Research Article

Lucía Sandoval-Pillajo, Luis Perugachi-Chávez, Marco Pusdá-Chulde, Adriana Giret-Boggino and Iván García-Santillán*

# Multi-class weed detection based on deep learning using unmanned aerial vehicle images in potato fields

https://doi.org/10.1515/comp-2025-0058
Received May 2, 2025; accepted April 14, 2026;
published online July 2, 2026

**Abstract:** Early weed detection is a critical task in precision agriculture, as weeds compete with crops for resources and significantly reduce yield and quality. This study proposes a multi-class weed detection approach based on deep learning using unmanned aerial vehicle (UAV) images acquired in real potato fields under uncontrolled environmental conditions. A custom dataset containing five classes (potato crop and four weed categories) was created, annotated, and made publicly available to support reproducibility and future comparisons. Two modern object detection architectures, based on convolutional neural networks and Transformer models, were evaluated through multiple training variants to analyze the impact of class balancing, image resolution, and hyperparameter optimization on detection performance. The models were assessed using standard object detection metrics and complemented with statistical validation to verify the reliability of the predictions in real-field conditions. Experimental results show that both architectures achieve robust and comparable performance, demonstrating their suitability for near real-time weed detection and precision agriculture applications. The findings confirm the effectiveness of combining advanced deep learning models with UAV imagery for accurate multi-class weed identification in complex agricultural environments.

**Keywords:** weed detection; deep learning; UAV images; object detection; CNN; transformer

***Corresponding author: Iván García-Santillán**, Universidad Técnica del Norte, Ibarra, Ecuador, E-mail: idgarcia@utn.edu.ec. https://orcid.org/0000-0001-6404-5185
**Lucía Sandoval-Pillajo**, Universitat Politècnica de València, Valencia, Spain; and Universidad Técnica del Norte, Ibarra, Ecuador, E-mail: ansanpil@doctor.upv.es. https://orcid.org/0000-0003-1463-017X
**Luis Perugachi-Chávez and Marco Pusdá-Chulde**, Universidad Técnica del Norte, Ibarra, Ecuador,
E-mail: laperugachic@utn.edu.ec (L. Perugachi-Chávez), mrpusda@utn.edu.ec (M. Pusdá-Chulde).
https://orcid.org/0000-0003-4265-999X (M. Pusdá-Chulde)
**Adriana Giret-Boggino**, Universitat Politècnica de València, Valencia, Spain, E-mail: agiret@dsic.upv.es. https://orcid.org/0000-0002-2311-0785

## 1 Introduction

### 1.1 Statement problem

According to the United Nations, by 2030, more than 600 million people will lack a healthy diet, and the demand for food will grow by 70 % by 2050 [1]. This means agricultural production must be increased to meet food needs through more environmentally sustainable methods.

Weeds are undesirable plants that develop spontaneously near crops and affect the yield of agricultural production because they directly compete for resources such as nutrients and space, generating economic losses for farmers [2]. Chemical control is a common method for combating weeds [3]. However, it can harm the crop, the environment, and health due to excessive and inappropriate use of chemicals [4].

One of the objectives of precision agriculture is to reduce the excessive use of chemicals in weed control by using tools that optimize their use and reduce their dispersion in the environment [5]. However, the task of detecting crops and weeds is very challenging due to the similarity of their physical characteristics, their intertwined growth, occlusion, and the influence of environmental factors [6].

Traditional weed control methods are time-consuming and labor-intensive, making them an unviable option for large-scale agriculture [7]. Quantifying weed cover in crop fields is crucial for optimizing control methods. This assessment helps to correctly dose herbicides and ensure effective control through precise and localized application, without harming the crop [8]. Likewise, detecting multiple specific weed types instead of a single generic class is crucial for precision agricultural management. Different weeds

Open Access. © 2026 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

require differentiated control strategies and efficient treatments, avoiding excessive herbicide application and collateral damage to soil biodiversity and health.

Deep Learning algorithms have demonstrated an exceptional ability to recognize and learn unique patterns hierarchically from data [9]. Convolutional neural networks (CNNs) have been widely used in object detection and classification [10]–[15], including weeds, due to the excellent results they have shown [4], [16]–[19]. Specifically, the YOLO (You Only Look Once) network has been widely used in computer vision tasks due to its accuracy and speed, especially the performance of its recent version, YOLOv11 [20].

Furthermore, Transformer networks have proven to be extremely efficient in solving problems with sequential data [21]. Due to this great success, this architecture has also been adapted to solve computer vision tasks [22]. Detection Transformer (DETR) was the first Transformer-based model for the task of object detection [23]. This architecture eliminated the dependency on manually designed components, such as anchor generation and NMS (non-maximum suppression) [24] at the cost of high computational costs [25]. Consequently, several DETR-based architectures, such as DN-DETR, have been presented to accelerate convergence during training [26]; DINO (distillation with no labels) improves accuracy by using a contrastive training approach to reduce noise in predictions [27], and recently, Real-Time Detection Transformer (RT-DETR) [28] to improve real-time inference speed.

Therefore, the objective of this work is to detect four types of native weeds (Kikuyu, Dandelion, Broadleaf dock, and other unidentified weeds) in potato crops using modern state-of-the-art architectures, namely RT-DETR-1 and YOLOv11, with five variations each based on hyperparameter optimization. These deep networks were recently proposed to improve real-time inference speed. A comparison of the results (accuracy and speed) in weed detection using these current architectures based on Transformer and CNN, respectively, is novel and highly useful in precision agriculture using technology and computer science in uncontrolled environments (lighting, weather, plants morphology, drone heights, camera movements, etc.), and is also vital for maintaining sustainable food production. Precise weed identification and timely action are essential for efficient weed control. Potato cultivation was selected because it is an iconic product in the northern area of Ecuador, where this research is carried out. The categories of treated weeds are the most prevalent in this region. Besides, there are very few studies for weed recognition on potato crops, considered under-researched [29].

The contribution of this work does not rely on the novelty of the architectures themselves, but on their application and evaluation under challenging real-field conditions. Specifically, this study provides: (i) a publicly available multi-class UAV dataset for weed detection in potato crops, (ii) a systematic comparison of CNN- and Transformer-based models under consistent experimental settings, (iii) an analysis of class imbalance and data augmentation strategies, and (iv) a combined evaluation framework integrating detection metrics with statistical validation.

The algorithms were developed online on the Kaggle platform with an NVIDIA TESLA P100 GPU using the framework PyTorch and the Ultralytics library, using the Knowledge Discovery in Databases (KDD) methodology for data analysis [30], which allowed for a structured and systematic process, from acquiring field images to evaluating the generated models. Some versions of these algorithms are tested on a custom image dataset, adjusting the hyperparameters until achieving an optimal model to make it more specialized and robust for weed detection. Adapting these modern network architectures to native weed detection using our custom dataset represents a significant contribution to the field of precise weed control under dynamic field conditions. Uncontrolled environments present major challenges in Deep Learning, as models must be robust, adaptable, and able to generalize to unpredictable conditions without manual tuning.

The unmanned aerial vehicle (UAV) images that make up the custom dataset for this study were captured in the provinces of Carchi and Imbabura (Ecuador) during the years 2023 and 2024 in potato fields. Initially, 3,579 random RGB images ($5{,}472 \times 3{,}648$) of crops and weeds were collected, leaving 700 after an appropriate selection. We have made this custom dataset publicly available on the Kaggle platform for academic and comparative purposes in the following link (click here). This symbolizes another important contribution due to the difficulty and time involved in collecting and annotating various native weeds in images with their respective bounding boxes and classes. We found no public dataset containing UAV RGB images with these three weeds (Kikuyo, Dandelion, Broadleaf dock) in potato crops [31].

The metrics based on Intersection over Union (IoU) are mainly used to evaluate the quality of the model segmentation (mean Average Precision, mAP), which is critical for practical applications [32]. It is complemented with the Student's *t*-test to verify global coherence in the predicted and actual number of plants (weeds/crops). Combining both metrics offers a comprehensive approach to validating the model in practical applications. To our knowledge, no work

has fully exploited the joint application of these evaluation criteria in weed detection, constituting support for the research subject.

The remainder of the manuscript is as follows: Section 1.2 presents related work and literature gaps. Section 2 indicates the methodology, dataset, and software used to develop this study for detecting different kinds of weeds. Section 3 shows the main results and statistical tests obtained in this proposal. Section 4 considers a discussion with other related research, and finally, the conclusions and future work are presented in Section 5.

## 1.2 Related work

Below, we summarize some relevant works closely related to this study that served as a theoretical basis for our proposal and subsequently for comparison. However, some of them used only ground images instead of UAV images.

Regarding works based on the Transformer architecture, Abuhani et al. [33] used the Detection Transformer DETR architecture (and others such as YOLOv8) to identify weeds in sugar beet and sunflower crops. A dataset of 1,600 sugar beet images in Germany and another of 500 sunflower images in Italy were acquired, both captured by a UGV robot (not a drone). The model was trained with the AdamW optimizer for 20 epochs, a learning rate of 1e-4, and a batch size of 2. The best DETR results were 0.353 mAP and 0.43 mAP for the beet and sunflower datasets. YOLOv8 was better at weed detection, achieving 0.813 and 0.770, respectively. UAV images were not used in this work, and a key challenge was the detection of smaller weeds. Thus, weed detection seemed to be more challenging than crop detection.

Iqbal et al. [34] explored several models to identify corn crops and weeds. Images were captured in RGB format at $640 \times 480$ and $1{,}280 \times 720$ pixels with two sensors mounted on a ground-based data acquisition platform (no drone). The dataset contains 3,574 specimens and was collected over three weeks during the corn crop growth stage. Training was performed using a 32 GB NVidia Tesla V100 GPU (Graphics Processing Unit) with CUDA (Compute Unified Device Architecture). The models used were Faster R-CNN, RetinaNet, FCOS, YOLO, and DINO, achieving 0.716, 0.68, 0.693, 0.854, and 0.758 mAP, respectively, with YOLO being the best model.

Li & Shi [35] used a semi-supervised learning approach to train the DETR model to identify *Brassica chinensis* at the early stages of growth. The study was conducted in a greenhouse using five spherical cameras positioned at a height of 2.5 m and a UAV system equipped with a Sony action camera. 439 images at $1{,}920 \times 1{,}024$ pixels containing 7,399 objects were collected, and 80 % were allocated for training and 20 % for testing. The best result achieved was 0.779 mAP.

Borgogno et al. [36] developed a system for multiclass weed detection at different growth stages. Image collection was carried out in corn and tomato crops, where they were captured using a drone at 11 m height at a resolution of $6{,}000 \times 3{,}376$ and GSD = 0.17 cm/px at approximately 1:00 p.m. during the years 2020 and 2021. A dataset of 1,133 and 1,845 images of corn and tomato crops with 10 weed species, respectively, was created. The DETR architecture was trained with the GAN model to augment the training data. The best result was 0.513 mAP. The final model weighed 166.5 MB, and inference took 117.45 s on 2,390 corn images.

Valicharla et al. [37] trained the Swin-Transformer tiny model for knotweed classification using RGB images captured by drones. The study was conducted in six plots in Virginia, USA, between 2022 and 2024. The DJI Mavic 2 Enterprise Advanced drone was used, flying at 16 different altitudes ranging from 5 to 80 m. In the image collection, 26,735 images were extracted from 4 K videos, processed into selected $256 \times 256$ fragments, and classified into three classes: giant and Japanese knotweeds at the vegetative stage and knotweeds at the flowering stage. The dataset was split into 70 %, 20 %, and 10 % for training, validation, and testing. The following hyperparameters were used: lr = 0.001, epoch = 300, dropout = 0.2, opt = AdamW, batch size = 128, and weight decay = 0.05, using the NVIDIA RTX 3090 GPU for model training. The results indicated that the optimal drone height was 20 m, where the model achieved a precision of 99.47 %, an accuracy of 99.47 %, and a recall of 99.48 %. The mAP metric is not reported in this study.

Guo et al. [38] proposed a DETR-based architecture called RMS-DETR (with 40.8 M params) to identify weeds in rice crops during the tillering stage and facilitate herbicide spraying using drones. The work was carried out in China in 2022. The DJI M300 drone was used to capture weeds at a height of 30 m with a resolution of $8{,}102 \times 5{,}460$ and a ratio of 0.375 cm/px. 171 RGB images were obtained and extracted into 3,094 $600 \times 600$ sub-images. In the annotation process, three classes were labeled: field ridge, consecutive patches of barnyard grass, and single barnyard grass. The dataset was split 70:20:10 for training, validation, and testing, respectively, and four data augmentation techniques were employed. The architecture was modified to extract high- and low-level semantic information, reducing computational redundancy and inference time. The model was trained with an NVIDIA RTX 5000 GPU and the following hyperparameters: epochs = 500, batch = 4, opt = SGD,

lr = 0.001, momentum = 0.9, and weight decay = 1e-4. The model achieved 0.792 mAP on the validation set and 8.19 ms inference time, where the single barnyard grass class penalized performance due to its limited size.

Arriola-Valverde et al. [39] conducted a comparative study of three deep learning algorithms for coffee plant detection using images captured by Unmanned Aerial Systems (UAS). The dataset was collected from six coffee farms in the Central Valley of Costa Rica, spanning a variety of climatic and topographical conditions. Images were captured by a UAS system consisting of the DJI Matrice 310, DJI Matrice 600, and DJI Mavic 2 Pro drones, and the DJI Pilot 2 app was used to automate flights. Images were captured at an altitude of 30 m and with a ground sampling distance (GSD) of 1–3 cm/px. A total of 1,368 images resized to 640 × 640 were obtained. Coffee class annotation and data augmentation were performed using Roboflow, and the dataset was then split 80:10:10 for training, validation, and testing. The trained algorithms were Deep Forest (32.1 M params), RT-DETR (32 M), and YOLOv9 (58 M) using lr = 0.001 and optimizer = SGD. The Deep Forest model was trained for 75 epochs and achieved approximately 0.8 mAP. The RT-DETR and YOLOv9 models were trained for 200 epochs, obtaining 0.947 and 0.911 mAP, respectively, where RT-DETR was the best model.

In García-Navarrete et al. [40], YOLO (v8s, v9s, v10s, and v11s versions) and RT-DETR-1 architectures were trained and compared to detect four types of weeds (*Lolium rigidum, Sonchus oleraceus, Saolanun nigrum, and Poa annua*) using images of maize. The experiment was conducted in Spain in 2024 within a climate-controlled greenhouse. A support trolley was used to acquire the RGB images (6,000 × 4,000) without a drone. The trolley supports the vision system, consisting of a Canon 850D SLR camera positioned 1 m above the crop substrate, resulting in 1,640 images. The models' training utilized an NVIDIA Tesla V100 GPU with PyTorch 2.5.1. The YOLOv9s model had the best overall performance, with an mAP@0.5 of 0.834 (YOLOv11s reached a 0.824), whereas RT-DETR-1 had an mAP@0.5 of 0.828, making both models suitable for implementation in precision weeding systems.

Regarding works based on YOLO architecture, Deng et al. [31] compared YOLOX, YOLOv8, and DINO (Transformer) models for weed detection in cotton fields. The dataset used contains 6,664 images of eight weed types collected in 2021 and 2022 with hand-held color cameras (no drone), including: Waterhemp, Carpetweed, Morninglory, Goosegrass, Spotted Spurge, Palmer Amaranth, Purslane, and Ragweed. Images in JPG format were resized to 800 × 800 and split into 75 %, 5 %, and 20 % for training, validation, and testing. Images were manually labeled using VGG Image Annotator v2.10. The training environment was PyTorch on an NVIDIA RTX A6000 GPU. The best result was 0.987 mAP with the DINO model, while YOLOX and YOLOv8 achieved 0.974 and 0.966, respectively. Furthermore, in this study, 36 publicly available image datasets for weed recognition are identified, where 8 of them contain UAV images, and most consider only one generic weed class.

Pavithra et al. [41] deployed a drone equipped with two cameras to detect weeds and diseases using separate YOLOv8 models. The models were trained on two datasets comprising 3,713 ground-level images (no drone), resized to 416 × 416, of 50 categories that cover a range of crops, weeds, and diseases, with class imbalances. Image augmentation techniques were employed to reduce the impact of class imbalance. Both models were saved and subsequently uploaded to a drone's SD card for the single-board computer (SBC, Nvidia Jetson B0 model). Each model is 51 MB, achieving 0.87 mAP for identifying weeds and 0.47 mAP for detecting diseases.

Rehman et al. [42] proposed a YOLOv5-based architecture for weed detection in soybean crops. The network backbone was modified to capture image complexities that improve model learning. A total of 2,247 images were captured using the DJI Phantom 4 Multispectral drone (altitude not mentioned), and weeds (1 class) were labeled using the Roboflow tool. The dataset was split into 70 %, 20 %, and 10 % for training, validation, and testing. Each image was cropped to 640 × 640. The model was trained for 200 epochs using the SGD optimizer, with a learning rate 0.01. Two NVIDIA GeForce RTX 2080 Ti GPUs, the PyTorch v2.0.1 framework, and Python v3.11.4 were used. The YOLOv5 model with 3.99 M parameters achieved an accuracy of 73.9 mAP and an inference speed of 7.1 ms.

Saini & Nagesh [43] used the YOLOv5 model for multiclass detection of two weed species in cotton crops. 7,578 460 × 460 images captured with a Sony DSC-WX150 camera (not a drone) were collected under different weather conditions from 7 a.m. to 7 p.m. After selecting and labeling three classes in the dataset, 1,737 images were retained, which were split 80:20 for training and validation. Three versions of the YOLOv5 model (v5m, v5l, and v5x) were trained for 300 epochs with the SGD optimizer and a Learning rate 0.01. The training environment was Google Colab equipped with a 14.2 GB GPU. The largest model YOLOv5x (with 97.2 M parameters) achieved the best performance of 0.875 mAP and 72.5 mAP:50:95. However, the model had difficulties with the horse purlane class due to its low representation, and the purple nutsedge class obtained a lower accuracy than the cotton class, despite having a good number of specimens.

Silva et al. [44] evaluated some deep learning algorithms for weed detection and segmentation in soybean and bean crops using 793 RGB images (640 × 640) captured by the DJI Phantom 3 Standard UAV at 20 m height. The classes bean, soybean, and weeds (in general) were annotated and divided into portions of 75:12:13 for training, validation, and testing. The U-Net, Mask R-CNN, and YOLO models (v5, v7, and v8) were trained. The hyperparameters used for the YOLO models were: batch size = 16, optimizer = SGD, lr = 0.01, and epochs = 500. In the weed detection task, the best model was YOLOv8s (with 11.2 M parameters), which achieved an accuracy of 0.970 mAP. In contrast, the YOLO v5s and v7 variants scored 0.945 and 0.954 mAP, respectively.

Gao et al. [45] conducted a study for the real-time detection of corn panicles (maize tassels) using images captured at 20 m height by the DJI Inspire 2 drone. The dataset comprises 900 images resized to 640 × 640 and split into 3:1:1 (60 %, 20 %, 20 %) for training, validation, and testing. The architecture used was YOLOv5-T, incorporating attention modules (CBAM and ASPP) to improve *panicle* detection in complex agricultural environments. The model was trained for 300 epochs with Python v3.9, the PyTorch v1.11 Framework, and an NVIDIA RTX 3090 graphics card. The best model achieved 0.987 mAP and an inference speed of 23 ms.

Arsalan et al. [46] trained two YOLO v5 and v6 models for the precise spraying of tobacco plants. 7,000 images (640 × 480) of the crop were collected over 40 days using the Mavic Mini drone at a height of 4 m. The methodology used object detection to identify crops and segmentation to estimate plant dimensions in real time. The dataset was split into 70 % for training and 30 % for testing. The following hyperparameters were used for training: optimize SGD, lr = 0.01, momentum = 0.937, batch = 16, and epochs = 1,000. The YOLO v5s and v6s models achieved the same results of 0.979 mAP, but YOLOv6s outperforms YOLOv5s in terms of inference speed with 9.2 ms.

Tian et al. [47] conducted a study to determine the appropriate planting density to maximize cotton crop yield in a 3.27-ha field in Mississippi, USA. Two datasets, comprising 275 and 226 images, were prepared by capturing the images 2 and 6 weeks after planting at a height of 10–15 m using the Phantom 4 Advanced drone. The YOLOv7 model was used to quantify the plants, and then the Hough transform was employed to remove weeds from the image. This model was trained for 200 epochs using a batch size of 6, lr = 0.01, and the SGD optimizer. The first dataset achieved the best metric, reaching 96.9 % mAP.

Various approaches and advances are observed in the related work to detect weeds in agricultural fields widely using convolutional neural networks (CNN) and, more recently, Transformers. Furthermore, it is observed that most studies using UAV images chose to designate a single generic class, such as “weeds” to refer to all types of weeds present in crops, rather than the five classes used in our study. However, a small number of studies also address certain weed categories in a limited way, which is essential for specific treatments in diverse real-life contexts.

Regarding the publicly available datasets, we have found 12 of them [48]–[59] that contain UAV images in soybean, corn, sesame, beet, sorghum, rice, alfalfa, cotton, tobacco, and wheat crops. However, we have not found images in potato crops. Regarding the kinds of weeds, one dataset [57] contains images of Broadleaf dock (*Rumex obtusifolius*), and another [58] considers grass and Broadleaf (in general). There are no images of the Dandelion class, specifically. Thus, no dataset considers these three weeds together in potato crops, as in our study. Besides, the differences in spectral reflectance, color, textures, and plant morphology between the testing and training data in different periods (seasons and years) and land locations cause distributional shifts that present confounding factors for the model [60]. It is important to consider the similarity factor between the source and target domains to ensure the model’s success [5]. This shows the need to create an up-to-date custom dataset for potato crops, considering the native weeds of our northern region in Ecuador, which have their own morphological characteristics.

# 2 Materials and methods

## 2.1 KDD methodology

The Knowledge Discovery in Databases (KDD) methodology [30] was used for UAV image data analysis, which consists of five phases: data collection; data selection, preprocessing, and transformation; data mining; evaluation and interpretation. Figure 1 presents the study’s general workflow, facilitating understanding of the KDD stages and their interconnections. The KDD phases are detailed below.

### 2.1.1 Data collection

RGB images (5,472 × 3,648) were collected from 8 different potato crop fields in Carchi and Imbabura (Ecuador) to incorporate variability in climatic conditions, land, and plants. In total, 3,579 images were acquired, trying to balance the different weed classes, of which 2,447 images were captured in 2023 and 1,132 in 2024. The DJI Mavic 2 Pro UAV was used with the Android DroneDeploy application v5.7 to configure all flight parameters: flight height, 9–10 m; flight

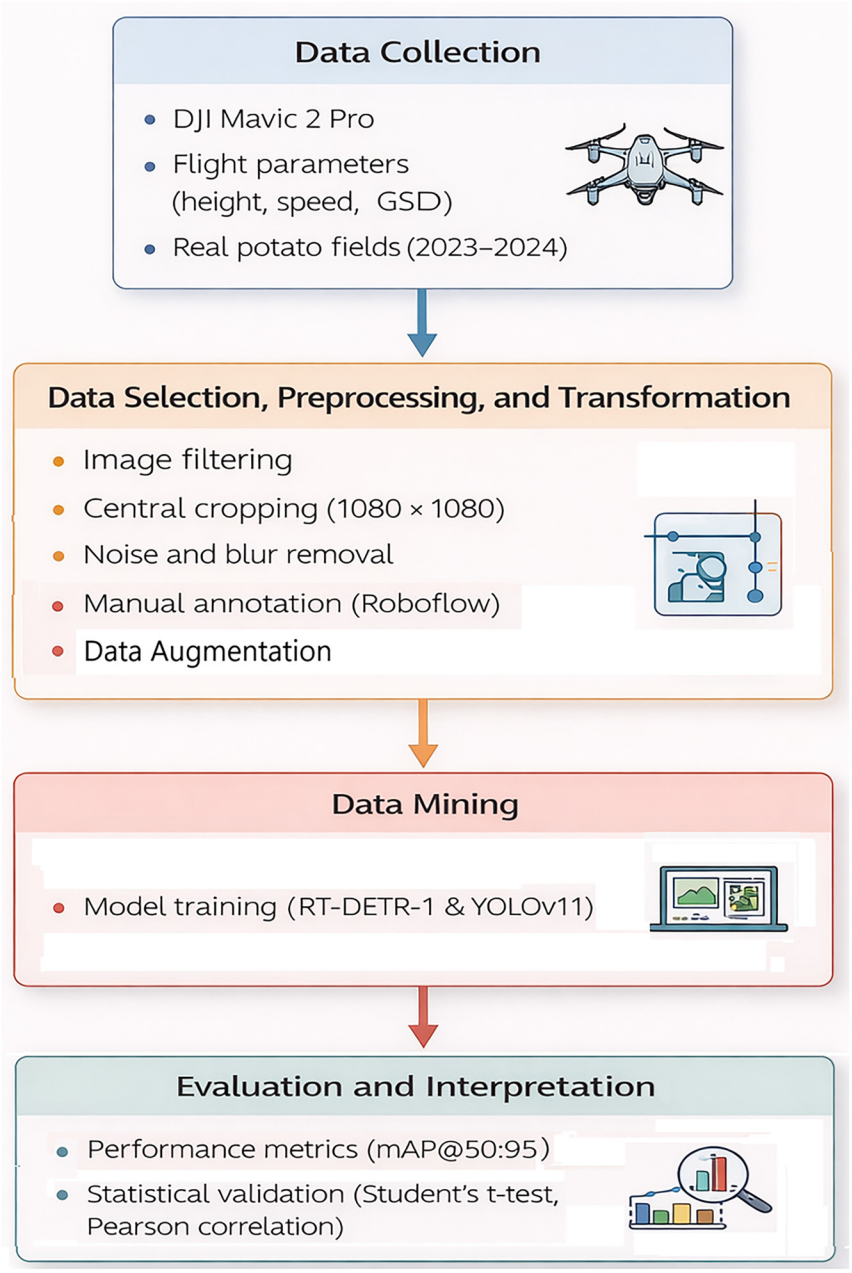



**Figure 1:** General workflow of the proposed methodology, illustrating the KDD stages from UAV image acquisition to model evaluation and statistical validation.

speed, 1 m/s; and cm/px ratio, 0.25. The distance between the camera and the ground is not strictly fixed to create more image variations. The flight height was selected because the best results were observed on images acquired at 10 m [47], representing a compromise between ground sample distance (GSD) and ground coverage [61]. It allows for capturing a larger area; hence, the efficiency of the UAV flying is much higher [62]. Likewise, flight speed must be substantially reduced at relatively low flight altitudes to avoid motion blur [61]. Figure 2 illustrates images collected in potato fields with various types of weeds.

### 2.1.2 Data selection, preprocessing, and transformation

Images not suitable for the research were discarded according to the following criteria: (i) images that do not contain weeds, (ii) images with an exaggerated density of weeds, (iii) adjacent images without relevant changes, and (iv) blurred



(a) (b)

**Figure 2:** Examples of UAV RGB images (5,472 × 3,648) taken at 9–10 m height by the DJI Mavic 2 Pro UAV in potato fields in Carchi (a) and Imbabura (b) in northern Ecuador; (b) example of an image cropped in the center with a size of 1,080 × 1,080.

images. After image selection, 700 RGB images were finally maintained.

These images were then cropped to the central portion with a size of 1,080 × 1,080 as shown in Figure 2(b). The central region of the image usually contains most of the relevant information, offering several advantages, especially in terms of processing efficiency and data relevance [63], such as: reducing input size and computational cost; removing visual noise and ensuring that the model learns from regions containing significant information; reducing the likelihood of the model learning spurious or irrelevant patterns, thus improving its generalization ability; ensuring that objects of interest occupy a more significant proportion of the image, making model learning easier; facilitating data labeling and annotation, making them faster and more accurate. On the other hand, peripheral areas of the image may contain redundant information, with no relevant or irrelevant changes. These areas may contain noise (such as shadows, lens distortion, or unnecessary elements) that can confuse the model. Lens distortion affects these peripheral areas of the images the most, while it is minimal in the central region, making them less prone to geometric deformations caused by distortion. The square size of the image (1,080 × 1,080) is to take advantage of deep learning library optimizations when performing matrix calculations.

After selecting and cropping the images, the plants were manually annotated into the five categories of Potato, Dandelion, Kikuyu, Broadleaf dock, and “other weeds” for species that were difficult to identify visually due to their small size or the presence of other species that were not relevant in the visited fields. This generic category is because in real crop fields, it is very likely that plants that were not considered in the training dataset will appear.

The online platform Roboflow was used for this image annotation task and to keep track of individual copies and versions of the dataset [64]. The annotation process for the 700 images took 230 h of work, almost 20 min for each image.

After annotating, two dataset versions were prepared as detailed in Table 1. The most unbalanced dataset was version one (v1), which shows an overrepresentation of the Potato class (41.67 %) compared to the other categories, mainly Kikuyu (9.75 %). However, in version two (v2), an attempt was made to improve the balance of each class by eliminating images/instances of the majority classes. Thus, each class has a representation greater than 12 %, which can be useful for extracting relevant plant features and training the model. The classes are not completely balanced because several plant classes are eliminated in different proportions when an image is eliminated.

The annotated image dataset was split into training, validation, and testing sets at 80 % (560 images), 10 % (70 images), and 10 % (70 images), respectively. The 80/10/10 split is a good balance in general, where one wants to maximize data usage for model training without sacrificing adequate validation (hyperparameter tuning) or testing of the model’s generalization ability on unseen data [65].

**Table 1:** Summary of the two versions of the weed and crop dataset. Version 1 is unbalanced, while version 2 is more balanced.

| Type of plant | Dataset v1 | | Dataset v2 | |
|---|---|---|---|---|
| | Number of plants | % | Number of plants | % |
| Dandelion | 8,211 | 17.14 | 3,022 | 12.92 |
| Kikuyo | 4,672 | 9.75 | 3,155 | 13.49 |
| Broadleaf dock | 6,676 | 13.93 | 4,247 | 18.16 |
| Other weeds | 8,380 | 17.49 | 5,461 | 23.35 |
| Patato (crop) | 19,960 | 41.67 | 7,499 | 32.06 |
| **Total** | **47,899** | **100** | **23,384** | **100** |

Data augmentation was used to enhance plant representation and improve model training by increasing data variability without affecting object characteristics. Data augmentation was applied after manual annotation to increase the number of training instances, improve class balance, and enhance model generalization. Geometric and photometric transformations were applied consistently to images and their corresponding bounding boxes, ensuring that each augmented image preserved the spatial and semantic integrity of the labeled plants. Specifically, the following operations were performed: HSV hue increase of 0.015, saturation increase of 0.7, HSV brightness increase of 0.4, horizontal/vertical image shift with 10 % probability, image scaling of 0.5, image flipping from left to right with 50 % probability, 100 % for image mosaicking, 40 % random erasure of a portion of the image, and 1.0 fractional cropping [66]. In total, nine types of data augmentation were applied, yielding 5,040 images ($560 \times 9$) for model training. Thus, the original training set was expanded by a factor of 9, as shown in Table 2, yielding a larger, more diverse dataset while preserving the semantic integrity of the labeled objects. This augmentation strategy ensures sufficient representation of minority classes while maintaining realistic field conditions.

It is important to clarify that the instance counts reported before and after data augmentation correspond exclusively to the training subset (80 % of the dataset). Data augmentation was applied only after the dataset split, ensuring that validation and test sets remained unaltered to provide an unbiased evaluation of model performance.

The data augmentation process significantly increased the number of annotated instances while preserving the original class distribution trends. Although augmentation improves the representation of minority classes, it does not fully eliminate class imbalance; rather, it mitigates its effect by increasing intra-class variability. Additionally, the applied transformations simulate realistic variations in illumination, scale, and spatial configuration commonly observed in UAV-based agricultural scenarios. This improves the robustness and generalization capacity of the trained models when exposed to unseen field conditions.

**Table 2:** Number of annotated plant instances before and after data augmentation for dataset versions v1 and v2 (training set only). Instance counts refer to annotated bounding boxes rather than images.

| Dataset | Class | Instances before augmentation | Instances after augmentation (×9) |
|---|---|---|---|
| v1 | Dandelion | 8,211 | 73,899 |
| | Kikuyu | 4,672 | 42,048 |
| | Broadleaf dock | 6,676 | 60,084 |
| | Other weeds | 8,380 | 75,420 |
| | Potato (crop) | 19,960 | 179,640 |
| | **Total** | **47,899** | **431,091** |
| v2 | Dandelion | 3,022 | 27,198 |
| | Kikuyu | 3,155 | 28,395 |
| | Broadleaf dock | 4,247 | 38,223 |
| | Other weeds | 5,461 | 49,149 |
| | Potato (crop) | 7,499 | 67,491 |
| | **Total** | **23,384** | **210,456** |

### 2.1.3 Data mining

a) RT-DETR-1 Architecture

The Real-Time Detection Transformer (RT-DETR) deep learning algorithm [28] is an end-to-end architecture designed to take advantage of the advantages of its predecessor model, DETR [23] and significantly improve in terms of computational efficiency and accuracy. DETR is a hybrid encoder-decoder architecture that combines elements of convolutional neural networks (CNNs) and Transformers. This combination leverages both technologies' strengths to solve the object detection task efficiently and end-to-end. The main role of the CNN is to extract relevant features from the image (backbone), such as corners, edges, textures, etc. The Transformer network processes these features through its attention mechanisms (self-attention), which are mathematical operations performed on a dataset to pay attention to specific parts of it. Thus, CNNs capture local features (fine details), while the Transformer model captures global dependencies (context). To generate the final prediction, a Feed Forward Network (FFN) is used, which, in our case, returns as output the class and the coordinates of the bounding box of the detected object [23]. DETR simplifies the object detection process by removing additional components used in CNNs, such as spatial anchors or non-maximum suppression (NMS), to predict object locations directly.

In contrast, RT-DETR is optimized for real-time detection, addressing the efficiency and speed limitations present in DETR. The computational cost is primarily reduced through the following strategies:

- Decoupled encoder-decoder: These modules work more independently, processing information more efficiently.
- Use of lightweight and efficient backbones such as CSPNet or Swin Transformer [22], instead of ResNet.
- Integration of optimized (lightweight) Attention mechanisms with multi-scale structure (like FPN – Feature Pyramid Network), improving the detection of objects of different sizes, even small and complex ones.

The RT-DETR architecture has several variants in the PyTorch framework that adapt to different needs regarding accuracy and speed. These variants differ primarily by model size and the backbone used, and some that tend to have higher accuracy are the following:

- RT-DETR-L (Large): This is a moderately sized variant that balances accuracy and speed, suitable for real-time applications requiring high accuracy.
- RT-DETR-X (Extra-Large): The largest variant, designed to maximize accuracy, although with a lower inference speed. It is used in scenarios where accuracy is the highest priority.

#### 2.1.3.1 Training the RT-DETR model

The RT-DETR-L variant of the RT-DETR-1 model was employed in this project, aiming to strike a balance between accuracy and speed, with the goal of its potential application in embedded devices or real-time scenarios. This version has 32 M parameters with 681 layers, achieves 114 FPS on a T4 GPU, and takes up 64 MB of disk space [67]. In this study, the model was trained online on the Kaggle platform with the NVIDIA TESLA P100 GPU with 16 GB VRAM for 12 h of continuous use [68]. The development environment used was Python v3.10, CUDA v12.6, PyTorch v2.4.0, and Ultralytics v8.3.49.

Five variants of the RT-DETR-L model were trained, applying different hyperparameter settings, image size, and class balancing (v1 and v2 of the dataset) to each one as indicated in Table 3. The resolutions of 640, 800, 960, and 1,020 were obtained by resizing the central part of the cropped image (1,080 × 1,080) as indicated before.

The baseline model (variant 1) was trained with the default hyperparameters from the Ultralytics library and with the unbalanced v1 dataset. Starting with variant 2, dropout was activated at 0.5 to avoid model overfitting, the warm-up epochs were extended to 5, and the learning rate was adjusted according to the cosine curve to achieve good model convergence [66]. The batch size, early stopping, and model epochs remained the same across all model variants. The most significant adjustments were made starting with variant 3, which used the most balanced dataset (v2), the Adam optimizer, and progressively increased image resolution. These adjustments directly impact model performance.

b) YOLOv11 architecture

YOLOv11 [20] is a modern architecture proposed by the Ultralytics team to improve the model in terms of accuracy and speed, making it high performing in computer vision tasks such as object detection, instance segmentation, and classification. Key features introduced in this release include improved feature extraction and processing, an architecture optimized for performance and speed, and increased accuracy with fewer parameters. These improvements are primarily due to a refined and optimized architecture, primarily of the backbone (with efficient convolutions and parameter reduction techniques) and the neck (the intermediate layer between the backbone and the prediction head), which facilitates the fusion of multi-scale information, contributing to more accurate detection of large and small objects. YOLOv11 has five variants for different computer vision tasks, including: nano (YOLOv11n), small (YOLOv11s), medium (YOLOv11m), large (YOLOv11L), extra-large (YOLOv11x) [69]. The variants tested in this work are YOLO11s and YOLO11m, considering a balance between accuracy and computational efficiency for general object detection applications with a view to their possible application in embedded or real-time devices. YOLO11s with 9.4 million parameters achieves an mAP of 47.0 on the COCO dataset and occupies 18.4 MB in size. In comparison, YOLO11m with 20.1 M parameters increases the accuracy to 51.5 mAP at the cost of higher complexity with 38.8 MB.

**Table 3:** Hyperparameters used in training the five variants of the RT-DETR-L model.

| Hyperparameters | Variant 1 | Variant 2 | Variant 3 | Variant 4 | Variant 5 |
|---|---|---|---|---|---|
| Dataset version | v1 | v1 | v2 | v2 | v2 |
| Epochs | 100 | 100 | 100 | 100 | 100 |
| Image size | 640 | 960 | 800 | 960 | 1,020 |
| Batch size | 8 | 8 | 8 | 8 | 8 |
| Early stopping | 20 | 20 | 20 | 20 | 20 |
| Optimizer | AdamW | AdamW | Adam | Adam | Adam |
| Learning rate | 0.001111 | 0.002500 | 0.000714 | 0.001111 | 0.001111 |
| Dropout | 0.0 | 0.5 | 0.5 | 0.5 | 0.5 |
| Learning rate scheduler | Linear lr | Cosine lr | Cosine lr | Cosine lr | Cosine lr |
| Warmup epochs | 3 | 5 | 5 | 5 | 5 |

#### 2.1.3.2 Training the YOLOv11 model

The YOLOv11s and YOLOv11m variants were utilized to balance speed, size, and accuracy. The training process employed was performed in the same way as the RT-DETR architecture, using the Kaggle platform with the same resources and libraries. Five variants of the YOLOv11 model (small and medium) were trained, applying different hyperparameter configurations, image size, and class balancing (v2 of the dataset) to each one, as indicated in Table 4. Thanks to its lighter weight, the YOLOv11s version was tested with three different configurations. In contrast, due to its weight, YOLOv11m was tested with two variants.

### 2.1.4 Evaluation and interpretation of models

The performance of the RT-DETR-1 and YOLOv11 models was evaluated with the common metrics for the object detection task, which are: mAP, mAP@50:95, precision, and recall [70].

Precision (P) calculates the ratio of correct positive predictions to the total positive forecasts made by the model; it answers the question: How reliable are the model's positive predictions? Precision is calculated using Equation (1), where TP stands for True-Positive, representing the number of weed samples correctly detected by the model. FP stands for False-Positive, indicating the number of incorrectly predicted weed samples [38].

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

Recall ($R$) measures the proportion of true positive cases that the model can correctly identify; it answers the question: How many of the true positive cases are detected by the model? See Equation (2), where FN stands for False-Negative, referring to the number of undetected weed samples.

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

In object detection, these metrics are calculated considering an IoU (Intersection over Union) threshold, which is a metric that measures how well a predicted box overlaps with the actual box, and is critical for practical applications [32]. A prediction is considered correct (TP) if the IoU between the predicted and actual bounding box exceeds a defined threshold (e.g., IoU $\geq 0.5$).

The average precision (AP) metric combines precision and recall into a single metric, providing an overview of how well the model detects (localizes) and classifies objects in a dataset. The AP calculation is based on the precision and recall values obtained at different prediction confidence thresholds.

The mean average precision (mAP) metric is the average of the AP calculated for all classes in a dataset, see Equation (3), where $N$ is the number of classes, which is five in this case. The mAP assesses the overall quality of object detection models and represents how well the model can detect objects of various classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{3}$$

For a more comprehensive and stable evaluation, mAP@50:95 is used, which calculates the average AP over a range of IoU thresholds from 0.5 to 0.95, in increments of 0.05.

Finally, to complement the data analysis and to contrast the number of plants counted from the model predictions (RT-DETR-1/YOLOv11) and the ground truth, the Student's $t$-test and Pearson correlation coefficient ($r$) statistical tests were used to verify whether there are statistically significant differences and a linear association between both measurements [71], respectively.

**Table 4:** Hyperparameters used in training the five variants of the YOLOv11 model (small and medium).

| Hyperparameters | Variant 1 | Variant 2 | Variant 3 | Variant 4 | Variant 5 |
|---|---|---|---|---|---|
| Model variant | Small | Small | Small | Medium | Medium |
| Dataset version | v2 | v2 | v2 | v2 | v2 |
| Epochs | 100 | 100 | 100 | 100 | 100 |
| Image size | 640 | 800 | 960 | 960 | 1,020 |
| Batch size | 32 | 32 | 32 | 16 | 16 |
| Early stopping | 20 | 20 | 20 | 20 | 20 |
| Optimizer | AdamW | AdamW | Adam | Adam | AdamW |
| Learning rate | 0.001 | 0.0001 | 0.001 | 0.0001 | 0.0001 |
| Dropout | 0.0 | 0.5 | 0.5 | 0.5 | 0.5 |
| Learning rate scheduler | Linear lr | Cosine lr | Cosine lr | Cosine lr | Cosine lr |
| Warmup epochs | 3 | 5 | 5 | 5 | 5 |

# 3 Results

## 3.1 Performance metrics of the RT-DETR-1 model

Table 5 summarizes the results achieved in the validation stage of the five RT-DETR (Large) model variants. The best model obtained was variant 5 (1,020 × 1,020), achieving a mean Average Precision (mAP) of 0.809. However, these results are similar to those obtained by variant 4 (960 × 960). This suggests that, beyond a certain point, increasing the image resolution does not significantly improve the model's performance. Comparing the baseline model (variant 1) with the others, the hyperparameter and image size adjustment (variant 2) increased the model's performance by 3.5 % in terms of mAP, while the best class balancing (variant 3) increased it by 7.3 %. Compared to model variant 4, the increase was 9.2 %, whereas variant 5 improved by 9.3 %.

**Table 5:** Performance results of the five RT-DETR-L variants on the validation set (10 %, 70 images) for 100 epochs.

| Variant | Dataset | mAP | mAP@50:95 | Precision | Recall |
|---|---|---|---|---|---|
| 1 (baseline) | v1 | 0.740 | 0.412 | 0.732 | 0.725 |
| 2 | v1 | 0.766 | 0.443 | 0.766 | 0.743 |
| 3 | v2 | 0.794 | 0.441 | **0.789** | 0.751 |
| 4 | v2 | 0.808 | 0.448 | 0.761 | **0.779** |
| 5 | v2 | **0.809** | **0.448** | 0.786 | 0.773 |

The best values are in bold.

Regarding the rest of the metrics, model variant 5 achieves the highest value in mAP@50:95 (0.448) and is very similar to the best values (in bold) in precision and recall, with 0.786 and 0.773, respectively. All model variants, except the baseline, achieve a very similar mAP@50:95 around 0.44, indicating good performance at different overlap thresholds (50 %–95 % in IoU) for the object detection task (crop/weeds).

Figure 3 shows the performance graphs during training of the five variants of the RT-DETR model, respectively,

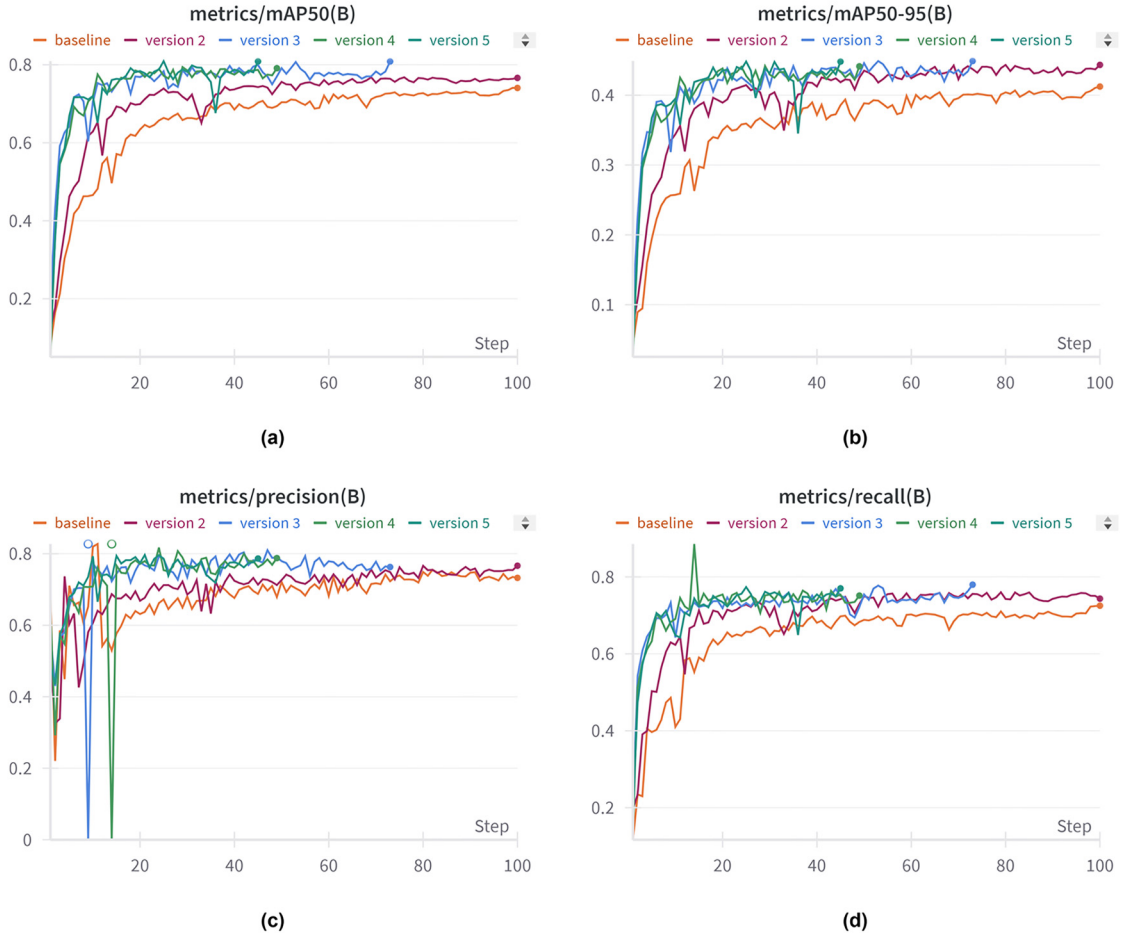



**Figure 3:** Performance metrics of the five variants of the RT-DETR model on the validation set (10 %): (a) mAP, (b) mAP@50:95, (c) precision, and (d) recall.

**Table 6:** Performance of the RT-DETR model (variant 5) on the validation (10 %) and test (10 %) sets.

| Approach | Metric | Validation set | Test set |
|---|---|---|---|
| By classes | Potato AP | 0.913 | 0.945 |
| | Kikuyo AP | 0.846 | 0.775 |
| | Dandelion AP | 0.919 | 0.863 |
| | Broadleaf dock AP | 0.838 | 0.896 |
| | Other weeds AP | 0.528 | 0.526 |
| By model | mAP | 0.809 | 0.801 |
| | mAP@50:95 | 0.448 | 0.443 |
| | Precision | 0.786 | 0.778 |
| | Recall | 0.773 | 0.765 |

on the validation set (10 %) using the four metrics: mAP, mAP@50:95, precision, and recall.

Table 6 presents the average precision (AP) of the best RT-DETR model (variant 5) for each class in the validation and test sets (data not seen during training), both consisting of 10 % of the dataset (70 images each).

Considering the test set, the Potato class has the highest AP = 0.945, due to its good representation of examples in the dataset, followed by the Broadleaf dock class with AP = 0.896 and Dandelion with AP = 0.863. The Kikuyu class has an acceptable AP = 0.775, while the lowest class is "other weeds" with AP = 0.526 because this generic class has a limited size and encompasses different types of weeds that could not be visually identified as a specific category. This makes it difficult for the Transformer model to find and learn unique patterns of the plants in this category, significantly harming the mAP of the model. Still, the model obtained an mAP = 0.801, indicating that, on average, the model detects all five classes well. In addition, an mAP@50:95 = 0.443, precision = 0.778, and recall = 0.765 were obtained.

The performance of the model on the validation and test sets (Table 6) is comparable in term of mAP, i.e., the difference between these two-evaluation metrics is small (<1 %), suggesting that both data sets are representative, the model generalizes well to unseen data, and there is no evidence of overfitting or underfitting.

## 3.2 YOLOv11 model performance metrics

The results obtained from the YOLOv11 model are shown in Table 7. The best model was variant 4 medium (960 × 960), achieving a mean average precision (mAP) of 0.825. However, these results are similar to those obtained by variant 2, small (800 × 800), and variant 5, medium (1,020 × 1,020). This suggests that, beyond a certain point, increasing image resolution and model complexity does not significantly improve performance. Comparing the baseline model (variant 1, small) with the rest, adjusting hyperparameters and image size (variant 2, small) increased model performance by 2.4 % in terms of mAP, while with variant 3 (small), it only increased by 1.4 %. Concerning model variant 4 (medium), it increased by 2.9 %; with variant 5 (medium), it improved by 2.6 %.

Regarding the rest of the metrics, model variant 4 achieves the highest value in mAP@50:95 (0.457) and is very similar to the best values (in bold) in precision and recall with 0.757 and 0.760, respectively. Except for the baseline, all model variants achieve a similar mAP@50:95 around 0.45, indicating good performance at different overlap thresholds (50 %–95 % in IoU) for the object detection task (crop/weeds).

Figure 4 shows the training performance graphs of the five variants of the YOLOv11 model, respectively, on the validation set (10 %) using the four metrics: mAP, mAP@50:95, precision, and recall.

Table 8 presents the average precision (AP) of the best YOLOv11m model (variant 4) for each class in the validation and test sets (data not seen during training), both consisting of 10 % of the dataset (70 images each).

Considering the test set, the Potato class has the highest with AP = 0.942, due to its good representation of examples in the dataset, followed by the Dandelion class with AP = 0.883 and the Broadleaf dock class with AP = 0.863. The Kikuyu class has an acceptable AP = 0.791, while the lowest class is "other weeds" with AP = 0.564 because, as mentioned, this generic class has a limited size and encompasses different types of weeds that cannot be visually identified

**Table 7:** Performance results of the five versions of YOLOv11 on the validation set (10 %, 70 images) for 100 epochs.

| Variant | Model version | Dataset | mAP | mAP@50:95 | Precision | Recall |
|---|---|---|---|---|---|---|
| 1 (baseline) | Small | v2 | 0.802 | 0.437 | **0.766** | 0.722 |
| 2 | Small | v2 | 0.821 | 0.445 | 0.749 | 0.759 |
| 3 | Small | v2 | 0.813 | 0.449 | 0.738 | **0.776** |
| 4 | **Medium** | v2 | **0.825** | **0.457** | 0.757 | 0.760 |
| 5 | Medium | v2 | 0.823 | 0.456 | 0.764 | 0.760 |

The best values are in bold.

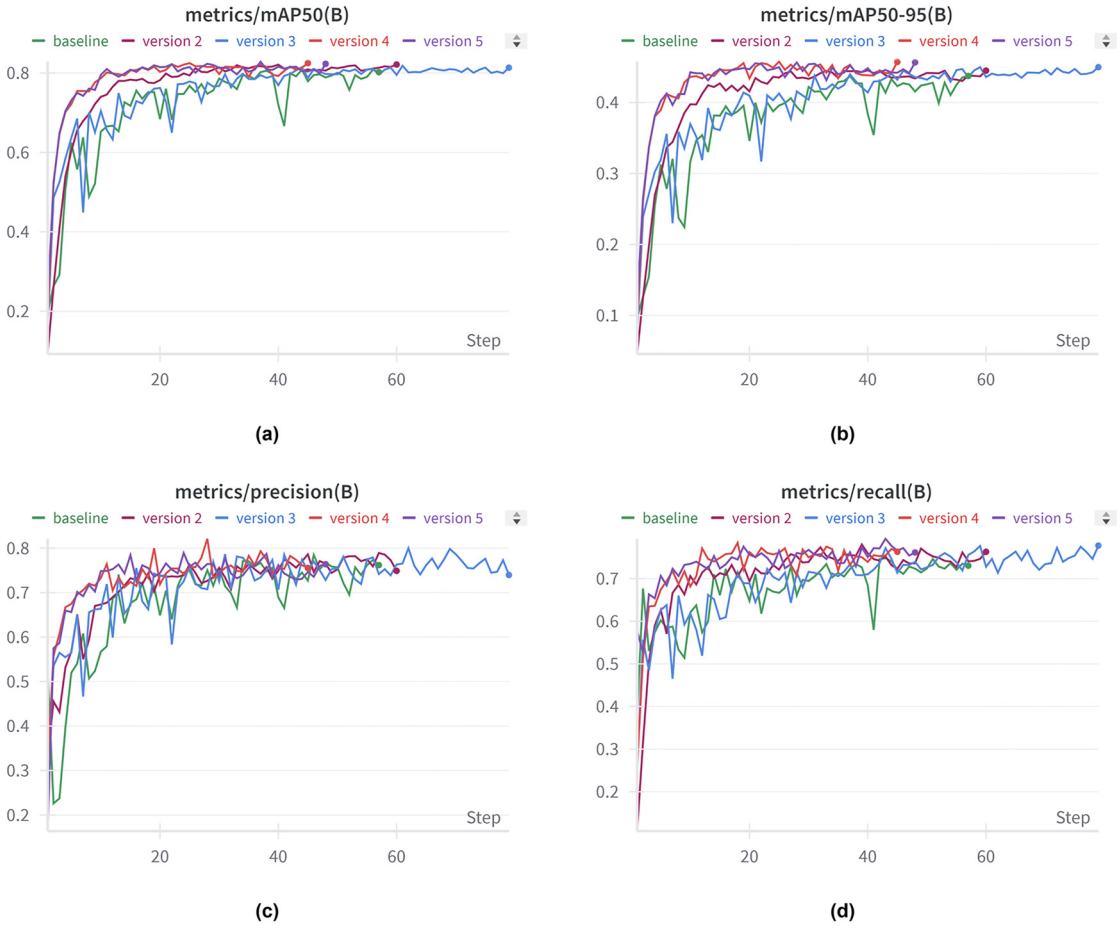



**Figure 4:** Performance metrics of the five variants of the YOLOv11 model on the validation set: (a) mAP, (b) mAP@50:95, (c) precision, and (d) recall.

**Table 8:** Performance of the YOLOv11m model (variant 4) on the validation (10 %) and test (10 %) sets.

| Approach | Metric | Validation set | Test set |
|---|---|---|---|
| By classes | Potato AP | 0.938 | 0.942 |
| | Kikuyo AP | 0.855 | 0.791 |
| | Dandelion AP | 0.933 | 0.883 |
| | Broadleaf dock AP | 0.834 | 0.863 |
| | Other weeds AP | 0.565 | 0.564 |
| By model | mAP | 0.825 | 0.809 |
| | mAP@50:95 | 0.457 | 0.448 |
| | Precision | 0.757 | 0.742 |
| | Recall | 0.760 | 0.745 |

as a specific category. This makes it difficult for YOLOv11 to find and learn unique patterns of the plants in this category, significantly harming the mAP of the model. Still, the model obtained an mAP = 0.809, indicating that, on average, the model detects all five classes well. In addition, an mAP@50:95 = 0.448, precision = 0.742, and recall = 0.745 were obtained. The performance of the model on the validation and test sets (Table 8) is comparable in term of mAP, i.e., the difference between these two-evaluation metrics is small (<2 %), suggesting that both data sets are representative, the model generalizes well to unseen data, and there is no evidence of overfitting or underfitting.

Figure 5 illustrates examples from the test set, the ground-truth annotations, and the predictions of the RT-DETR (variant 5) and YOLOv11m (variant 4) models. Overall, there is good agreement between the model predictions and the ground truth.

## 3.3 Statistical validation of model predictions

The Student's *t*-test is a parametric statistical test to determine, in this case, whether there is a significant difference in the means of the number of plants between the ground truth and the values predicted by the models (RT-DETR/YOLOv11). This test allows us to determine whether the model's predictions are objectively reliable. Here, they are considered dependent samples in the test since the manual assessments

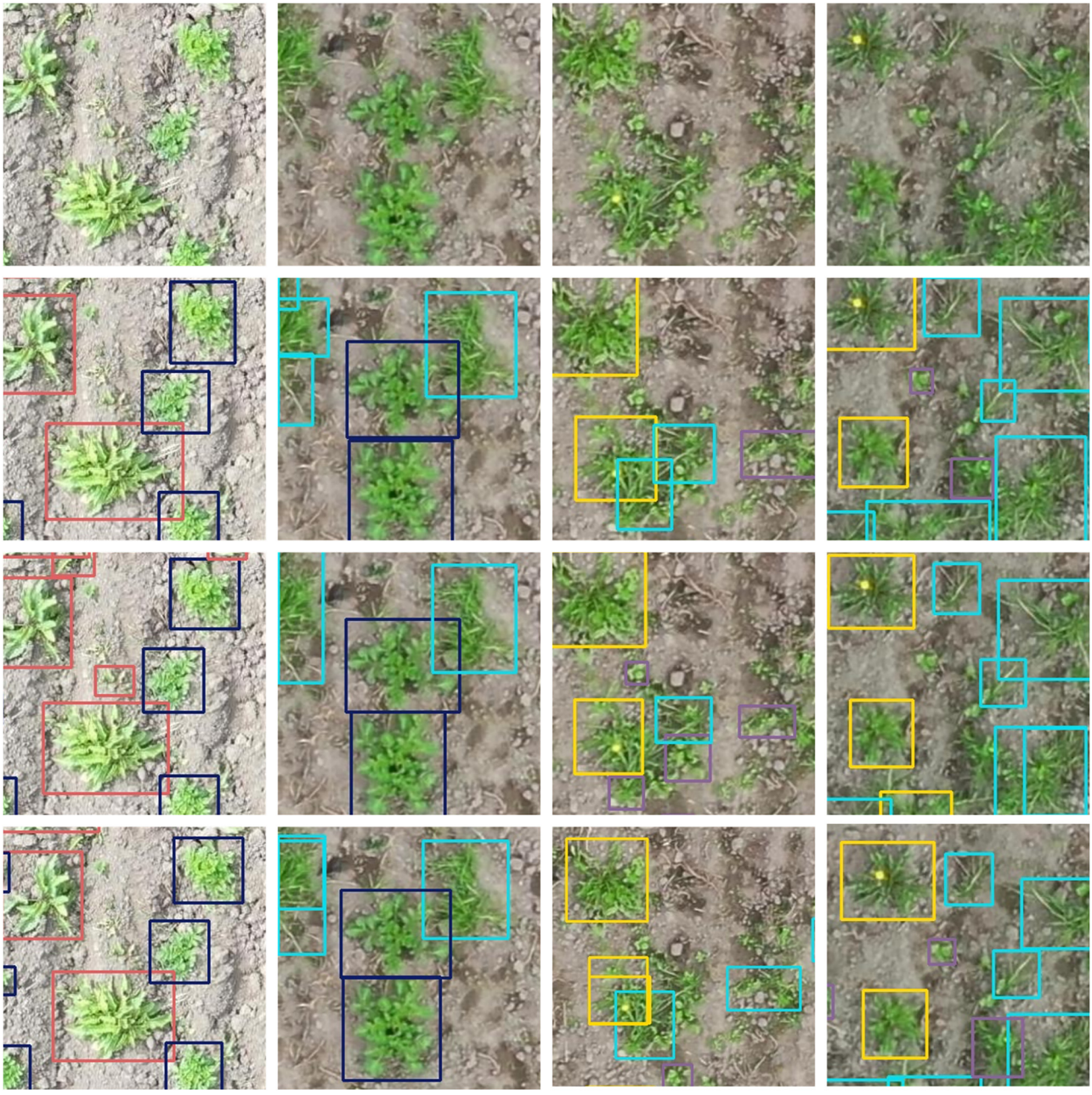


**Figure 5:** Examples of images from the test set extracted in RGB format (first row), ground truth (second row), RT-DETR v5 model predictions (third row), and YOLOv11m v4 model predictions (fourth row). The Potato is marked in blue, the Dandelion in yellow, the Kikuyu in light blue, the Broadleaf dock in red, and the other weeds in purple.

(ground truth), and those predicted by the model are carried out on the same set of images [72]. The null hypothesis $H_0$, and alternative hypothesis $H_1$ are established as follows:

$H_0$: $d = 0$. There is no significant difference between the means of the actual and predicted number of plants.
$H_1$: $d \neq 0$

The established decision rule is: if $p_value \leq 0.05$, then reject $H_0$.

Statistical analysis was performed on paired samples (ground truth and predictions) using a test set of 70 images (10 % of the total). The sample size was large enough ($n \geq 30$) to ensure, in accordance with the Central Limit Theorem (CLT), that the differences between observations

approximate a normal distribution [72]. This ensures the validity and robustness of the Student's *t*-test.

a) Statistical validation of RT-DETR

When performing the Student's *t*-test for the Potato class, a *p*-value of 0.688 was obtained. This indicates no significant differences ($p > 0.05$) between the quantification of this plant and the ground truth, suggesting that the model does not differ systematically from the manual count. The Pearson correlation coefficient $r = 0.971$ indicates a very strong linear association between both measurements.

Using the Student's *t*-test to measure Dandelion quantification, a *p*-value of 0.922 was obtained, meaning there were no significant differences ($p > 0.05$). The coefficient $r = 0.773$ indicates a high linear correlation between the two measurements.

Applying to the Kikuyu class, a *p*-value of 0.968 was reached, indicating a high concordance ($p > 0.05$) between both measurements. Furthermore, according to Pearson's coefficient $r = 0.989$, it is demonstrated that there is a very high linear association between both measurements.

Regarding the Broadleaf dock class, a *p*-value of 0.606 was reached, showing no significant differences ($p > 0.05$). According to the coefficient $r = 0.977$, there is a very strong linear association between both measurements.

Finally, applying to the "other weeds" class, a $p = 0.015$ was obtained. This result indicates a significant difference ($p < 0.05$) between the ground truth quantification and the model's predictions. The Pearson coefficient $r = 0.583$ indicates a moderate linear association between both measurements.

According to Student's *t*-tests, there is a high level of agreement between the model's prediction and the ground truth in most categories: Potato, Kikuyu, Dandelion, and Broadleaf dock. This level of agreement indicates the model's good generalization capacity to locate previously unseen plants (test set), a fundamental characteristic for its implementation in real-life agricultural environments. However, in the "other weeds" class, there is evidence of disagreement between the predictions and the ground truth, according to the *t*-Student's *t*-test ($p < 0.05$) and also a low precision (AP = 0.526 in Table 6). This could be due to several factors related to a limited class size on the dataset and unusual characteristics of the plants (size, shape, density), variability in light conditions, capture angles, presence of shadows, overlapping plants, abrupt changes in the texture, presence of species not adequately represented in the training data, noise in the images (mislabeled, blurry examples) and foreign objects (stones, sticks, branches). Figure 6 illustrates some incorrect predictions (second row) made by the RT-DETR model (variant 5), where the unusual conditions mentioned are evident.

b) Statistical validation of YOLOv11

When performing the Student's *t*-test for the Potato class, a *p*-value of 0.747 was obtained. This indicates no significant differences ($p > 0.05$) between the quantification of this plant and the ground truth, suggesting that the model does not differ systematically from the manual count. The Pearson correlation coefficient $r = 0.966$ indicates a very strong linear association between both measurements.

Using the Student's *t*-test to measure Dandelion quantification, a *p*-value of 0.832 was obtained, meaning there were no significant differences ($p > 0.05$). The coefficient $r = 0.813$ indicates a high linear correlation between the two measurements.

Applying to the Kikuyu class, a *p*-value of 0.808 was reached, indicating a high concordance ($p > 0.05$) between both measurements. Furthermore, according to Pearson's coefficient $r = 0.975$, it is demonstrated that there is a very high linear association between both measurements.

Regarding the Broadleaf dock class, a *p*-value of 0.925 was reached, showing no significant differences ($p > 0.05$). According to the coefficient $r = 0.978$, there is a very strong linear association between both measurements.

Finally, applying to the "other weeds" class, a $p = 0.277$ was obtained. This result indicates that there is also no significant difference ($p > 0.05$) between the ground truth quantification and the model's predictions. The Pearson coefficient $r = 0.610$ indicates a moderate linear association between both measurements.

Figure 6 illustrates some incorrect predictions (third row) made by the YOLOv11 model (variant 4) where some unusual conditions are evident.

Once the Student's *t*-tests were completed, it was determined that there was generally high agreement between the model's prediction and the ground truth in all categories: Potato, Kikuyu, Dandelion, Broadleaf dock, and "other weeds". This level of agreement indicates the model's excellent generalization ability in locating previously unseen plants (test set), which is crucial in practical applications.

The discrepancy between the average precision (AP) metric and the Student's *t*-test, based on the number of plants, may be common in object detection problems, particularly in certain weed classes. In the specific case where a low AP is obtained for the other weeds class (AP = 0.564 in Table 8), but contradictorily the Student's *t*-test indicates that there are no significant differences ($p > 0.05$) between the predicted number of plants and the ground truth, the different nature of both metrics may explain it.

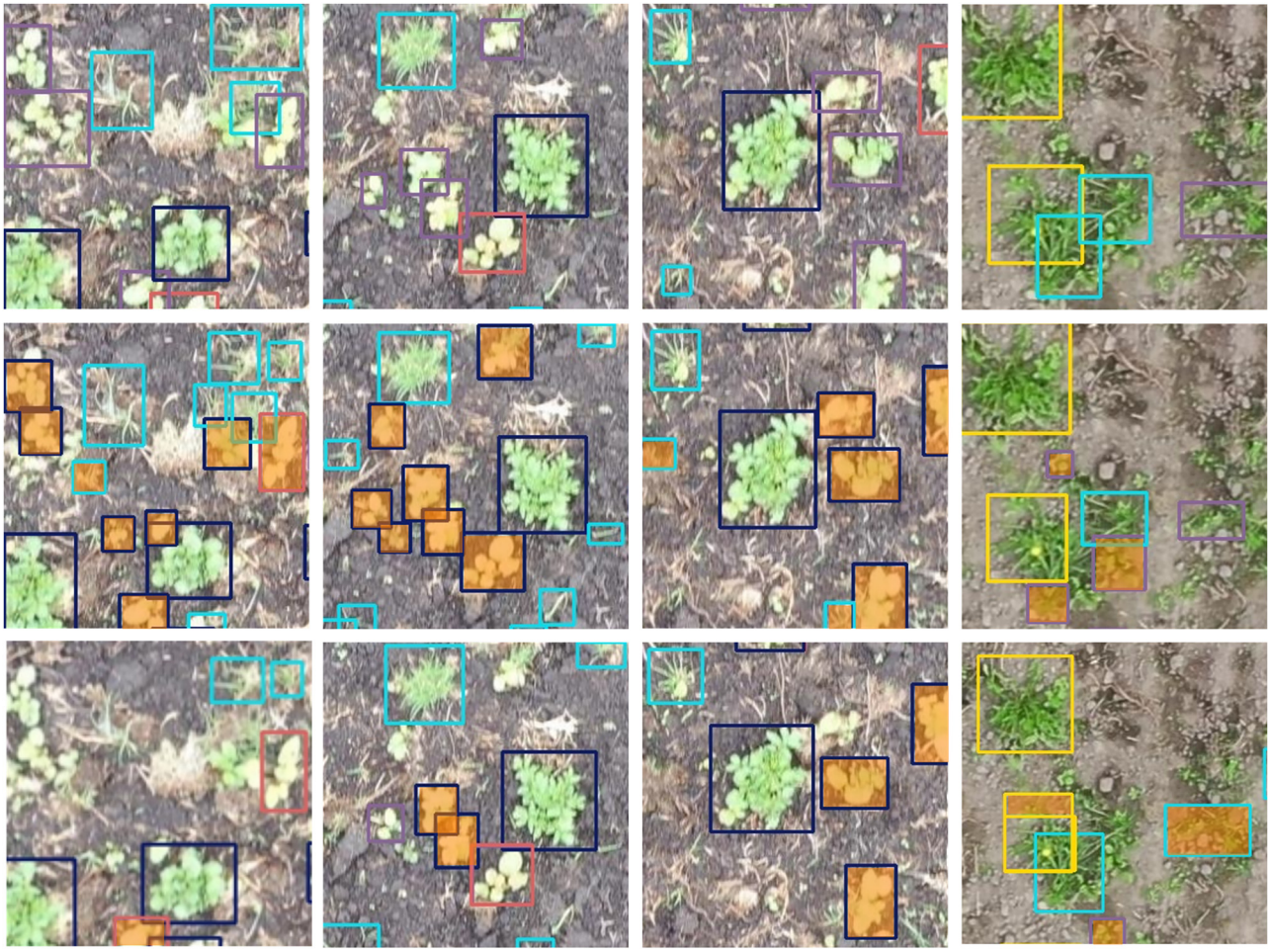


**Figure 6:** Examples of images with the highest level of discrepancy extracted from the ground truth (first row), the RT-DETR model predictions (second row), and the YOLOv11 model predictions (third row). Potatoes are marked by blue, Dandelion by yellow, Kikuyu by light blue, Broadleaf dock by red, and other weeds by purple. All incorrect predictions are filled with an orange box.

On the one hand, AP is a performance metric that evaluates how many predictions are correct according to an IoU (Intersection over Union) threshold, typically $\geq$0.5. IoU is a local metric that measures the precise overlap between predicted boxes and ground truth boxes, penalizing shape, position, or edge errors. Therefore, it is especially useful in scenarios where weeds' shape and precise location are important, such as in spot treatment applications.

In contrast, the Student's *t*-test analyzes differences in a global metric such as the total number of plants detected. This test tends to smooth out the effects of local errors, provided the total number or area of detections is similar to the ground truth. Therefore, it is more appropriate in contexts where the objective is to estimate the total amount of weeds, without requiring an exact location.

Furthermore, weed edges are problematic for detection models due to the transition between the background and other vegetation types. Small errors at the edges can significantly degrade the IoU and the AP, even though their impact on the total object count is minimal.

Thus, the discrepancies between AP and Student's *t*-test reflect the difference between metrics sensitive to local errors (IoU/AP) and those that capture global trends (Student's *t*-test). Combining both evaluation criteria offers a comprehensive approach to validating our models in practical applications.

On the other hand, regarding the processing time of the models, RT-DETR (variant 5) achieved an inference time of 57.5 ms (17.4 fps) in the Kaggle environment with an NVIDIA Tesla P100 GPU (16 GB of VRAM) on 1,020 $\times$ 1,020 images; while on a laptop with an NVIDIA GeForce RTX 3050 Ti

GPU (4 GB of VRAM) it reached 76.6 ms (13 fps). As for the YOLOv11m model (variant 4), it took an inference time of 37.77 ms (26.5 fps) in the Kaggle environment on 960 × 960 images; while on the laptop, it reached 70.7 ms (14.1 fps). A demo of each RT-DETR and YOLOv11 network, running in near real-time, can be seen at the following link (click here).

# 4 Discussion

In this work, modern state-of-the-art architectures of RT-DETR-1 (Transformer) [67] and YOLOv11 (CNN) [20] are used for weed detection in potato crops in near real time (1,020 × 1,020 and 960 × 960 UAV images, respectively) in uncontrolled environments. Five variants of each model are trained on a custom image dataset, optimizing the different hyperparameters, and applying image scaling and class balancing. The models have been trained on a medium-sized and diverse data set, allowing them to learn robust and discriminative features. The models are validated using robust performance metrics (mAP@50, mAP@50:95, precision, recall), and are statistically complemented with the Student's *t*-test (based on plant count) and Pearson correlation coefficient. The relevance of this study lies not in the novelty of the architectures but in their validation under realistic agricultural conditions and the integration of complementary evaluation strategies.

It is important to note that, although quantitative metrics are used to evaluate model performance in this study, direct comparisons with results reported in previous works should be interpreted with caution. Differences in datasets – such as crop type, weed species, number of classes, image acquisition methods (UAV vs. ground-based systems), spatial resolution, and environmental conditions – introduce variability that can significantly influence performance outcomes. Moreover, variations in evaluation protocols, including IoU thresholds and reported metrics, further limit strict comparability. Therefore, while quantitative results are presented and discussed, comparisons with related studies are intended to provide contextual and indicative insights rather than definitive performance rankings.

## 4.1 RT-DETR architecture

According to Table 6, the best RT-DETR model (variant 5) achieves an mAP of 0.809 and mAP@50:95 of 0.448 on the validation set. This represents a 9.3 % improvement in the mAP metric compared to the baseline model (variant 1), indicating that the various tuning strategies (Table 3) applied during model training play a crucial role in enhancing performance.

Table 9 compares the performance of RT-DETR (variant 5) against different current models based on Transformer architectures for weed/crop detection using UAV images (for a fair comparison) and the mAP evaluation metric.

Considering the models' accuracy, the RT-DETR by Arriola-Valverde et al. [39] achieved the best performance with 0.947 mAP, followed by our RT-DETR v5 model, which achieved an mAP of 0.801, outperforming the other models, including the RMS-DETR by Guo et al. [38], DETR by Li & Shi [35], and DETR by Borgogno et al. [36], which achieved 0.792, 0.779, and 0.513 mAP, respectively. Valicharla et al. [37] unfortunately do not report the obtained mAP.

Considering the number of classes, the RT-DETR model by Arriola-Valverde et al. [39] detects only one class corresponding to coffee plants. In contrast, our work considers five different classes, detecting the crop (potato) and four classes of weeds, which is a much more complex task for learning essential patterns and characteristics of the different plants. Regarding model size and inference times, both RT-DETR-based architectures are similar. In Guo et al. [38], the results of the RMS-DETR model (mAP = 0.792) are slightly lower than our RT-DETR (mAP = 0.801). The RMS-DETR model considers three classes, while in our work, it considers five. The DETR model by Li & Shi [35] also obtains acceptable results with 0.779 mAP, but is limited to detecting a single class. The DETR model by Borgogno et al. [36], achieves the worst performance with mAP = 0.513, which is somewhat justified by the complexity of detecting

**Table 9:** Comparison of current transformer-based models for crop and weed detection using UAV images.

| Author | Year | Model | Number of classes | mAP |
|---|---|---|---|---|
| Li & Shi [35] | 2024 | DETR | 1 class: Brassica chinensis | 0.779 |
| Borgogno et al. [36] | 2024 | DETR | 12 classes: maize, tomato and 10 species of weeds | 0.513 |
| Valicharla et al. [37] | 2024 | Swin-transformer | 3 classes: giant and Japanese knotweeds in different phenological stages | – |
| Guo et al. [38] | 2024 | RMS-DETR | 3 classes: field rige, continuous patches of barnyard grass, and single barnyard grass | 0.792 |
| Arriola-Valverde et al. [39] | 2024 | RT-DETR | 1 class: coffee plant | **0.947** |
| Our proposal | 2025 | RT-DETR | 5 classes: potato, Kikuyo, Dandelion, Broadleaf dock, and other weeds | 0.801 |

The best value is in bold.

12 plant categories, including 10 weed types. This shows that, in general, the lower the number of classes detected, the higher the accuracy obtained by the models.

On the other hand, dataset balancing (v2 vs v1) in our RT-DETR model improves model performance by 5.6 % mAP, as observed in Table 5 (variants 2 and 5). Also, increasing image resolution (640, 800, 960, 1,020) positively impacts model performance. However, beyond 960 pixels, the improvement is minimal between variants 4 (960 × 960) and 5 (1,020 × 1,020).

Our study presents an RT-DETR-1 (variant 5) model based on the Transformer architecture, achieving an mAP of 0.801, which is comparable to or superior to other algorithms. This model was trained on a custom dataset with a larger number of classes (5), representing a significant contribution to the field of study.

## 4.2 YOLOv11 architecture

According to Table 8, the best YOLOv11m model (variant 4) achieves an mAP of 0.825 and mAP@50:95 of 0.457 on the validation set. This represents a 2.9 % improvement in mAP compared to the baseline model (variant 1), which means that the various tuning strategies (Table 4) applied during model training are important in improving performance.

Table 10 compares the performance of YOLOv11m (variant 4) against different current YOLO-based models in the literature for weed/crop detection using only UAV images (for a fair comparison) and the mAP evaluation metric.

According to Table 10, it is observed that most current studies address the detection of a single crop/weed class, a much simpler problem than a multi-class task, which turns out to be much more complex in uncontrolled environments. YOLO models trained for single-class detection [39], [45]–[47] (except for the improved YOLOv5 [42]) report the highest scores with mAP $>$ 0.90, with YOLOv5-T [45] being the best with 0.987 mAP. This confirms that the lower the number of classes, the higher the accuracy obtained. Although, our YOLOv11m (5 classes) with mAP = 0.809 outperforms the modified YOLOv5 model [42] with a single class, mAP = 0.739.

Regarding multiclass detection models, only in [44], 3 classes (bean, soybean, and weeds) are considered using YOLOv8s, and they obtain an mAP = 0.970, which outperforms our work with mAP = 0.809, considering five classes. The YOLOv8s model is lighter (with 11.2 M parameters) and is trained on 640 × 640 images, unlike the YOLOv11m variant 4 model (with 20.1 M parameters), which uses 960 × 960 images. However, such work with YOLOv8s only detects a single generic weed class, unlike our work, which identifies four weed categories (Kikuyu, Dandelion, Broadleaf dock, other weeds), offering greater applicability in real agricultural scenarios with multiple weed species, suitable for site-specific treatments in precision agriculture. As mentioned, weed detection seemed to be more challenging than crop detection, mainly due to the detection of smaller weeds [33].

Additionally, we compare our results with those obtained in the recent work by García-Navarrete et al. [40] using the same two algorithms (YOLOv11 and RT-DETR-1) to detect five classes (the same number), although they experimented on corn plants and did not use AUV images. These authors obtain an mAP of 0.824 for YOLOv11s and 0.828 for RT-DETR-1, while our study obtains mAPs of 0.809 and 0.801, respectively. However, they use images acquired with a support trolley at 1 m height inside a climate-controlled greenhouse, while we use UAV images taken at 9–10 m in uncontrolled environments. Despite these important configuration differences, the metrics in both studies are similar, within the same order of magnitude. This demonstrates that our proposal is suitable for challenging agricultural environments.

Comparing both modern architectures tested in this study for weed detection using our custom dataset, YOLOv11 (mAP = 0.809) obtains slightly better accuracy than RT-DETR (mAP = 0.801) by about 1 %. This result (mAP) partially contradicts [39], where a better performance is achieved with RT-DETR compared to YOLOv9, suggesting that the recent

**Table 10:** Comparison of current YOLO-based models (CNN) for crop and weed detection using UAV images.

| Author | Year | Model | Number of classes | mAP |
|---|---|---|---|---|
| Rehman et al. [42] | 2024 | Improved YOLOv5 | 1 class: weeds in soybean fields | 0.739 |
| Silva et al. [44] | 2024 | YOLOv8s | 3 classes: bean, soybean, and weeds | 0.970 |
| Gao et al. [45] | 2024 | YOLOv5-T | 1 class: maize tassels | **0.987** |
| Arsalan et al. [46] | 2024 | YOLOv6s | 1 class: tobacco | 0.979 |
| Tian et al. [47] | 2024 | YOLOv7 | 1 class: cotton | 0.969 |
| Arriola-Valverde et al. [39] | 2024 | YOLOv9 | 1 class: coffee | 0.911 |
| Our proposal | 2025 | YOLOv11m | 5 classes: potato, Kikuyo, Dandelion, Broadleaf dock and other weeds | 0.809 |

The best value is in bold.

version YOLOv11 is more accurate than its predecessor. This could be attributed to several key factors. YOLOv11's design is specifically tailored to handle objects of varying sizes and conditions, which can be advantageous in agricultural settings with weeds of diverse characteristics and appearances, in addition to its ability to handle smaller datasets. On the other hand, RT-DETR, although powerful, may require larger and more varied datasets to achieve optimal performance and effectively capture spatial and contextual relationships.

Overall, both YOLOv11 and RT-DETR models present adequate performances comparable to those found in the existing literature, but in our case, considering a larger set of categories (five). However, the "other weeds" class still penalizes the model's overall performance. Furthermore, YOLOv11 is faster (26.5 fps) than RT-DETR (17.4 fps) processing a 960 × 960 and 1,020 × 1,020 image, respectively, on NVIDIA TESLA P100 GPU (16 GB), both suitable for near-real-time applications. This comparison of results (accuracy and speed) in weed detection using Transformer- and CNN-based architectures is novel and very useful for the field of study in uncontrolled agricultural environments.

From a practical deployment perspective, the reported inference speed and model efficiency suggest that both RT-DETR and YOLOv11 architectures are suitable for near real-time applications in precision agriculture. In UAV-based monitoring scenarios, inference times in the order of tens of milliseconds per image can enable on-the-fly weed detection, supporting decision-making during flight operations. In the context of precision spraying systems, real-time or near real-time detection is critical to activate spraying mechanisms only when weeds are detected, reducing herbicide usage and environmental impact. The balance between accuracy and inference speed observed in this study indicates that these models could be integrated into embedded systems, such as onboard processors in drones or edge computing devices. However, deployment in real-world settings would require further optimization, including model compression, hardware-specific acceleration, and robustness testing across varying environments. These aspects represent an important direction for upcoming work.

Finally, regarding this study's limitations, the dataset is restricted to five categories. The variability in morphology, size, and spatial distribution within the "other weeds" category introduces additional complexity, particularly under UAV-based imaging conditions. Future work should consider a more fine-grained annotation strategy to separate these species into more homogeneous classes, potentially improving both detection accuracy and model explainability. Additionally, the dataset is limited to a specific stage of crop growth. Another restriction is the exclusive use of the RT-DETR-1 and YOLOv11 architectures without personalization or manipulation of their internal components, aiming to better adjust to the silhouettes and sizes of the plants to improve model performance. Finally, our work uses only a low-cost UAV to capture RGB images, mainly because they are popular and affordable, offering a cost-efficient solution [8], [73].

# 5 Conclusions and future work

In this work, modern RT-DETR-1 and YOLOv11 models are successfully trained for the task of detecting several types of native weeds (Ecuador) in potato crops in near real time: Dandelion, Kikuyu, Broadleaf dock, and other weeds. Model training comprises hyperparameter optimization, image scaling, and class balancing on a diverse custom dataset. The results of the Transformers-based RT-DETR-1 model (mAP = 0.801) and the CNN-based YOLOv11 model (mAP = 0.809) on the test set are comparable or superior to those found in existing literature, but in our case, considering a larger set of categories (five). RT-DETR reaches a speed of 17.4 fps processing a 1,020 × 1,020 image on GPU NVIDIA TESLA P100 (16 GB VRAM), whereas YOLOv11 reaches 26.5 fps processing a 960 × 960 image, both suitable for near real-time applications in an agricultural context. Overall, YOLOv11 is slightly more accurate (about 1 % in terms of mAP) and faster than RT-DETR on our custom dataset.

For future work, it is recommended to expand the number of weed categories to minimize the model performance penalty due to the "other weeds" class of the dataset. It is also advisable to propose modifications to the internal architecture of RT-DETR-1 or YOLOv11 to adapt it to real-time weed detection better. Following this research line, weed mapping is necessary for creating georeferenced maps that indicate the spatial distribution and density of weeds within a field.

**Acknowledgments:** This research is supported through the Software Engineering and Artificial Intelligence research group from the Universidad Técnica del Norte (Ecuador) and the GTI-IA research group from Universitat Politècnica de València (Spain). Additionally, this work has been partially supported by Grant PID2021-123673OB-C31 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

**Research ethics:** Not applicable.

**Informed consent:** Not applicable.

**Author contribution:** Lucía Sandoval-Pillajo: Writing – original draft, Visualization, Investigation, Conceptualization. Luis Perugachi-Chávez: Software, Data Curation, Formal Analysis, Writing – Original Draft. Marco Pusdá-Chulde: Writing – review & editing. Adriana Giret-Boggino: Writing – review & editing. Iván García-Santillán: Writing – review & editing, Validation, Supervision, Conceptualization. All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

**Use of Large Language Models, AI and Machine Learning Tools:** We used Grammarly software to improve language in the manuscript.

**Conflict of interest:** The authors state no conflict of interest.

**Funding information:** This research was mostly funded by the Universidad Técnica del Norte, Ecuador, under Grant Number: InvestigaUTN-2024-1353.

**Data availability statement:** Datasets used in this study are publicly available for academic and comparative purposes and can be found at the following link (last accessed on 08 April 2025): https://www.kaggle.com/datasets/f50d-53884d81757caff683a23399bc80a2f7da95937ad15efe00826aa-b46f4f5.

# References

[1] United Nations, "The sustainable development goals report 2023: Special edition," [Online]. Available: https://unstats.un.org/sdgs/report/2023/ [Accessed: Dec. 20, 2024].

[2] A. Babaei-Ghaghelestany, M. T. Alebrahim, S. Farzaneh, and M. Mehrabi, "The anticancer and antibacterial properties of aqueous and methanol extracts of weeds," *J. Agric. Food Res.*, vol. 10, p. 100433, 2022.

[3] A. Paušič, S. Tojnko, and M. Lešnik, "Permanent, undisturbed, in-row living mulch: A realistic option to replace glyphosate-dominated chemical weed control in intensive pear orchards," *Agric. Ecosyst. Environ.*, vol. 318, p. 107502, 2021.

[4] M. Vasileiou *et al*., "Transforming weed management in sustainable agriculture with artificial intelligence: A systematic literature review towards weed identification and deep learning," *Crop Prot.*, vol. 176, p. 106522, 2024.

[5] J. Gao *et al*., "Cross-domain transfer learning for weed segmentation and mapping in precision farming using ground and UAV images," *Expert Syst. Appl.*, vol. 246, p. 122980, 2024.

[6] I. D. García-Santillán and G. Pajares, "On-line crop/weed discrimination through the Mahalanobis distance from images in maize fields," *Biosyst. Eng.*, vol. 166, pp. 28—43, 2018.

[7] K. Osorio, A. Puerto, C. Pedraza, D. Jamaica, and L. Rodríguez, "A deep learning approach for weed detection in lettuce crops using multispectral images," *AgriEngineering*, vol. 2, no. 3, pp. 471—488, 2020.

[8] K. Zou, X. Chen, F. Zhang, H. Zhou, and C. Zhang, "A field weed density evaluation method based on UAV imaging and modified U-net," *Remote Sens.*, vol. 13, no. 2, p. 310, 2021.

[9] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70—90, 2018.

[10] C.-P. Anderson, P.-C. Marco, T.-E. Diego, C.-S. Victor, and G.-S. Iván, "Binary classification of defects in multiple coffee beans using lightweight convolutional neural networks for embedded systems," *Data Metadata*, vol. 4, p. 840, 2025.

[11] M. Cevallos, L. Sandoval-Pillajo, V. Caranqui-Sánchez, C. Ortega-Bustamante, M. Pusdá-Chulde, and I. García-Santillán, "Morphological defects classification in coffee beans based on convolutional neural networks," *Commun. Comput. Inf. Sci.*, vol. 2276, pp. 3—15, 2025.

[12] F. Ulloa, L. Sandoval-Pillajo, P. Landeta-López, N. Granda-Peñafiel, M. Pusdá-Chulde, and I. García-Santillán, "Identification of diabetic retinopathy from retinography images using a convolutional neural network," *Commun. Comput. Inf. Sci.*, vol. 2276, pp. 121—136, 2025.

[13] S. Montenegro, M. Pusdá-Chulde, V. Caranqui-Sánchez, J. Herrera-Tapia, C. Ortega-Bustamante, and I. García-Santillán, "Android mobile application for cattle body condition score using convolutional neural networks," *Commun. Comput. Inf. Sci.*, vol. 1705, pp. 91—105, 2023.

[14] F. Salazar-Fierro, C. Cumbal, D. Trejo-España, C. León-Fernández, M. Pusdá-Chulde, and I. García-Santillán, "Detection of scoliosis in X-ray images using a convolutional neural network," *Commun. Comput. Inf. Sci.*, vol. 2276, pp. 167—183, 2025.

[15] B. Chacua *et al*., "People identification through facial recognition using deep learning," in *2019 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2019*, 2019.

[16] B. Espejo-Garcia, N. Mylonas, L. Athanasakos, S. Fountas, and I. Vasilakoglou, "Towards weeds identification assistance through transfer learning," *Comput. Electron. Agric.*, vol. 171, p. 105306, 2020.

[17] M. R. Pusdá-Chulde, F. A. Salazar-Fierro, L. Sandoval-Pillajo, E. P. Herrera-Granda, I. D. García-Santillán, and A. De Giusti, "Image analysis based on heterogeneous architectures for precision agriculture: A systematic literature review," *Adv. Intell. Syst. Comput.*, vol. 1078, pp. 51—70, 2020.

[18] K. Vinueza, L. Sandoval-Pillajo, A. Giret-Boggino, D. Trejo-España, M. Pusdá-Chulde, and I. García-Santillán, "Automatic weed quantification in potato crops based on a modified convolutional neural network using drone images," *Data Metadata*, vol. 4, no. 194, 2025, https://doi.org/10.56294/dm2025194.

[19] L. Sandoval-Pillajo, I. García-Santillán, M. Pusdá-Chulde, and A. Giret, "Weed detection based on deep learning from UAV imagery: A review," *Smart Agric. Technol.*, vol. 12, 2025. https://doi.org/10.1016/j.atech.2025.101147.

[20] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," 2024. [Online]. Available:https://arxiv.org/abs/2410.17725.

[21] A. Vaswani *et al*., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017, pp. 5999—6009, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762v7.

[22] Z. Liu *et al*., "Swin transformer: Hierarchical vision transformer using shifted windows," 2021. [Online]. https://github [Accessed: Dec. 22, 2024].

[23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers,"

in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12346, Cham, Springer, 2020, pp. 213—229.
[24] M. Yang, R. Xu, C. Yang, H. Wu, and A. Wang, "Hybrid-DETR: A differentiated module-based model for object detection in remote sensing images," *Electronics*, vol. 13, no. 24, p. 5014, 2024.
[25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *ICLR 2021 — 9th International Conference on Learning Representations*, 2020. https://arxiv.org/abs/2010.04159v4 [Accessed: Dec. 22, 2024].
[26] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query DeNoising," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2239—2251, 2022.
[27] H. Zhang *et al*., "DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection," in *11th International Conference on Learning Representations, ICLR 2023*, 2022 [Online]. Available: https://arxiv.org/abs/2203.03605v4 [Accessed: Dec. 23, 2024].
[28] Y. Zhao *et al*., "DETRs beat YOLOs on real-time object detection," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16965—16974.
[29] M. Darbyshire, S. Coutts, P. Bosilj, E. Sklar, and S. Parsons, "Review of weed recognition: A global agriculture perspective," *Comput. Electron. Agric.*, vol. 227, p. 109499, 2024.
[30] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, no. 3, p. 37, 1996.
[31] B. Deng, Y. Lu, and J. Xu, "Weed database development: An updated survey of public weed datasets and cross-season weed detection adaptation," *Ecol. Inform.*, vol. 81, p. 102546, 2024.
[32] X. Kong, T. Liu, X. Chen, X. Jin, A. Li, and J. Yu, "Efficient crop segmentation net and novel weed detection method," *Eur. J. Agron.*, vol. 161, p. 127367, 2024.
[33] D. A. Abuhani, M. H. Hussain, J. Khan, M. Elmohandes, and I. Zualkernan, "Crop and weed detection in sunflower and sugarbeet fields using single shot detectors," in *2023 IEEE International Conference on Omni-Layer Intelligent Systems, COINS 2023*, 2023.
[34] N. Iqbal, C. Manss, C. Scholz, D. Konig, M. Igelbrink, and A. Ruckelshausen, "AI-based maize and weeds detection on the edge with CornWeed dataset," in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems, FedCSIS 2023*, 2023, pp. 577—584.
[35] H. Li and F. Shi, "A DETR-like detector-based semi-supervised object detection method for Brassica Chinensis growth monitoring," *Comput. Electron. Agric.*, vol. 219, p. 108788, 2024.
[36] E. Borgogno, J. M. Peña, A. I. de Castro, I. Borra-Serrano, and J. Dorado, "Cognitive computing advancements: Improving precision crop protection through UAV imagery for targeted weed monitoring," *Remote Sens.*, vol. 16, no. 16, p. 3026, 2024.
[37] S. K. Valicharla, R. Karimzadeh, K. Naharki, X. Li, and Y. L. Park, "Detection and multi-class classification of invasive knotweeds with drones and deep learning models," *Drones*, vol. 8, no. 7, p. 293, 2024.
[38] Z. Guo, D. Cai, Y. Zhou, T. Xu, and F. Yu, "Identifying rice field weeds from unmanned aerial vehicle remote sensing imagery using deep learning," *Plant Methods*, vol. 20, no. 1, pp. 1—17, 2024.
[39] S. Arriola-Valverde, R. Rimolo-Donadio, K. Villagra-Mendoza, A. Chacón-Rodriguez, R. García-Ramirez, and E. Somarriba-Chavez, "A comparative study of deep learning frameworks applied to coffee plant detection from close-range UAS-RGB imagery in Costa Rica," *Remote Sens.*, vol. 16, no. 24, p. 4617, 2024.
[40] O. L. García-Navarrete, J. H. Camacho-Tamayo, A. B. Bregon, J. Martín-García, and L. M. Navas-Gracia, "Performance analysis of real-time detection transformer and you only look once models for weed detection in maize cultivation," *Agronomy*, vol. 15, no. 4, 2025, https://doi.org/10.3390/agronomy15040796.
[41] S. Pavithra, D. Kachroo, V. Kadam, H. Padala, and R. Purbey, "Drone-based weed and disease detection in agricultural fields to maximize crop health using a Yolov8 approach," in *2023 IEEE 7th Conference on Information and Communication Technology, CICT 2023*, 2023.
[42] M. U. Rehman, H. Eesaar, Z. Abbas, L. Seneviratne, I. Hussain, and K. T. Chong, "Advanced drone-based weed detection using feature-enriched deep learning approach," *Knowl. Based Syst.*, vol. 305, 2024, https://doi.org/10.1016/J.KNOSYS.2024.112655.
[43] P. Saini and D. S. Nagesh, "CottonWeeds: Empowering precision weed management through deep learning and comprehensive dataset," *Crop Prot.*, vol. 181, p. 106675, 2024.
[44] J. A. O. S. Silva *et al*., "Deep learning for weed detection and segmentation in agricultural crops using images captured by an unmanned aerial vehicle," *Remote Sens.*, vol. 16, no. 23, p. 4394, 2024.
[45] R. Gao, Y. Jin, X. Tian, Z. Ma, S. Liu, and Z. Su, "YOLOv5-T: A precise real-time detection method for maize tassels based on UAV low altitude remote sensing images," *Comput. Electron. Agric.*, vol. 221, p. 108991, 2024.
[46] M. Arsalan *et al*., "Real-time precision spraying application for tobacco plants," *Smart Agric. Technol.*, vol. 8, p. 100497, 2024.
[47] F. Tian, C. J. Ransom, J. Zhou, B. Wilson, and K. A. Sudduth, "Assessing the impact of soil and field conditions on cotton crop emergence using UAV-based imagery," *Comput. Electron. Agric.*, vol. 218, p. 108738, 2024.
[48] M. Danilevicz, R. L. Rocha, J. Batley, P. E. Bayer, M. Bennamoun, and D. Edwards, "Segmentation of sandplain lupin weeds from morphologically similar narrow-leafed lupins in the field," 2023 [Online]. Available: https://figshare.com/articles/dataset/Segmentation_of_sandplain_lupin_weeds_from_morphologically_similar_narrow-leafed_lupins_in_the_field/21746669 [Accessed: Feb. 20, 2025].
[49] I. Moazzam, "Tobacco aerial dataset," *Mendeley Data*, vol. 2, 2023. https://doi.org/10.17632/5DPC5GBGPZ.2.
[50] I. Sa *et al*., "WeedMap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming," *Remote Sens.*, vol. 10, no. 9, p. 1423, 2018.
[51] I. Koshelev, M. Savinov, A. Menshchikov, and A. Somov, "Drone-aided detection of weeds: Transfer learning for embedded image processing," *IJSTA*, vol. 16, pp. 102—111, 2023.
[52] N. Genze, R. Ajekwe, Z. Güreli, F. Haselbeck, M. Grieb, and D. G. Grimm, "Deep learning-based early weed segmentation using motion blurred UAV images of sorghum fields," *Comput. Electron. Agric.*, vol. 202, p. 107388, 2022.
[53] H. Huang, J. Deng, Y. Lan, A. Yang, X. Deng, and L. Zhang, "A fully convolutional network for weed mapping of unmanned aerial

vehicle (UAV) imagery," *PLoS One*, vol. 13, no. 4, p. e0196302, 2018.

[54] M. Krestenitis *et al*., "CoFly-WeedDB: A UAV image dataset for weed detection and species identification". [Data set]. *Zenodo*. https://doi.org/10.5281/ZENODO.6697343.

[55] I. Sa *et al*., "WeedNet: Dense semantic weed classification using multispectral images and MAV for smart farming," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 588—595, 2018.

[56] N. Rai *et al*. "ImageWeeds: An image dataset consisting of weeds in multiple formats to advance computer vision algorithms for real-time weed identification and spot spraying application," *Mendeley Data*, vol. 2, 2023. https://doi.org/10.17632/8KJCZTBJZ2.2.

[57] J. Valente and L. Kooistra, "Dataset on UAV high-resolution images from grassland with broad-leaved dock (Rumex Obtusifolius)". [Data set]. *Zenodo*. https://doi.org/10.5281/ZENODO.5119205.

[58] A. dos Santos Ferreira, D. Matte Freitas, G. Gonçalves da Silva, H. Pistori, and M. Theophilo Folhes, "Weed detection in soybean crops using ConvNets," *Comput. Electron. Agric.*, vol. 143, pp. 314—324, 2017.

[59] M.-D. Yang, H.-H. Tseng, Y.-C. Hsu, C.-Y. Yang, M.-H. Lai, and D.-H. Wu, "A UAV open dataset of rice paddies for deep learning practice," *Remote Sens. (Basel)*, vol. 13, no. 7, 2021, https://doi.org/10.3390/rs13071358.

[60] I. L. Bretas *et al*., "Detection and mapping of *Amaranthus spinosus* L. in bermudagrass pastures using drone imagery and deep learning for a site-specific weed management," *Agron. J.*, vol. 116, no. 3, pp. 990—1002, 2024.

[61] J. Anderegg *et al*., "On-farm evaluation of UAV-based aerial imagery for season-long weed monitoring under contrasting management and pedoclimatic conditions in wheat," *Comput. Electron. Agric.*, vol. 204, p. 107558, 2023.

[62] Q. Wang *et al*., "An image segmentation method based on deep learning for damage assessment of the invasive weed Solanum rostratum Dunal," *Comput. Electron. Agric.*, vol. 188, p. 106320, 2021.

[63] M. V. Ornhag, P. Persson, M. Wadenback, K. Astrom, and A. Heyden, "Efficient real-time radial distortion correction for UAVs," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1750—1759.

[64] Roboflow Docs, "Create a project | Roboflow Docs 2024," [Online]. Available: https://docs.roboflow.com/datasets/create-a-project [Accessed: Dec. 27, 2024].

[65] J. Solawetz, "Train, validation, test split for machine learning," [Online]. Available: https://blog.roboflow.com/train-test-split/ [Accessed: Dec. 27, 2024].

[66] Ultralytics, "Model training with Ultralytics YOLO," [Online]. Available: https://docs.ultralytics.com/modes/train/#train-settings [Accessed: Mar. 27, 2025].

[67] Ultralytics, "RT-DETR (real-time detection transformer) — Ultralytics docs," [Online]. Available: https://docs.ultralytics.com/models/rtdetr/#pre-trained-models [Accessed: Mar. 19, 2025].

[68] Kaggle, "How to use Kaggle," [Online]. Available: https://www.kaggle.com/docs [Accessed: Apr. 29, 2024].

[69] Ultralytics, "Ultralytics YOLOv11: Code repository," 2024. [Online]. Available at: https://github.com/ultralytics/ultralytics.

[70] R. Padilla, S. L. Netto, and E. A. B. Da Silva, "A survey on performance metrics for object-detection algorithms," in *International Conference on Systems, Signals, and Image Processing*, IEEE, 2020, pp. 237—242.

[71] A. Juma, J. Rodríguez, J. Caraguay, M. Naranjo, A. Quiña-Mera, and I. García-Santillán, "Integration and evaluation of social networks in virtual learning environments: A case study," *Commun. Comput. Inf. Sci.*, vol. 895, pp. 245—258, 2019.

[72] D. Lind, W. Marchal, and S. Wathen, *Basic Statistics in Business and Economics*, 10th ed. Columbus, OH, McGraw Hill, 2022.

[73] L. Sandoval-Pillajo, M. Pusdá-Chulde, J. Pazos-Morillo, P. Granda-Gudiño, and I. García-Santillán, "Multi-class weed quantification based on U-net convolutional neural networks using UAV imagery," *Appl. Sci.*, vol. 16, no. 7, p. 3149, 2026.