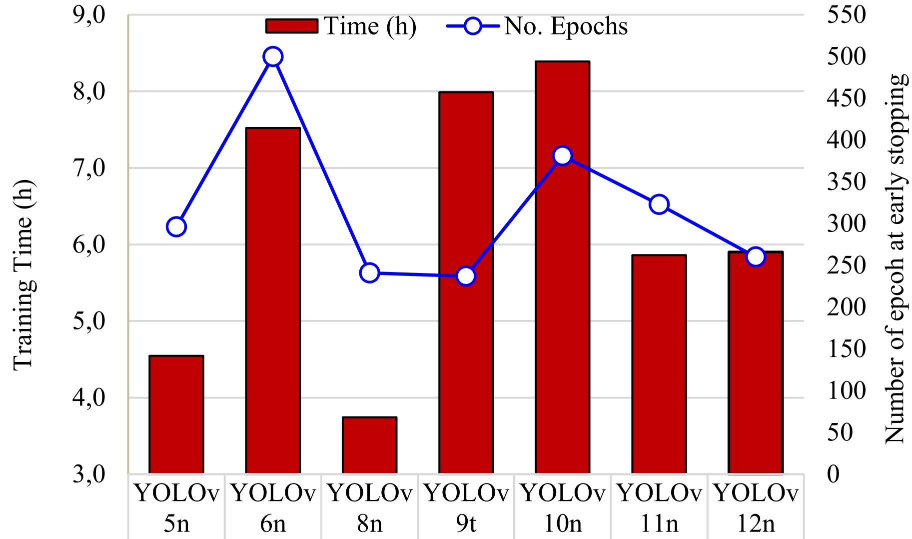


Figure 6. Mean and the standard deviation of inference Speed (FPS) for each algorithm)



	YOLOv5n	YOLOv6n	YOLOv8n	YOLOv9t	YOLOv10n	YOLOv11n	YOLOv12n
Time (h)	4,55	7,52	3,75	7,99	8,39	5,86	5,90
No. Epochs	296	500	241	237	381	323	260

Figure 7. Training time and number of epochs of each algorithm

Field parameters

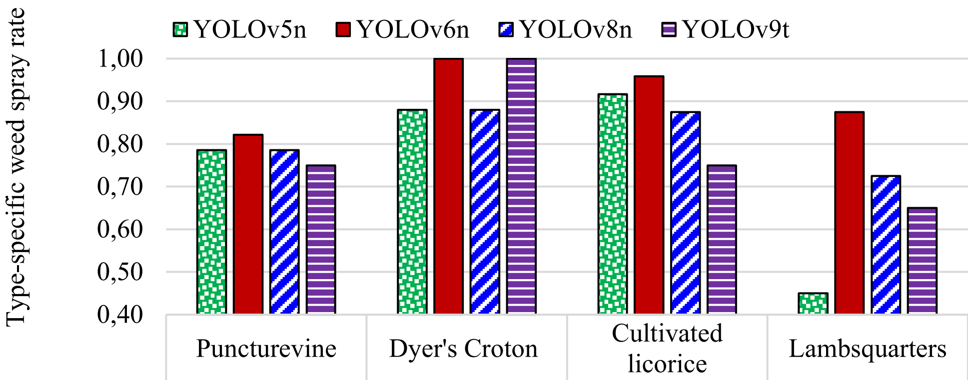
The performance of YOLOv5n, YOLOv6n, YOLOv8n, and YOLO9t were tested in the field. However, YOLOv11 and YOLOv12 were excluded from the field investigation because of their low inference speed with the Raspberry Pi 4 which did not qualify them for real time detection as we mentioned above. As illustrated in

Table 3, in relation to the true positive rate, YOLOv6n significantly surpassed other algorithms and then achieved the lowest false negative rate. In relation to the false positive rate, YOLOv5n and YOLOv6n achieved the lowest values, while YOLOv8n and YOLOv9t achieved the highest values. in relation to the true negative rate, YOLOv5n and YOLOv6n achieved the highest values, while YOLOv8n and YOLOv9t achieved the lowest ones.

The results of the type-specific weed spray rate (Figure 8) illustrated that YOLOv6n generally surpassed other algorithms. However, Puncturevine and lambsquarters showed the lowest spray rate, especially for lambsquarters. These results were expected because of the lowest recall value obtained from the validation process following the training process of each algorithm (the data are not shown in this article). The aforementioned results may be increased when using upper versions other than nano version (n) along with higher computational capability processors. This study revealed that the characteristics of the object detected and its differentiation from the background and from other objects play a key role in an algorithm detection capability for that specific object.

Table 3.
Means and standard deviations of field parameters accompanied by the Tukey's test differences as superscript letters.

Parameter	Model	Mean	Standard deviation
True positive rate	YOLOv5n	0.36 ^{b}	0.127
	YOLOv6n	**0.83** ^{a}	0.083
	YOLOv8n	0.50 ^{b}	0.144
	YOLOv9t	0.44 ^{b}	0.096
False negative rate	YOLOv5n	0.64 ^{a}	0.127
	YOLOv6n	**0.17** ^{b}	0.083
	YOLOv8n	0.50 ^{a}	0.144
	YOLOv9t	0.56 ^{a}	0.096
False positive rate	YOLOv5n	**0.14** ^{b}	0.127
	YOLOv6n	**0.19** ^{b}	0.173
	YOLOv8n	0.42 ^{ab}	0.083
	YOLOv9t	0.56 ^{a}	0.096
True negative rate	YOLOv5n	**0.86** ^{a}	0.127
	YOLOv6n	**0.81** ^{a}	0.173
	YOLOv8n	0.58 ^{ab}	0.083
	YOLOv9t	0.44 ^{b}	0.096



	Puncturevine	Dyer's Croton	Cultivated licorice	Lambsquarters
YOLOv5n	0,79	0,88	0,92	0,45
YOLOv6n	0,82	1	0,96	0,88
YOLOv8n	0,79	0,88	0,88	0,73
YOLOv9t	0,75	1	0,75	0,65

Figure 8. Type-specific weed spray rate of each algorithm

Conclusions

The study was conducted to evaluate the performance of seven versions of YOLO algorithm in relation to some training, validation, and field indicators for detecting and spraying four types of weeds. A locally manufactured remotely controlled robot provided with a Raspberry Pi 4 was used as a tested platform. The results of YOLOv6n with the locally manufactured robot were promising because it achieved the highest performance indicators with the highest inference speed of (18.83 fps) and could detect and spray the highest type-specific weed spray rate of the studied weeds ranged from 0.82 for Puncturevine to 1 for Dyer's Croton. However, these results obtained from YOLOv6n, or the other studied versions can be increased using the upper model of each algorithm such as the small (s), or medium (m) with a higher computational capability processor rather than that of Raspberry Pi to obtain a reasonable inference speed.

Author Contributions: Conceptualization, A.S. and S.P.; methodology, A.S. and S.P.; validation, M.M.; investigation, M.M.; data curation, A.Al-A.; writing-original draft preparation, A.S. and S.P.; writing-review and editing, A.Al-A., A.S. and S.P.; visualization, A.Al-A.; supervision, A.S. and S.P.

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

Funding: This research received no external funding.

References

Ajayi, O. G., Ashi, J., & Guda, B. (2023). Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images. *Smart Agricultural Technology*, *5*, 100231. https://doi.org/10.1016/j.atech.2023.100231.

Al-Hababi, M. A. M., Habib, A., Thabit, F., & Liu, Y. (2024). A Novel Pre-Processing Approach and Benchmarking Analysis for Faster, Robust, and Improved Small Object Detection Methods. *Remote Sensing*, *16*(20), 3753. https://doi.org/https://doi.org/10.3390/rs16203753.

Al-Khafaji, Mustafa. J., Safi, S. M. A., & Hammood, W. F. (2023). Effect of herbicides on growth, grain yield and quality of barley. *Iraqi Journal of Agricultural Sciences*, *54*(4), 1094–1100.

Al-ziady, S. H. A., Hassan, D. S., & Al-Mashhadani, A. S. A. (2021). Interaction of foliar nutrients and herbicides into production. *Iraqi Journal of Agricultural Sciences*, *52*(5), 1225–1237. https://doi.org/10.36103/ijas.v52i5.1460.

Azghadi, M. R., Olsen, A., Wood, J., Saleh, A., Calvert, B., Granshaw, T., Fillols, E., & Philippa, B. (2025). Precision robotic spot-spraying: Reducing herbicide use and enhancing environmental outcomes in sugarcane. *Computers and Electronics in Agriculture*, *235*, 110365. https://doi.org/https://doi.org/10.1016/j.compag.2025.110365.

Chen, Z., Chen, B., Huang, Y., & Zhou, Z. (2025). GE-YOLO for Weed Detection in Rice Paddy Fields. *Applied Sciences (2076-3417)*, *15*(5). https://doi.org/10.3390/app15052823.

Deng, L., Miao, Z., Zhao, X., Yang, S., Gao, Y., Zhai, C., & Zhao, C. (2025). HAD-YOLO: An Accurate and Effective Weed Detection Model Based on Improved YOLOV5 Network. *Agronomy*, *15*(1), 57. https://doi.org/10.3390/agronomy15010057.

Ghadi, N. M., & Salman, N. H. (2022). Deep Learning-Based Segmentation and Classification Techniques for Brain Tumor MRI: A Review. *Journal of Engineering*, *28*(12), 93–112. https://doi.org/10.31026/j.eng.2022.12.07.

Gómez, A., Moreno, H., & Andújar, D. (2025). Intelligent Inter-and Intra-Row Early Weed Detection in Commercial Maize Crops. *Plants*, *14*(6), 881. https://doi.org/https://doi.org/10.3390/plants14060881.

He, Z., Ding, L., Ji, J., Jin, X., Feng, Z., & Hao, M. (2024). Design and Experiment of Variable-Spray System Based on Deep Learning. *Applied Sciences*, *14*(8), 3330. https://doi.org/https://doi.org/10.3390/app14083330.

Jawad, B. R. A., & Subr, A. K. (2024). COMPARISON OF TWO TYPES OF SENSORS AND THEIR EFFECT ON SPRAY QUALITY PEAR TREES. *Iraqi Journal of Agricultural Sciences*, *55*(2), 836–844.

Kanade, A. K., Potdar, M. P., Kumar, A., Balol, G., & Shivashankar, K. (2025). Weed detection in cotton farming by YOLOv5 and YOLOv8 object detectors. *European Journal of Agronomy*, *168*, 127617. https://doi.org/https://doi.org/10.1016/j.eja.2025.127617.

Leise, A., Singh, M., La Menza, N. C., Knezevic, S., & Jhala, A. J. (2025). Evaluating precision sprayers for targeted weed management in corn and soybean. *Agrosystems, Geosciences & Environment*, *8*(1), e70055. https://doi.org/https://doi.org/10.1002/agg2.70055.

Milanowski, M., Subr, A., Parafiniuk, S., & Rózańska-Boczula, M. (2022). The Effect of Adjuvant Concentration on Changes of Spray Characteristics and Spraying Parameters for Selected Types of Nozzles. *Agricultural Engineering*, *26*(1). https://doi.org/10.2478/agriceng-2022-0010.

Nasir, F. E., Tufail, M., Haris, M., Iqbal, J., Khan, S., & Khan, M. T. (2023). Precision agricultural robotic sprayer with real-time Tobacco recognition and spraying system based on deep learning. *PLoS ONE*, *18*(3 MARCH). https://doi.org/10.1371/journal.pone.0283801.

Oun, D. A., & Subr, A. (2023). Effect of vertical boom position of the greenhouse sprayer on the penetration of spray to the eggplants canopy. *IOP Conference Series: Earth and Environmental Science*, *1262*(9), 092009. https://doi.org/10.1088/1755-1315/1262/9/092009.

Parafiniuk, S., Kopiński, Ł., Ogrodniczek, J., Karpiński, P., Rabier, F., & Subr, A. (2024). Quality Evaluation of the Operation of the XAG R 150 Autonomous Sprayer in Currant Cultivation. *International Symposium on Farm Machinery and Processes Management in Sustainable Agriculture*, 357–366. https://doi.org/10.1007/978-3-031-70955-5_39.

Portnov, A., Shubin, A., & Frolova, G. (2024). The effect of pre-and post-processing techniques on tree detection in young forest stands from images of snow cover using YOLO neural networks. *European Journal of Forest Engineering*, *10*(2), 149–159. https://doi.org/https://doi.org/10.33904/ejfe.1462335.

Rahman, A., Lu, Y., & Wang, H. (2022). Deep Neural Networks for Weed Detections Towards Precision Weeding. *2022 ASABE Annual International Meeting*, 1.

Subr, A., Al-Ahmadi, A., & Abbas, M. (2020). Effect of nozzle type and some locally used surfactants on the spray quality. *Iraqi Journal of Agricultural Sciences*, *51*(3), 856–864. https://doi.org/10.36103 /ijas.v51i3.1040.

Subr, A. K., Alheidary, M. H. R., & Al-Ahmadi, A. H. (2019). The informatics adequacy on the spraying technology in Iraqi agricultural researches: A literature review. *Journal of Physics: Conference Series*, *1294*(9), 092007. https://doi.org/10.1088/1742-6596/1294/9/092007.

Sulzbach, E., Scheeren, I., Veras, M. S. T., Tosin, M. C., Kroth, W. A. E., Merotto Jr, A., & Markus, C. (2025). Deep learning model optimization methods and performance evaluation of YOLOv8 for enhanced weed detection in soybeans. *Computers and Electronics in Agriculture*, *232*, 110117. https://doi.org/https://doi.org/10.1016/j.compag.2025.110117.

Vijayakumar, V., Neto, A., & Ampatzidis, Y. (2025). AI-driven autonomous spraying for precision weed management in specialty crop production. In *Precision agriculture'25* (pp. 1014–1021). Wageningen Academic.

Wang, B., Yan, Y., Lan, Y., Wang, M., & Bian, Z. (2023). Accurate Detection and Precision Spraying of Corn and Weeds Using the Improved YOLOv5 Model. *IEEE Access*, *11*, 29868–29882. https://doi.org/10.1109/ACCESS.2023.3258439.

Wang, X., Wang, Q., Qiao, Y., Zhang, X., Lu, C., & Wang, C. (2024). Precision Weed Management for Straw-Mulched Maize Field: Advanced Weed Detection and Targeted Spraying Based on Enhanced YOLO v5s. *Agriculture*, *14*(12), 2134.

Yang, D., Chen, L., Hao, X., & Zhang, Y. (2025). Object Detection Post Processing Accelerator Based on Co-Design of Hardware and Software. *Information*, *16*(1), 63. https://doi.org/https://doi.org/10.3390/info16010063

Zhang, Z., Qiao, Y., Guo, Y., & He, D. (2022). Deep Learning Based Automatic Grape Downy Mildew Detection. *Frontiers in Plant Science*, *13*, 872107. https://doi.org/10.3389/fpls.2022.872107.

ROZWÓJ I OCENA ROBOTA OPRYSKUJĄCEGO OPARTEGO NA ALGORYTMIE YOLO DO DETEKCJI CHWASTÓW W CZASIE RZECZYWISTYM

Streszczenie. Zwalczanie chwastów za pomocą środków chemicznych to trudny proces, który należy przeprowadzać w sposób racjonalny, aby ograniczyć negatywny wpływ tych środków na otaczające środowisko. Rozwój algorytmów sztucznej inteligencji zachęca naukowców do tworzenia inteligentnych robotów do opryskiwania, które wykrywają i opryskują chwasty, odróżniając je od głównych upraw. Dzięki temu możliwe jest zrównoważone stosowanie chemikaliów i osiągnięcie niektórych celów zrównoważonego rozwoju. Jednakże dostępnych jest niewiele badań pozwalających na kompleksowe porównanie różnych wersji algorytmu YOLO do wykrywania chwastów. W niniejszym badaniu oceniono siedem wersji algorytmów YOLO pod kątem ich skuteczności w wykrywaniu i opryskiwaniu czterech rodzajów chwastów, a mianowicie lukrecji gładkiej (*Glycyrrhiza glabra* L.), krotonu barwierskiego (*Chrozophora verbascifolia*), komosy białej (*Chenopodium album* L.) i buzdyganka naziemnego (*Tribulus terrestris* L.), za pomocą zdalnie sterowanego robota opryskowego lokalnej produkcji. Wyniki pokazały, że YOLOv6n przewyższył inne algorytmy, które osiągnęły najwyższą precyzję (0,89), odwołanie (0,80), wynik F1 (0,84), mAp@0,50 (0,86), szybkość wnioskowania (18,83 fps), a także wskaźniki terenowe, w tym wskaźnik wyników prawdziwie dodatnich (0,83), wskaźnik wyników fałszywie ujemnych (0,17), wskaźnik wyników fałszywie dodatnich (0,19) i wskaźnik wyników prawdziwie ujemnych (0,81).

Słowa kluczowe: YOLO, robot opryskujący, wykrywanie chwastów, sztuczna inteligencja, nauczanie maszynowe