pagepress

YOLO deep learning algorithm for object detection in agriculture: a review

Kamalesh Kanna S,[1] Kumaraperumal R,[1] Pazhanivelan P,[2] Jagadeeswaran R,[1] Prabu P.C[1]

[1]Department of Remote Sensing and Geographic Information System, Tamil Nadu Agricultural University, Coimbatore
[2]Centre for Water and Geospatial Studies, Tamil Nadu Agricultural University, Coimbatore, India

Abstract

YOLO represents the one-stage object detection also called regression-based object detection. Object in the given input is directly classified and located instead of using the candidate region. The accuracy from two-stage detection is higher than one-stage detection where one-stage object detection speed is higher than two-stage object detection. YOLO has become popular because of its detection accuracy, good generalization, open-source, and speed. YOLO boasts exceptional speed due to its approach of using regression problems for frame detection, eliminating the need for a complex pipeline. In agriculture, using remote sensing and drone technologies YOLO classifies and detects crops, diseases, and pests, and is also used for land use mapping, environmental monitoring, urban planning, and wildlife. Recent research highlights YOLO's impressive performance in various agricultural applications. For instance, YOLOv4 demonstrated high accuracy in counting and locating small objects in UAV-captured images of bean plants, achieving an AP of 84.8% and a recall of 89%. Similarly, YOLOv5 showed significant precision in identifying rice leaf diseases, with a precision rate of 90%. In this review, we discuss the basic principles behind YOLO, different versions of YOLO, limitations, and YOLO application in agriculture and farming.

Correspondence: Kumaraperumal R, Department of Remote Sensing and Geographic Information System, Tamil Nadu Agricultural University, Coimbatore, India.
E-mail: kumaraperumal.r@tnau.ac.in

Key words: agriculture; computer vision; deep learning; object detection; real-time farming, YOLO.Contributions: all the authors contributed equally, read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

Conflict of interest: the authors declare that they have no competing interests, and all authors confirm accuracy.

Received: 14 March 2024.
Accepted: 13 July 2024.

Journal of Agricultural Engineering 2024; LV:1641
doi:10.4081/jae.2024.1641

Introduction

An increasing population results in an increasing demand for food, which results in the need for increased production of agricultural products. Even though there is an increase in production we cannot meet the need because of several factors like pest and disease attacks, improper harvesting, climate factors, biodiversity, *etc*. Using advanced technologies like drones, Artificial Intelligence (AI), and robots we can manage those factors. Thus, by introducing AI techniques we can improve the agriculture sector production and reduce crop loss by reducing pest and disease attacks, improving nutrient management, timely harvest, *etc*. Using deep learning, big data, and Internet of Things (IoT) we can monitor crops, predict yield, manage irrigation, manage weeds, detect plant stress, *etc*. In computer vision, the most challenging and fundamental task is object detection. Object detection involves accurately finding the object in the input image and classifying it according to the labels. In object detection, object classification, instance segmentation, and semantic segmentation are related (Xiao *et al*., 2020; Zhang and Cloutier, 2021). Due to the development of deep learning methods, object detection has improved from machine learning to deep learning methods which are based on analytics. In remote sensing object detection is more challenging due to the smaller number of datasets available, and low-resolution images (Teng *et al*., 2019; Yin *et al*., 2018, 2019). Object detection is classified into two types based on their working stage: Two-stage and single-stage object detection. Two-stage object detection is represented by R-CNN (Region-based Convolutional Neural Network) (Girshick, 2015). It involves object detection in two stages. First, the candidate region is generated in the images. Then, regression processing and object classification were performed on the candidate region (Papageorgiou *et al*., 1998; Zhou *et al*., 2018). In our review, YOLO represents the one-stage object detection also called regression-based object detection. Here the object in the given input is directly classified and located instead of using the candidate region. The accuracy from two-stage detection is higher than one-stage detection where one-stage object detection speed is higher than two-stage object detection (Dollár *et al*., 2014; Song *et al*., 2011). In real-time object detection, the YOLO (You Only Look Once) algorithm is remarkable for its accuracy and speed when compared to other algorithms such as DPM (Deformable Parts Model), OverFeat, SSD (Single Shot MultiBox Detector), RCNN (Regions with Convolutional Neural Network features), SPPNet (Spatial Pyramid Pooling Network), fast RCNN, Mask RCNN, etc., providing reliable results in a brief period. The main idea of GoogleNet (Zhong *et al*., 2015) has been implemented into the

YOLO algorithm of their networks. We can improve the Deep convolution network by implementing large-scale image datasets such as ImageNet and COCO. In agriculture YOLO classifies and detects crops (Espinoza-Hernández *et al*., 2023; Tian *et al*., 2019; Wu *et al*., 2020),weeds (Ajayi *et al*., 2023), diseases and pests (Lippi *et al*., 2021), land use mapping (Cheng *et al*., 2021), environmental monitoring (Zakria *et al*., 2022), urban planning (Qing *et al*., 2021), and wildlife (Roy *et al*., 2023).

YOLO (you only look once)

YOLO is a real-time object identification technique that was introduced in 2015 by Redmon and colleagues in their research paper "You only look once: unified, real-time object detection" (Redmon *et al*., 2016) (Figure 1). YOLO approached the object detection problem as a regression problem spatially. The YOLO method is a direct object detection technique that employs a solitary neural network to forecast several bounding boxes and the corresponding probability of each box's class. YOLO directly trains and enhances detection performance on full-size photos, so implicitly adding contextual knowledge about classes and their visual properties. Notably, Fast R-CNN, a leading detection method, tends to misinterpret background patches as objects due to its limited contextual awareness.

History of YOLO

YOLOv1 - you only look once version 1

YOLO spatially detects objects present on the images based on regression by creating associated class probabilities and separate bounding boxes. YOLO's speed is fast because it only requires the image to be input into the network to obtain the final detection result. This makes it possible for YOLO to perform real-time object detection on videos as well (Jiang *et al*., 2022). YOLO vaticinate class probabilities and bounding boxes by a single neural network in a single evaluation. By this, the optimization of the algorithm on object detection can increased directly (Redmon *et al*., 2016). By this real-time object detection has been achieved and we can detect objects even in videos having more fps. Fast RCNN an object detection algorithm, makes errors by identifying background patches as objects in an image but YOLO makes less than half an error when compared to the RCNN algorithm.

Because of YOLO's generalizability, it is more stable when applied to a new domain of interest or unexpected inputs, when YOLO was introduced with artworks trained with natural images, it outperformed other algorithms such as RCNN and DPM but the accuracy of YOLOv1 less. During test time YOLO is extremely fast because it requires only a single network evaluation. Non-maximum suppression (NMS) has been used here to reduce the multiple detection error. Source code: https://github.com/pjreddie/darknet

YOLO9000

YOLO9000 can detect over nine thousand object categories. A 2% improvement in mAP is achieved by adding batch normalization to all Convolutional layers in YOLO, by this we can remove dropouts from the model without overfitting. YOLO9000 has been adjusted to work better in higher resolution inputs, by this increase of 4% mAP is achieved. For predicting bounding boxes instead of using fully connected layers, YOLOv2 used anchor boxes. The network operation has been reduced to 416 images instead of 448 x 448 to achieve an odd number of locations in the feature map so that we have a single centre cell. In YOLOv2 they used multiscale training. For every 10 batches of iterations, different resolutions have been chosen by the algorithm itself. By this, the algorithm can detect objects at different resolutions. It is seen that, for lower resolution, the algorithm works fairly accurately by producing 69 mAP, and for higher resolution still operates above real-time speed producing 78.6 mAP (Redmon and Farhadi, 2017). Using PASCAL VOC 2012 dataset YOLOv2 runs faster than other algorithm methods achieving 73.4 mAP (Table 1). YOLOv2, we have been cooperatively using the detection dataset (detects bounding boxes, and objectness and classifies common objects) and classification dataset (expands the number of categories the algorithm can detect). Here we use the multi-label model to combine data that are not mutually exclusive. If an image is labelled for detection, our network can backpropagate based on the complete YOLOv2 loss function. When YOLOv2 encounters an image that requires classification, it only backpropagates the loss from the parts of the architecture that are specific to classification. Source code: https://pjreddie.com/darknet/yolov2/

YOLOv3

Redmon and Farhadi published YOLOv3 in ArXiv in 2018. Despite bigger architecture, they maintained the real-time performance of YOLOv3.YOLOv3 architecture is made up of 53 convolutional networks. It predicts the object using the multiscale perdition method where it uses bounding boxes of different grid sizes which improves the prediction of smaller objects. Using regression YOLOv3 predicts the objectness score for each bounding box. Anchor boxes having the highest overlap with ground truth objects have given 1 as the objectness score whereas other boxes have given 0 as the objectness score. In YOLOv3, the author added Spatial Pyramid Pooling as the backbone of the architecture which improves AP_{50} by 2.7%. 36.2% average precision AP is achieved by YOLOv3-spp in the COCO MS dataset, and 60.6% AP_{50} at 20 FPS is achieved by YOLOv3 2 times faster (Redmon and Farhadi, 2018). Source code: https://pjreddie.com/darknet/yolo/

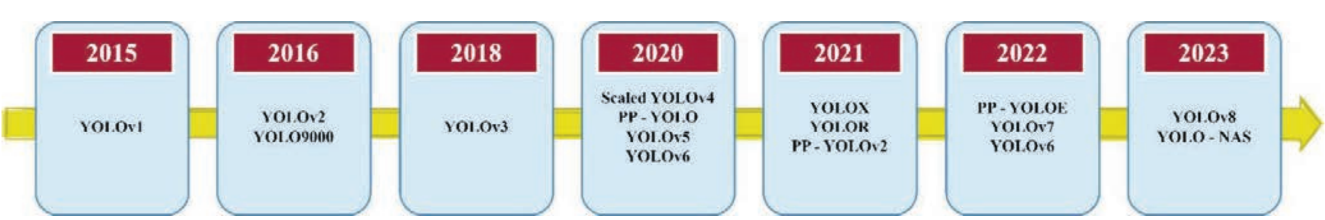

Figure 1. Timeline of different versions of YOLO algorithms.

Table 1. YOLOv2 performance using PASCAL VOC 2007 dataset and PASCAL VOC 2012 dataset.

Detection frame	Resolution	FPS	mAP (%)
VOC 2007 dataset			
YOLOv2 (Redmon and Farhadi, 2017)	288 x 288	91	69
	352 x 352	81	73.7
	416 x 416	67	76.8
	480 x 480	59	77.8
	544 x 544	40	78.6
VOC 2012 dataset			
YOLOv2	544 x 544		73.4
SSD (Liu *et al.*, 2016)	512 x 512		74.9
SSD	300 x 300		72.4
YOLOv1			57.9
Fast R-CNN (Girshick, 2015)			68.4
Faster R-CNN (Zhang *et al.*, 2016)			70.4.

Table 2. Some studies related to object detection using YOLO in agriculture.

Author	YOLO model used	Number of images used for training	Accuracy	Resolution (px) (for training)	Inference
Buzzy *et al.*, 2020	Tiny-YOLOv3	>1000	Inference time 0.01 s F1 score 0.94 FPR 24%	410 x 410	Counting of plant leaves using Tiny YOLOv3 model
Hamidisepehr *et al.*, 2020	YOLOv2	478	AP 97% to 55.99%	570 x 430	Compared different object detection algorithms for corn damage assessment
Bazame *et al.*, 2021	Tiny-YOLOv3		mAP 84%, F1 score 82% Precision 83%, Recall 82%	800 x 800	Mapping, classification, and detection of coffee fruits from videos using computer visions (YOLOv3 Tiny) in Patos de Minas, Brazil regions
Ohnemüller and Briassouli, 2021	Scaled YOLOv4	3782	10% higher mAP score than the baseline model	480 x 480	Improvement of YOLOv4 MS COCO dataset accuracy and efficiency for detection of plants using
Nugroho *et al.*, 2022	YOLOv4	400	Average accuracy 94.6%	1024 x 720 720 x 480	Detection of tomato ripeness using different deep learning models. The prediction results improved as the total loss was reduced
Wiggers *et al.*, 2022	YOLOv3 and YOLOv4	68	AP 84.8% (YOLOv4) Recall 89% (YOLOv4)	416 x 416	Bean plants were captured using UAV and counted using YOLOv3 and YOLOv4 models. From this YOLOv4 performed because of Spatial Pyramid Pooling (SPP)
Zhang and Li, 2022	YOLO-VOLO LS	300	Recall 96.059% Precision 96.014% F1 score 96.039%	384 x 384	Used YOLO for object detection and VOLO for variety identification of early lettuce seedlings
Ajayi *et al.*, 2023	YOLOv5	254	Recall 69.2% Precision 82.3% F1 score 75.2%	416 x 416	Automatic detection of crops classified as banana, sugarcane, pepper, spinach, and weed using the YOLOv5 model in data collected through UAV. Too much of an epoch affects the model's strength
Yeh *et al.*, 2024	YOLOv4	94	Accuracy 0.97 F1 score 0.91	224 x 224 896 x 896	Using Mish function the accuracy of the YOLOv4 model is improved in counting and locating small objects
Haque *et al.*, 2022	YOLOv5	1500	Precision 90% Recall 67% mAP 76% F1 score 81%	416 x 416	Rice leaf diseases were detected and classified using the YOLOv5 model and trained in Google Colab
Sulemane *et al.*, 2022	Tiny YOLO versions	1696 (RGB only)	mAP <70%	406 x 406	To reduce water wastage in orchards, the gaps between the plantations were automatically identified using algorithms and found that Tiny YOLO performed well. Different spectralimages such as NDVI, and NDWI were used for the identification of gaps

YOLOv4

In April 2020, YOLOv4 was introduced by Bochkovskiy and colleagues in ArXiv. YOLOv4 aimed to discover the ideal equilibrium by exploring numerous modifications classified as "bag-of-freebies" and "bag-of-specials." "Bag-of-freebies" encompasses techniques altering the training strategy, and escalating training expenses, yet without a rise in inference time, with data augmentation being the predominant example. Conversely, "bag-of-specials" includes methods that slightly amplify inference costs but markedly enhance accuracy. In YOLOv4 Self-Adversarial Training (SAT) is used where it hides the ground truth object and detects the correct object based on original labels. AP of 43.5% is achieved in MS COCO dataset test-dev 2017 and 65.7% AP_{50} at more than 50 FPS is achieved using NVIDIA V100 (Bochkovskiy *et al*., 2020). Source code: https://github.com/AlexeyAB/darknet

YOLOv5

A few months after the release of YOLOv4, YOLOv5 is released by Glen Jocher. YOLOv5 was developed in PyTroch. They used the Auto Anchor method which adjusts and checks anchor boxes for unfitness for training settings and dataset. Installation of YOLOv5 in IoT devices is easier because it is written in Python programming language. Even though no articles were published by the author for YOLOv5 it is said that YOLOv5 outperforms the other previous versions. Different model versions of YOLOv5 have been released such as YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large) where their convolutional size is changed based on different hardware requirements and applications. YOLOv5x is developed for high-resource devices with high performance whereas YOLOv5s and YOLOv5n are developed for low-resource devices. AP of 50.7% is achieved by YOLOv5x having an image size of pixels in MS COCO dataset test-dev 2017. Source code: https://github.com/ultralytics/yolov5

YOLOv6

Meituan Vision AI Department published YOLOv6 in ArXiv in 2022. Using post-training quantization (PTQ) and quantization-aware training (QAT), YOLOv6 inference speed was boosted without much reduction in performance. YOLOv6 was mainly developed and focused on industry applications. YOLOv6 was suffused with a self-distillation strategy. In network designing for the construction of the backbone RepBlock (Ding *et al*., 2021) is used for small models, and CSP (Wang *et al*., 2020) block is used for large models. For neck (Liu *et al*., 2018) constructs, PAN topology (used in YOLOv5 and YOLOv6) with RepBlocks or CSPStackRep Blocks is adopted to have Rep-PAN an enhanced version of PAN topology. Efficient Decoupled Head is used for head construction. For labelling task alignment learning (TAL) (Feng *et al*., 2021) is considered as more efficient. In YOLOv6, we employ a hybrid-channel strategy to create a more streamlined decoupled head. To be precise, we decrease the count of intermediate 3x3 convolutional layers to just one. The head's width is simultaneously adjusted by the width multiplier for both the backbone and the neck. These adjustments effectively diminish computational expenses, resulting in a decreased inference latency. YOLOv6 adopts an anchor-free detector (anchor point-based) (Ge *et al*., 2021; Tian *et al*., 2019) where the box regression branch accurately anticipates the distance from the anchor point to all four sides of the bounding boxes (Li *et al*., 2022). Source code: https://github.com/meituan/YOLOv6

YOLOv7

Wang and colleagues published YOLOv7 in ArXiv in July 2022. YOLOv7 outperformed all existing object detectors in both accuracy as well as speed from 5 FPS to 160 FPS range. Like YOLOv4, YOLOv7 underwent training solely on the MS COCO dataset without leveraging pre-trained backbones. YOLOv7 introduced several architectural modifications and a set of "bag-of-freebies", contributing to enhanced accuracy without compromising inference speed, with the only impact being on the training time.

ELAN is a strategy developed to improve the learning and convergence efficiency of a deep model by controlling the shortest longest gradient path. YOLOv7 introduced E-ELAN, a feature designed exclusively for models that include an endless number of stacked computational blocks. E-ELAN increases network learning by shuffling and merging cardinality among distinct groups, hence boosting the learning process without affecting the integrity of the original gradient path. It attains the maximum accuracy, exhibiting an astonishing 56.8% average precision (AP), outperforming all other real-time object detectors specifically intended for GPUs, such as the V100, when working at 30 FPS or above (C.-Y. Wang *et al*., 2023). Source code: https://github.com/WongKinYiu/yolov7

YOLOv8

YOLOv8 uses two loss functions to increase its performance. The CIoU and DFL loss functions are utilized for bounding box loss, whereas binary cross-entropy is employed for classification loss. These loss functions have been demonstrated to increase object detection performance, especially when dealing with tiny objects. The YOLOv8-Seg model has a prediction layer and five detection modules, which are similar the detection heads of YOLOv8. YOLOv8 has a semantic segmentation component known as YOLOv8-Seg. This model has exhibited leading performance on a range of object detection and semantic segmentation examinations, all while sustaining speedy processing and effectiveness. The model uses a CSPDarknet53 feature extractor as its backbone, followed by a C2f module instead of the usual YOLO neck architecture. For the prediction of semantic segmentation, the C2f module is followed by two segmentation heads. It also supports different integrations for labelling, training, and deployment. According to the MS COCO dataset test-dev 2017, YOLOv8x achieved an average accuracy (AP) of 53.9% with an image size of 640 pixels. This is a huge improvement compared to YOLOv5's AP of 50.7% on the identical input size 12. YOLOv8x gets a speed of 280 frames per second (FPS) when running on an NVIDIA A100 with TensorRT, as indicated in the paper by Terven and Cordova-Esparza (2023). Source code: https://github.com/ultralytics/ultralytics

YOLOv9

Yolov9 involves the use of PGI (Programmable Gradient Information) and a lightweight network called GELAN (Generalised Efficient Layer Aggregation Network). PGI is an auxiliary supervision framework developed to solve information bottleneck problems such as the loss of information during the feedforward mechanism. PGI consists of three components: main branch, auxiliary reverse branch and multi-level auxiliary branch. An auxiliary reversible branch has been implied in PGI to retain the information that has been lost due to an information bottleneck. By introducing GELAN (formed by combining CSPNet and ELAN), they improved the model's architecture and reduced the information bottleneck (Tishby and Zaslavsky, 2015) which generally occurs during the feedforward mechanism. C.-Y. Wang *et al.* (2024) used the MS COCO dataset to validate the model with other models. The training was done based on train-from-scratch object detection and a total training of 500 epochs was done. From Figure 2, we can see that YOLOv9 performed well by utilizing fewer parameters only. They also conducted ablation studies and found CSP block with ELAN has given good results, accuracy shows a linear relationship for 2 and more than 2 depth of ELAN and CSP block. Source code: https://github.com/WongKinYiu/yolov9

Metrics for measuring the accuracy of YOLO

Mean average precision

For analysis of the efficiency of object identification and segmentation, we often use a metric called mean average precision (mAP). Algorithms such as SSD, YOLO, and R-CNN use mAP to measure their performance. This statistic is often employed in benchmark challenges, including Pascal, VOC, COCO, *etc*. The

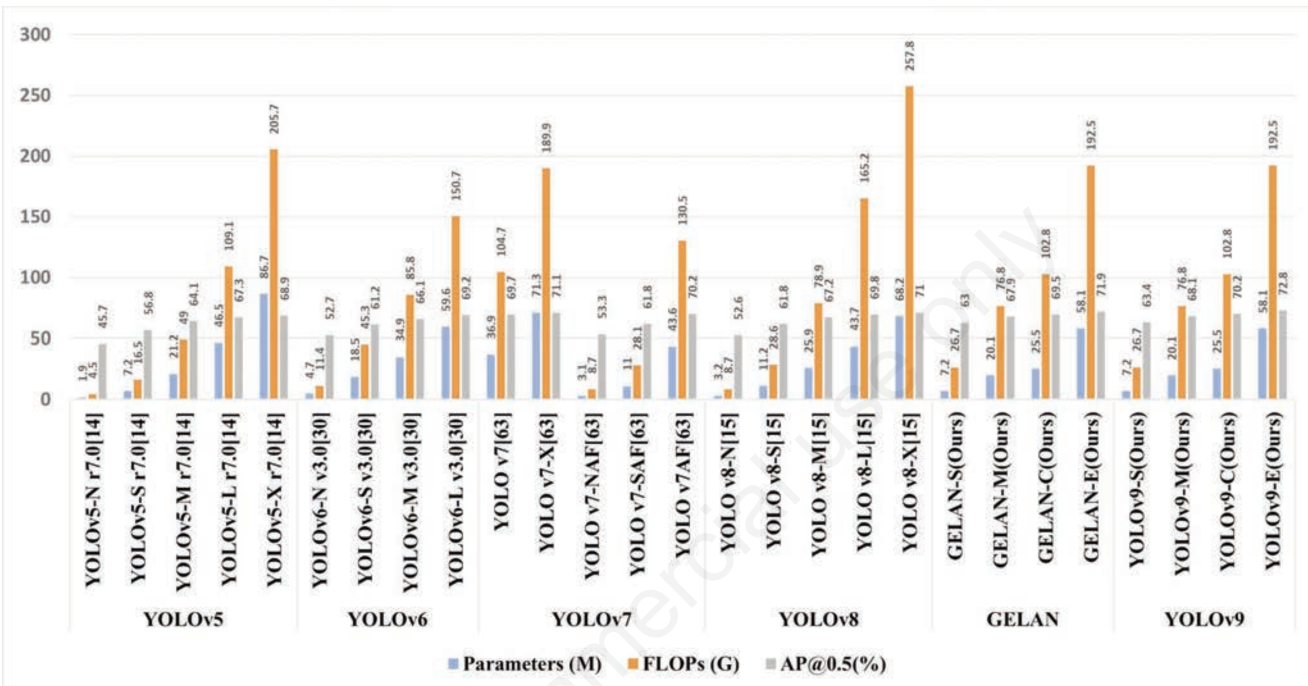

Figure 2. Comparison chart of YOLOv9 with other start of art object detection.

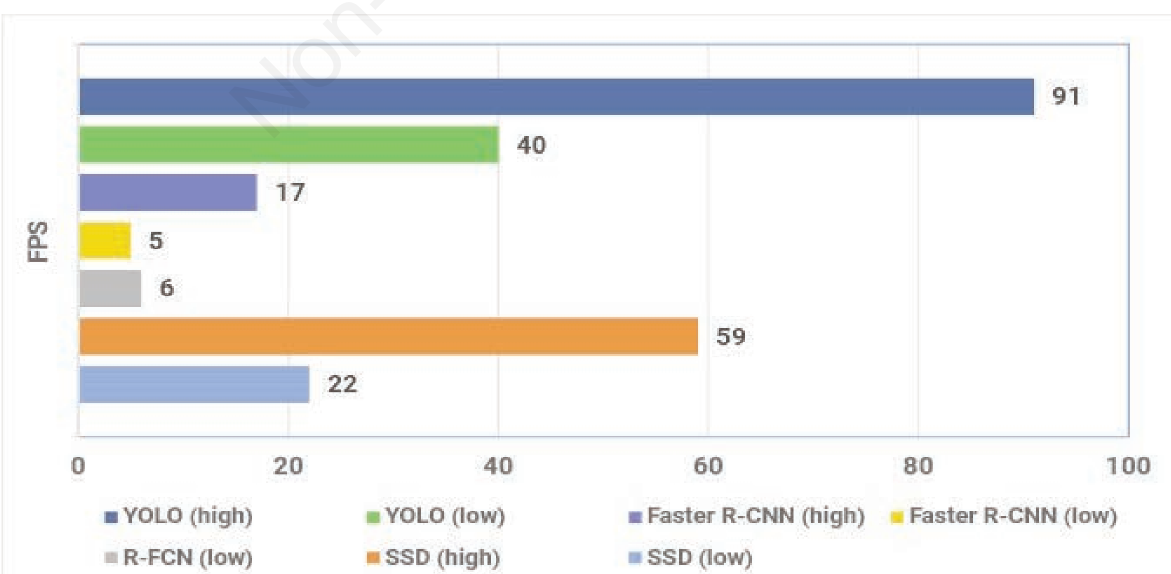

Figure 3. YOLO and other state-of-the-art object detectors speed comparison.

procedure requires obtaining the mean of average accuracy (AP) values, obtained across recall values that vary from 0 to 1. The mAP formula incorporates sub-metrics such as
- Confusion matrix
- Intersection over Union (IoU)
- Recall
- Precision

Confusion matrix

Confusion matrix is a highly famous measure utilized while solving classification difficulties. It can be applied to binary classification as well as to multi-class classification issues. Confusion matrices represent counts from predicted and actual values. The result "TN" stands for true negative which shows the number of negative situations identified accurately. Similarly, "TP" stands for true positive which shows the number of positive cases identified accurately. The term "FP" shows a false positive value, i.e., the number of actual negative cases classed as positive; while "FN" means a false negative value which is the number of actual positive examples classified as negative. To obtain a confusion matrix, users need to pass real values and expected values to the function (Kulkarni *et al*., 2020) (Figure 3).

Intersection over union (IoU)

Bounding boxes: Bounding boxes are rectangular zones that are drawn around the object of interest in images. We use x and y as coordinates to represent the coordinates of the bounding boxes. Object detection methods such as YOLO, CNN, and SSD use bounding boxes with probabilistic classes for identified objects (Breuers *et al*., 2016). Tracking of objects, instance segmentation (Hsu *et al*., 2019), and scene understanding were done in images using bounding boxes.

Intersection over Union (IoU): For the assessment of bounding boxes, we use IoU metrics. It involves the quantification of overlap between the predicted boxes and the ground truth boxes. IoU is the ratio of the area of interest of two bounding boxes and their area of union (Figure 4). The standard Pascal Visual Object Classes (VOC) Challenge 2007 requires that IoU values surpass 0.5 to be considered acceptable (Cowton *et al*., 2019).

Mathematically,

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

where Area of Intersection is the region where both the predicted and ground truth bounding boxes overlap, and Area of Union is the combined region covered by both the predicted and the ground truth bounding boxes.

Intersection over Union values range from 0 to 1. IoU of zero indicates no overlap between the ground truth bounding boxes and the predicted bounding boxes. IoU of one indicates the perfect match i.e. ground truth boxes were precisely aligned with the predicted bounding boxes. Figure 5 represents the threshold values to check whether the predicted value is true positive or false positive. For example, if the acquired IoU value exceeds the predefined threshold (e.g., 0.6) the predicted value will be true positive other than this the predicted value is treated as false positive.

Precision and recall

Precision and recall serve as commonly employed and favoured metrics in classification tasks. Precision assesses the model's accuracy in predicting positive values, thus quantifying

Table 3. YOLO applications in agriculture and key findings.

Application area	Specific task	Dataset used	YOLO version	Performance metrics	Key findings	Source
Crop monitoring	Counting and e stimation	Public and self-build dataset, corn Kernel counting	YOLOv5	mAP0.5 0.70	YOLOv5 outperformed other algorithms in localization and counting of on-ear corn Kernels	Hobbs et al., 2021
Crop monitoring	Citrus disease detection	Self-build dataset, Hafiz Tayyab Rauf dataset, PlantVillage dataset	YOLOv4	mAP 95.4%	They identified and classified the citrus leaf disease into citrus canker, citrus greening and healthy leaf	Song et al., 2020
Crop monitoring	Tomato leaf disease identification	Self-build dataset	YOLOv5	Accuracy 93%	This helps farmer to manage the disease affected and make the prevention measures	Rajamohanan and Latha, 2023
Crop monitoring	Apple flower detection	Apple flower dataset	YOLOv5s	mAP 77.5%	By apple flower detection we can find the apple thinning time and predict the yield	Chen Z. et al., 2022
Crop monitoring	Apple leaf disease detection	Apple leaf disease dataset (ALDD)	EADD-YOLO	mAP 95.5%	This model has few parameters with high calculation accuracy when compared with other models	Zhu et al., 2023
Crop monitoring	Sem blight and brown spot detection in Asparagus	Self-build dataset	YOLOv5s	Accuracy 85.45%	mechanized harvesting of Asparagus to reduce labor costs and increase production efficiency	Yu et al., 2022)
Crop monitoring	Weed detection	Weed and self-build dataset	YOLOv8L	mAP0.5 0.9795, mAP0.5 0.95-0.8123	Detection of weeds in different turfgrass. Such as manila grass, ryegrass and bermudagrass	Sportelli et al., 2023
Quality assessment	Recognize and categorize	bird-eye chillies dataset(private)	YOLOv5	AP 0.90 and mAP 0.94	This works helps farmers to do tedious work of categorization and recognize of bird eye chillies	Abhijit et al., 2023
Quality assessment	Grain counting	Self-build dataset	YOLOv3, YOLOv4, YOLOv5	YOLOv4 Accuracy 97.65%	From this study they found that YOLOv4 is suitable and effective for counting grains in images obtained from agricultural field	Ajikaran et al., 2023

the correctness of positive predictions. This measure is alternatively referred to as the positive predictive value. Recall, also termed sensitivity, evaluates a model's capability to predict positive outcomes effectively (Chen, 2021; Pedregosa *et al*., 2011).

A good F1 score suggests good precision and recall values were attained.

$$Recall = \frac{TP}{(TP + FN)}$$
$$Precision = \frac{TP}{(TP + FP)}$$
$$F1 - Score = \frac{(2 \text{ x } Recall \text{ x } Precision)}{(Recall + Precision)}$$

TP (true positive) = the objects were detected as that object.
FP (false positive) = objects other than those that were detected as those objects.
FN (false negative) = objects were not detected as those objects.

Non-maximum suppression algorithm

Non-maximum suppression (NMS) is employed as a post-processing methodology to enhance object detection by mitigating the occurrence of overlapping bounding boxes and enhancing overall accuracy. During the object detection process, the algorithm commonly produces numerous bounding boxes around the desired object, each accompanied by distinct confidence scores (Figure 6). To eliminate redundant and repetitive boxes and retain only the most accurate ones, we utilize NMS (Hosang *et al*., 2017).

General steps of NMS algorithm followed by Subramanyam (2021)

i) Confidence threshold and IoU threshold values are defined.
ii) Bounding boxes are sorted in descending order by confidence.
iii) If boxes have a confidence lower than the confidence threshold, they are removed.
iv) Then, a loop is executed, keeping the highest confidence box as the first.
v) Calculation of IoU of the current box is done with every remaining box that belongs to the same class.
vi) If the IoU of the two boxes exceeds the IoU threshold, the box with lower confidence from our list is removed.
vii) This step is repeated until we have gone through all the boxes in the list.

YOLO architecture and design principles

YOLO partitions an image into a grid with dimensions *S x S*. Within each grid cell, predictions are made for B bounding boxes and their corresponding confidence levels. The confidence of an object indicates the reliability and accuracy of the bounding box that both identifies and classifies the object (Štancel and Hulič, 2019). The core idea guiding the detection of an object within any grid cell is that the centre of the object must be situated inside that specific grid cell. The detection of a particular object is attributed to the responsibility of the grid cell, aided by an appropriate bounding box (Diwan *et al*., 2023). The grid cell forecasts parameters for a singular bounding box, with the initial five parameters being specific to that bounding box. However, the remaining parameters are common to all bounding boxes within the same grid, regardless of the bounding boxes present.

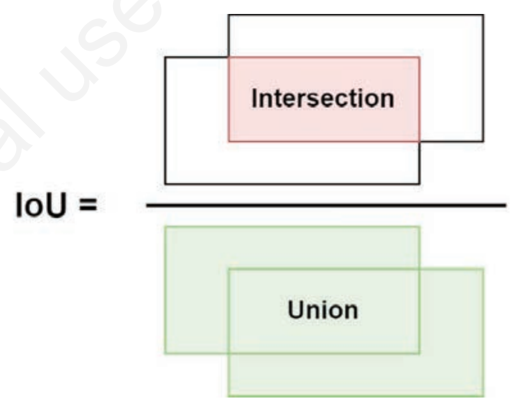

Figure 4. Visual representation of intersection over union (IoU).

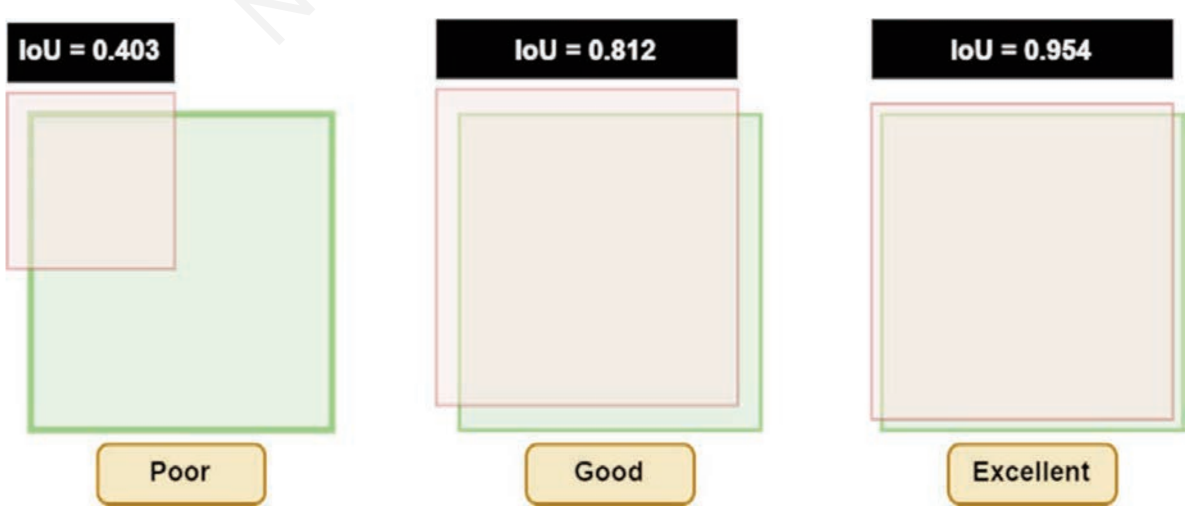

Figure 5. Evaluation of IoU.

p_c	b_x	b_y	b_w	b_h	$p(c_1)$	$p(c_2)$	…	…	$p(c_n)$

The variable denotes the probability of an object being present in the grid through the associated bounding box. The coordinates bx_W, b_y specify the centre of the predicted bounding box, while b_w, b_h indicate the anticipated dimensions of the bounding box. The term $p(c_i)$ signifies the conditional probability of the object belonging to the i^{th} class, given pc, where n is the total number of classes or categories. In total, a grid cell generates $(B \times 5 + n)$ values, where B represents the number of bounding boxes per grid cell. The shape of the output tensor is $S \times S \times (B \times 5 + n)$ since we divided the image into an $S \times S$ grid (Diwan *et al*., 2023).

The confidence score (c_S) for each bounding box in a grid is calculated by multiplying p_c with IoU between the ground truth and the predicted bounding box. If there is no object present in the grid cell, the confidence score is set to zero. We calculate the class-specific score (CSS) for each bounding box across all grid cells. This score reflects both the probability of the class being present in that box and the degree to which the predicted box accurately aligns with the object. Typically, these bounding boxes vary in size to accommodate different shapes and effectively capture various objects, referred to as anchor boxes. The objective is to detect an object in an image with a bounding box where the centre of the object lies. However, multiple object centres may fall within the same bounding box. The authors introduce the term "anchor boxes" to denote the bounding boxes associated with a single grid cell. Anchor boxes constitute a set of standardized bounding boxes, by analysing the dataset and objects in it, the anchor boxes were chosen. These selected anchor boxes aim to encompass most classes/categories by considering diverse combinations of width and height, such as vertical, square, or horizontal rectangles, *etc*. This ensures the representation of various aspect ratios and scales for all objects present in the dataset.

The CNN demonstrates remarkable performance in extracting features from visual input by efficiently transmitting low-level features from the beginning convolutional layers to subsequent ones that are present in a deep CNN. The key challenge lies in precisely identifying multiple objects and determining their precise positions within a single visual input. Effective handling of the YOLO object detection problem is facilitated by two essential CNN features: parameter sharing and the use of multiple filters.

Applications of YOLO in agricultural remote sensing

Detection of objects in satellite imagery

Benayad *et al*. (2023) employed YOLOv3 to discover geomembrane basins using satellite imagery automatically. They used 100 high-resolution satellite photos from Google Earth to train the model for this endeavor. The algorithm focused on classifying five main objects: geomembrane basins, crop areas, roads, houses, and bare fields. To enhance the training process, over 300 basins were enclosed and taught using Darknet, chosen for its exceptional precision and speed. During the evaluation of fresh images, an average precision of 80.6% is achieved, with a precision of 83.3% and recall. However, YOLOv3 exhibited poor performance when dealing with small or closely located objects.

Li *et al*. (2020) detected agricultural greenhouse (AG) in areas of Baoding, Hebei province, China by comparing different algorithms like faster R-CNN, YOLOv3, and SSD (single shot multibox detector). In their work they fused high resolution Gaofen-1 (2 m spatial resolution) and Gaofen- 2 (1 m spatial resolution) satellite images for detection of AG. All the architectures were implemented with the PyTorch framework (deep learning framework). The darknet model of YOLOv3 was converted to the PyTorch framework. By adaptation of the Feature Pyramid Network (FPN) and multilabel classification YOLOv3, the detection is enhanced. Among the different architectures, YOLOv3 performed well with mAP (GF-1 and GF-2) of 90.4% with an FPS of 73. They concluded that to increase the detection quality we need to increase the spatial resolution of the input images.

Tundia *et al*. (2020) in their studies detected minor irrigation structures using Google Satellite images. They compared the speed and accuracy of Faster R-CNN, YOLOv3, Tiny YOLOv3, and RetinaNet. From this Tiny YOLOv3 has the least inference time among the other architectures due to its reduced convolutional layer but its accuracy is reduced (Tables 2 and 3).

Tree detection

For the detection of date palm in regions of the Arabian Peninsula, North Africa, and the Middle East (Jintasuttisak *et al*., 2022) used state-of-the-art YOLOv5 (small, medium, large, and

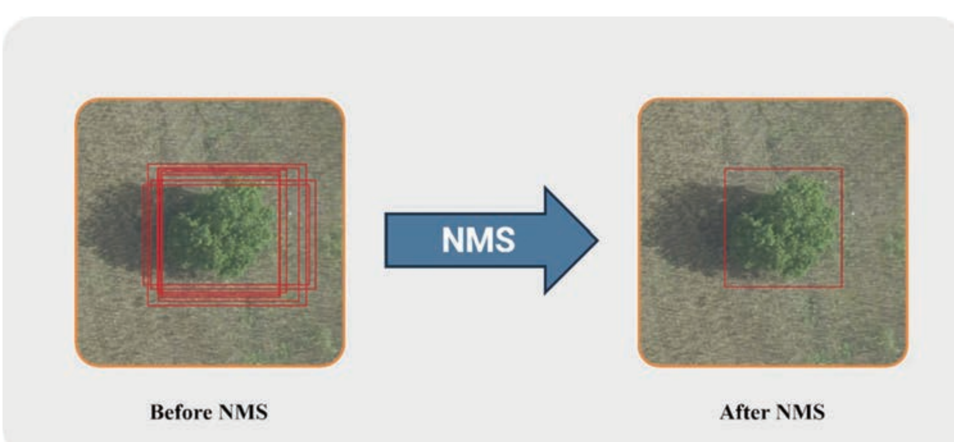

Figure 6. Application of NMS to remove redundant bounding boxes.

extra-large), YOLOv3, YOLOv4, and SSD300 are used. They randomly selected 125 images captured using an RGB drone camera from which they used 60% for training, 20% for validation, and 20% for testing then applied data augmentation which increased the range of the training dataset by five times. From their studies, they concluded that YOLOv5m (medium CNN depth) has performed better than other architecture with mAP of 92.34% and YOLOv5s has less training time (11.33 ms) because of their small CNN network. Nurhabib and Seminar (2022) identified and counted oil palm trees using YOLO with Citra satellite series (1, 2, 3) images. Özer *et al*. (2022) carried out an inter-comparative analysis of YOLOv5 where they compared the results of YOLOv5s, YOLOv5m, and YOLOv5x for the detection of cherry trees in Afyonkarahisar. A total of 889 images were obtained and 80% were used for training rest for testing. YOLOv5s model performed well and obtained precision, recall, and F1 scores of 0.983, 0.978, and 0.980 respectively. Palm tree detection was carried out by Ariyadi *et al*. (2023) using 500 UAV images. The detection is carried out using YOLOv7 with and precision of 98.5%, recall of 98.17%, overall accuracy of 98.31%, and mean average precision of 99.7%. For training, they used 80% of the data, and 20% of the data was used for testing. For each image, the detection time ranged from 17 ms to 18.4 ms.

Monitoring forests enables us to tackle the loss of biodiversity in forest ecosystems and tackle the effects of climate change. Straker and colleagues (2023) in their studies counted the number of trees and segmented the tree crowns using YOLOv5 and Tessellation approach. They used the "For Instance" dataset which consists of 4192 annotated images. The YOLO model performed 27% and 34% better than the Individual tree crown approach at point densities of 50 and 10 points m-2 respectively.

In countries such as India transmission lines passes through cultivation lands. It is important to monitor these transmission lines to avoid damage by trees growing under them. Xu *et al*. (2023) used YOLOv7 and YOLOv4 to classify tree species in transmission line corridors. They classified trees into betel nut, jackfruit, neem, banyan, rubber, and coconut trees with 9531, 4688, 1113, 2336, 2195 and 290 labels, respectively. The images were collected through drones mounted with an MS600 pro multispectral camera. They also applied image augmentations such as flipping, random cropping, colour dithering, rotation, scaling and affine transformation. Using three different band combinations, i.e., R-G-B, NIR-R-G, NIR-G-B the images were inputted. From this, YOLOv7 achieved an average accuracy of 75.77%. from the different band combinations RGB composition acquired higher mean mAP.

Weed detection

Etienne *et al*. (2021) used YOLOv3 for the identification of monocot and dicot weeds in the fields of corn and soybean research plots. They created four different training image sets with images acquired from 10 m above ground level (AGL), 30 m AGL, 30 m and 10 m AGL, and 10 m GL with only dicot weeds. The obtained images were reduced to 416 x 416 pixels before training. Weed instances of 25,560 were manually annotated. 91.48% and 86.13% of average precision (AP) scores were obtained at a threshold of 0.25.

Gallo *et al*. (2023) used UAV images due to their flexibility of data acquisition and high-resolution capability and created 12,113 bounding box annotations from 3000 collected RGB images through UAV. In their studies they used two datasets; one is specifically developed for chicory plantations called the chicory plant (CP) and another one is lincoln beet (LB). For detection, they used YOLOv7 and obtained mAP@0.5, precision and recall of 56.6%, 61.3%, and 62.1% respectively using CP datasets. Using the LB dataset, they obtained mAP, mAP for weeds, and mAP for sugar beets from 51% to 61%, 67.5% to 74.1%, and 34.6% to 48%. For spraying weedicide, Narayana and Ramana (2023) developed object detection using YOLOv7 which trained using two datasets are early crop weed detection dataset (contains 308 images) and the 4weed dataset (contains 618 RGB images). They used 90% of the dataset for training and 10% for the testing set. The model was trained and tested in Google Colab which is a cloud-based environment. mAP of 99.6% was obtained for the Early Weed dataset and 78.53% mAP was obtained for the 4weed dataset.

Fruit detection

Kumar and Kumar (2023) used a new approach to object detection applying a multi-head attention mechanism and depth values to YOLOv7 for the detection of apples in an orchard. The input data was acquired through DJI Mavic mini 3 and images from the video were extracted and then annotated with depth label creation and augmentations such as image mirroring, blurring of image, noisy image, etc we have done on the input. This modified YOLOv7 consists of three detection heads which also help to detect the depth of the apple in the orchard, which is further used to estimate distribution and density. In the end, YOLOv7 couldn't be able to identify all apples while detection but the modified YOLOv7 (i.e., multi-head detection mechanism) detected almost all apples which gave precision, recall, and F1 scores of 0.91, 0.96, and 0.92, respectively. For better marketing, ripeness is an important factor for tomatoes. Thus, tomatoes need to be harvested in the correct stage. For this, (Appe *et al*., 2023) used a modified version of YOLO called CAM – YOLO which used YOLOv5 for detecting ripened tomatoes using convolutional block attention model (CBAM). By this, they achieved an accuracy of 88.1% and performed better than the base YOLO. A tomato health monitoring system was developed by Quach *et al*. (2024) by a combined method of Mobilenetv2 and YOLOv8 for the classification, counting, and detection of tomato. YOLOv8 performed well in the detection of small objects because of the replacement of the C3 model in YOLOv8 from the C2f model used in YOLOv5 (Sohan *et al*., 2024). For annotation, they used RoboFlow and divided the dataset into 6:2:2 ratios for training, validation, and testing respectively. An image resolution of 640 x 640 is used for training for the development of the YOLOv8m and MobileNetv2 models. They achieved 95.76%, 95.74%, and 95.75% of precision, recall, and F1-Score respectively for YOLOv8m and MobileNetv2 models.

Fukada *et al*. (2023) used YOLOv5 (pre-trained using the COCO 2017 dataset) to analyse tomato growth using industry camera devices. This implementation of YOLOv5-based object detection reduced the effort required to analyse crop growth by 80%. Lawal (2021) detected tomatoes in complex environments using YOLO-Tomato (a modified version of YOLOv3). They divide the models into three types. Such as YOLO-Tomato-A, YOLO-Tomato-B and YOLO-Tomato-C. YOLO-Tomato-C has a mish activation function with a front detection layer (FDL) and SPP outperformed the other two types by producing an AP of 99.5%. The use of SPP results in improved AP of the model compared to the other two models. Fruits such as bananas, apricots, apples, and strawberries ripen faster than other fruits. Detection of ripened strawberries in fields by traditional methods is time-consuming and results in spoilage of fruits. An *et al*. (2022) developed a strawberry growth detection algorithm based on YOLOX. Though the model size remains the same as YOLOX, it has 3.64%, 2.04% and 4.08% higher accuracy, recall and precision respectively. This model also solves problems such as the low accuracy of models at complex environments. Chen *et al*. (2023) has overcome the dense and occluded grape detection and missing detection of grapes by developing a lightweight model called GA-YOLO. In this model, SE-CSPGhostnet is designed and introduced in the backbone with 82.79% reduced parameters. It has a mAP of 96.87% and a detection speed of 55.867 FPS. Using artificial intelligence as a classifier and cameras as sensors (Chen M.-C. *et al*., 2022) identified the external quality of fruits such as apples, oranges and lemons based on size, height, width, *etc*. This reduces the labour intensiveness and improves the work speed. They used the YOLOv3 algorithm for fruit detection and acquired an accuracy of 88% by testing on 6000 images. Detecting cherry fruits in open environments results in reduced accuracy due to shading. Thus, Gai *et al*. (2023) introduced an improved version of YOLOv4 called YOLOv4-dense which has a modified backbone of CSPDarknet53 combined with DenseNet. Image augmentation such as flipping, zooming, colour gamut changing, *etc*., were applied on input images. Also, they changed the rectangular bounding boxes into circular bounding boxes. By this the algorithm's speed is increased and feature extraction is also improved. This model produced 0.15 higher mAP than YOLOv4.

With the help of computer vision, we can reduce input costs, and labour costs and increase production efficiency. Gremes *et al*. (2023) counted green oranges directly from trees with green leaf backgrounds using YOLOv4. The performance of used YOLOv4 model was compared with an optimal object detector model, where in the captured video each orange were detected frame by frame. Thus, by combining these two techniques double-counting errors were reduced and the detected and actual oranges were almost equal. The algorithm obtained an mAP50, mAP50:95, precision, recall, F1-score, average IoU of 80.16, 53.83, 0.92, 0.93, 0.93 and 82.08%, respectively.

Disease detection

Amarasingam *et al*. (2022) used a one-stage object detector - YOLOv5 for detecting white leaf disease in sugarcane using DJI Phantom 4 equipped with RTK technology in regions of eastern Sri Lanka. The obtained images were augmented using Python augmentor package 0.2.9. 1200, 240, and 240 images were used for training, testing, and validation process. They conclude, that among the different algorithms used YOLOv5 outperformed other algorithms in precision, mAP@0.5. mAP@0.95 and has a very small model size of 14MB when compared to YOLOR, DETR, and Faster R-CNN. Amarasingam *et al*. (2022) conducted object detection using XGB, RF, DT, and KNN in the same fields and obtained very little accuracy than YOLOv5. Mathew and Mahesh (2022) used YOLOv3 for disease detection in apples. They identified diseases visible in apple tree leaves such as black rot, cedar rust, and apple scab. they classified the image dataset into four classes, and for each class for training and testing, they utilized 1500 and 500 images respectively. At the 700th iteration, they get an average loss of 0.6010. (da Silva *et al*., 2023) their studies for the detection of diseases in Citrus used YOLOv3 and Faster RCNN for detection tasks and concluded YOLO was faster than Faster R-CNN which utilizes less computation power when compared to Faster RCNN. They used LabelImg (Tzutalin, 2015). YOLOv3 and faster R-CNN were run on Keras back-end and evaluated using mAP. While detection they used GPS of mobile to map how the infection spread through the orchard spatially. To detect crop leaf diseases, Dai and Fan (2022) used YOLOv5- CAcT and Plant Village and AI Challenger datasets. The model achieved an accuracy of 94.24% and achieved 59 crop disease categories and 10 crop species with an average inference time of 1.563 ms and a model size of 2 MB.

Madhurya and Jubilson (2023) detected and classified plant leaf disease using the YOLOv7 framework called YR2S (YOLO-Enhanced Rat Swarm Optimizer - Red Fox Optimization (RFO-ShuffleNetv2)). They used PCFAN for the generation of feature maps. The model was detected and classified with a high accuracy of 99.69%. Bandi *et al*. (2023) used YOLOv5 for leaf disease and used U2-Net to remove the background of the affected leaf. They also used a vision transformer for classifying the disease into different stages such as high, medium, and low. They used open datasets like PlantDoc and Plant Village. They achieved an F1 score of 0.57 and a confidence score of 0.2 for YOLOv5 in disease detection. Bachhal *et al*. (2023) in their studies used CCN+YOLO compared with other models for the detection of maize plant disease. They used the Plant Village dataset with 100 images of common rust, 50 images of southern rust, 30 images of maize leaf blight, 30 images of turcicum leaf blight, 70 images of grey leaf spot, and 90 health leaf images. To detect verticillium fungus in olive trees, Mamalis *et al*. (2023) different models of YOLOv5 such as nano, medium, and small. For annotation they used the LabelImg package and classified them as healthy and damaged has withered effect. These images were trained in two image sizes 1216 x 1216 and 640 x 640. The YOLOv5m with model input of 640 x 640 size outperformed other models in their studies. They concluded that as the input size decreases and increases in model capacity, the performance increases. Pine Wilt Disease (PWD) is one of the most dangerous diseases in forest regions because of its rapid spread and management challenges. Traditional methods have more challenges such as excessive time consumption and poor accuracy. Detection of PWD in forest regions helps policy-makers to manage the situation based on the results. Zhu *et al*. (2024) used YOLOv7-SE for the detection of PWD from high-resolution helicopter images. The model achieved a precision rate of 0.9281, F1 score of 0.9117 and a recall of 0.8958. Similarly, Wu *et al*. (2024) used YOLOv3 for detecting PWD from UAV images. They used the CIoU loss function for detecting forest pests and diseases. Yao *et al*. (2024) developed a model called Pine-YOLO (modified version of YOLOv8) which identifies PWD. This model mAP@0.5 at 90.69%, mAP@0.5:0.95 at 49.72%, recall at 85.72%, precision at 91.31% and F1-score at 88.43%.

Crop detection

Espinoza-Hernández *et al*. (2023) determined agave plant density using high-resolution RGB images captured through remote pilot drones. They used YOLOv4 and YOLOv4 tiny for accurate detection at different phenological stages and produced a mean average accuracy of 0.99 for both architectures with 0.95 and 0.96 F1 Score for YOLOv4 and YOLOv4 tiny respectively. Qin *et al*. (2021) developed an algorithm from YOLO called Ag-YOLO which was operated in NCS2(Intel Neural Compute Stick 2). They also compared the developed model with YOLOv3 – Tiny. It is seen that Ag- YOLO outperformed YOLOv3 – Tiny producing a higher accuracy of 0.9205 (F1 Score) and a higher FPS of 36.5 which is two times faster than Tiny YOLOv3 using 12x fewer parameters. Counting rice seedlings traditionally is time-consuming and labour-intensive leading to errors. Yeh *et al*. (2024) developed a YOLO-based approach for counting and marking the location of rice seedlings in the field using a UAV UAV-based approach. In their studies, they used YOLOv4 for counting the seedlings. Though YOLO models are weak in detecting small objects they made changes in images by making data augmentation (image cropping) and changes in the activation function. They implemented the Mish function to improve the accuracy of architecture. They utilized the UAV dataset provided by AIdea (25 rice images with a resolution of 3000 x 2000, 19 images with a resolution of 2304 x 1728). The experiment was conducted in six models, and it was found that model 6 (modified YOLOv4 with mish activation function) had given accuracy of 0.97, an average precision of 0.917, and an F1-score of 0.91. Wang Y. *et al*. (2023) in their studies proposed a YOLOv5-AC model for detecting the efficiency of uncrewed rice transplanters. The model achieved an accuracy of 95.8% and F1 score of 93.39%. Lu *et al*. (2023) modified YOLOv8 for UAV-based object detection and developed a model for precise agriculture. When compared with YOLOv8-N, this model performed well by obtaining 0.921,0.883,0.937and 0.565 precision, recall, AP50 and AP50:95 respectively. Pu *et al*. (2023) used a modified version of YOLOv7 called Tassel-YOLO which used GSConv and VoVGSCSP module in the neck part and SIoU loss function in the head part. Tassel-YOLO achieves 96.14% mAp@0.5, with a counting accuracy of 97.55%. They used the global attention mechanism (GAM) (Liu *et al*., 2021) which improves the feature representation ability through channel attention and the accuracy of spatial data through spatial attention (Wang *et al*., 2018). Images were acquired using a DJI Mavic drone and the image resolution was reduced to 640 x 640 during the detection phase. Due to a lack of knowledge and experience, coffee farmers find it difficult to harvest coffee fruits at the time of harvest. Bazame *et al*. (2022) detected and classified coffee fruits into unripe(green), overripe(dry) and ripe(cherry) using YOLO. They used YOLOv3 and YOLOv4 for detection and classification. YOLOv4 and YOLOv4-tiny models performed well and obtained mAP of 81% and 79%. Camacho and Morocho-Cayamcela (2023) used YOLOv8 for the segmentation and detection of tomatoes at different maturity stages. YOLOv8 produced an R2 of 0.809,0.897,0.968 in ripe, half-ripe and green categories respectively. Tea quality is based on the correct identification and harvesting of perfect tea buds which improves the industry's profit. But the harvest is labour-intensive and time-consuming. By combining the YOLOv3 algorithm, semantic segmentation, minimum bounding rectangle and skeleton extraction (C. Chen *et al*., 2022) located the picking point of tea buds. YOLOv3 obtained an average accuracy of 71.96% for tea bud identification.

Wang C. *et al*. (2023) developed a modified version of YOLO called YOLOv5n for accurate and rapid target detection. This model can be used in lightweight applications and real-time detection. It also compared with other versions of YOLO and obtained an average accuracy of 95.2%.

Discussion

YOLO in agriculture

Monitoring of crops using drones is becoming popular in the upcoming days. We can make the YOLO algorithm use RGB images as well as multispectral bands to analyse chlorophyll content and monitor the stress condition of crops in real-time (Thomson and Sullivan, 2006). Increased fertilizer application results in wastage of input and has adverse effects on the environment. Using YOLO, we can apply fertilizers to specified crops through IoT technology. By this, we can reduce the input cost, reduce the wastage of the raw materials, and protect the environmental impact. Weeds play a crucial role in the agricultural field since they result in reduced yield of crops due to nutrient uptake by them (Nath *et al*., 2024). In some studies, weeds were detected and management practices using YOLO algorithms. By implementing YOLO, we need to distinguish between crops and weeds to apply site-specific management practices such as applying weedicides. By collecting high-resolution images of croplands through drones we can predict the yield and plan harvesting. The data obtained can be integrated with weather data, satellite imagery, and crop models to create a decision support system for farmers (Table 4).

Conclusions

In agriculture, YOLO has been used for crop detection (Sneha *et al*., 2024), fruit detection (Appe *et al*., 2023), and pest and disease detection (Amara *et al*., 2023). However, the base YOLO algorithm struggles with identifying small objects, which poses challenges for detecting crops like rice, sorghum, and maize. To address this, modified versions like Tassel-YOLO for maize tassel detection (Pu *et al*., 2023), Ag-YOLO for broader agricultural studies (Qin *et al*., 2021), and a modified YOLOv4 for cherry detection (Gai *et al*., 2023) have been developed. Further enhancements include modifying the activation function with the mish function and using a modified PANet in YOLOv4 to improve the counting and locating of small objects (Yeh *et al*., 2024). For real-time crop detection using UAVs, computational efficiency is crucial. A modified version of YOLOv7, called YR2S, has been used for disease detection, achieving a high accuracy of 99.69% (Madhurya and Jubilson, 2023). While YOLO has been increasingly used in agriculture, newer versions have not yet been widely adopted. In other applications, such as livestock management, a modified YOLOv3 has been used for detecting and monitoring cow estrus behavior, though challenges remain due to size variations in the animals. The introduction of the DenseBlock structure improved detection in these cases (Z. Wang *et al*., 2024). Developing software for UAVs compatible with low-computation YOLO models is highly valued. The latest YOLOv9 model, which requires less computational power, can enhance real-time performance on low-processing devices. YOLO has shown significant potential and versatility in agriculture and other fields, ongoing improvements and adaptations are necessary to fully leverage its capabilities in real-world applications.

Table 4. Future direction and research opportunities.

Research area	Description	Potential impact on agriculture	Current state of research
Real-time Processing	Need to enhance YOLO for quick inference time.	Enables real time monitoring and helps in decision making	Current improvements in architecture of model and hardware acceleration
Multispectral Imaging	Multispectral data integrating with YOLO.	Better detection of physiographic character study and disease detection	Combining multispectral camera with deep learning
Integration with Robotics	Monitoring and harvesting by integrating YOLO.	improves efficiency and automation of labour-intensive tasks	Computer vision combining with robotics in agriculture
Transfer learning	YOLO adopts to new datasets quickly	Less annotated data is sufficient	Pretrained models like Ag-YOLO are developing

References

Abhijit, Akhil, S., Kumar, V.A., Jose, B.K., Abubeker, K. 2023. Computer vision assisted bird–eye chilli classification framework using YOLO V5 object detection model. In: Shrivastava, V., Bansal, J.C., Panigrahi, B.K. (eds.), Power Engineering and Intelligent Systems. PEIS 2023. Lecture Notes in Electrical Engineering vol 1097. Singapore, Springer.

Ajayi, O G., Ashi, J., Guda, B. 2023. Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images. Smart Agr. Technol. 5:100231.

Ajikaran, R., Hewarathna, A.I., Palanisamy, V., Joseph, C., Thuseethan, S. 2023. An image analysis-based automated method using deep learning for grain counting. IEEE 17th Int. Conf. on Industrial and Information Systems (ICIIS), Peradeniya. pp. 25-30

Amara, S.J., Yamini, S., Sumathi, D. 2023. Pest detection using YOLO V7 model. In: Namasudra, S., Trivedi, M.C., Crespo, R.G., Lorenz, P. (eds.), Data Science and Network Engineering. ICDSNE 2023. Lecture Notes in Networks and Systems,vol 791. Singapore, Springer.

Amarasingam, N., Gonzalez, F., Salgadoe, A.S.A., Sandino, J., Powell, K. 2022. Detection of white leaf disease in sugarcane crops using UAV-derived RGB imagery with existing deep learning models. Remote Sens. (Basel) 14:6137.

An, Q., Wang, K., Li, Z., Song, C., Tang, X., Song, J. 2022. Real-time monitoring method of strawberry fruit growth state based on YOLO improved model. IEEE Access 10:124363-124372.

Appe, S.N., Arulselvi, G., Balaji, G. 2023. CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism. PeerJ Comp. Sci. 9: e1463.

Ariyadi, M.R.N., Pribadi, M.R., Widiyanto, E.P. 2023. Unmanned aerial vehicle for remote sensing detection of oil palm trees using you only look once and convolutional neural network. 10th Int. Conf. on Electrical Engineering, Computer Science and Informatics (EECSI), Palembang. pp. 226-230

Bachhal, P., Kukreja, V., Ahuja, S. 2023. Real-time disease detection system for maize plants using deep convolutional neural networks. Int. J. Comput. Dig. Syst. 14:10263-10275.

Bandi, R., Swamy, S., Arvind, C. 2023. Leaf disease severity classification with explainable artificial intelligence using transformer networks. Int. J. Adv. Technol. Eng. Explor. 10:278.

Bazame, H.C., Molin, J.P., Althoff, D., Martello, M. 2021. Detection, classification, and mapping of coffee fruits during harvest with computer vision. Comput. Electron. Agr. 183:106066.

Bazame, H.C., Molin, J.P., Althoff, D., Martello, M. 2022. Detection of coffee fruits on tree branches using computer vision. Sci. Agric. 80:e20220064.

Benayad, M., Houran, N., Aamir, Z., Maanan, M., Rhinane, H. 2023. Geomembrane basins detection based on satellite high-resolution imagery using DEEP learning algorithms. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 48:75-79.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M. 2020. Yolov4: Optimal speed and accuracy of object detection. a arXiv:2004.10934.

Breuers, S., Yang, S., Mathias, M., Leibe, B. 2016. Exploring bounding box context for multi-object tracker fusion. IEEE Winter Conf. Applications of Computer Vision (WACV), Lake Placid. pp. 18-8.

Buzzy, M., Thesma, V., Davoodi, M., Mohammadpour Velni, J. 2020. Real-time plant leaf counting using deep object detection networks. Sensors (basel) 20:6896.

Camacho, J.C., Morocho-Cayamcela, M.E. 2023. Mask R-CNN and YOLOv8 comparison to perform tomato maturity recognition task. In: Maldonado-Mahauad, J., Herrera-Tapia, J., Zambrano-Martínez, J.L., Berrezueta, S. (eds.), Information and Communication Technologies. TICEC 2023. Communications in Computer and Information Science vol 1885. Cham, Springer.

Chen, C., Lu, J., Zhou, M., Yi, J., Liao, M., Gao, Z. 2022. A YOLOv3-based computer vision system for identification of tea buds and the picking point. Comput. Electron. Agr. 198:107116.

Chen, J., Ma, A., Huang, L., Su, Y., Li, W., Zhang, H., Wang, Z. 2023. GA-YOLO: a lightweight YOLO model for dense and occluded grape target detection. Horticulturae 9:443.

Chen, L.-P. 2021. Practical statistics for data scientists: 50+ essential concepts using R and Python. Technometrics 63:272-273.

Chen, M.-C., Cheng, Y.-T., Liu, C.-Y. 2022. Implementation of a fruit quality classification application using an artificial intelligence algorithm. Sensors Mater. 34:151-162.

Chen, Z., Su, R., Wang, Y., Chen, G., Wang, Z., Yin, P., Wang, J. 2022. Automatic estimation of apple orchard blooming levels using the improved YOLOv5. Agronomy (Basel) 12:2483.

Cheng, L., Li, J., Duan, P., Wang, M. 2021. A small attentional YOLO model for landslide detection from satellite remote sensing images. Landslides 18:2751-2765.

Cowton, J., Kyriazakis, I., and Bacardit, J. 2019. Automated individual pig localisation, tracking and behaviour metric extraction using deep learning. IEEE Access 7:108049-108060.

da Silva, J.C., Silva, M.C., Luz, E.J., Delabrida, S., Oliveira, R.A. 2023. Using mobile edge AI to detect and map diseases in citrus orchards. Sensors (Basel) 23:2165.

Dai, G., Fan, J. 2022. An industrial-grade solution for crop disease image detection tasks. Front. Plant Sci. 13:921057.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J. 202). Repvgg: Making vgg-style convnets great again. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, Nashville. pp. 13728-13737.

Diwan, T., Anirudh, G., Tembhurne, J.V. 2023. Object detection using YOLO: Challenges, architectural successors, datasets and applications. Multimed. Tools Appl. 82:9243-9275.

Dollár, P., Appel, R., Belongie, S., Perona, P. 2014. Fast feature pyramids for object detection. IEEE T. Pattern Anal. 36:1532-1545.

Espinoza-Hernández, J., de Jesús López-Canteñs, G., López-Cruz, I.L., Romantchik-Kriuchkova, E. 2023. Agave plant density using convolutional neural networks on aerial imagery. Agrociencia 57. Online Ahead of Print.

Etienne, A., Ahmad, A., Aggarwal, V., Saraswat, D. 2021. Deep learning-based object detection system for identifying weeds using uas imagery. Remote Sens. (Basel) 13:5182.

Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W. 2021. Tood: Task-aligned one-stage object detection. IEEE/CVF Int. Conf. on Computer Vision (ICCV), Montreal. pp. 3490-3499.

Fukada, K., Hara, K., Cai, J., Teruya, D., Shimizu, I., Kuriyama, T., et al. 2023. An automatic tomato growth analysis system using YOLO transfer learning. Appl. Sci. 13: 6880.

Gai, R., Chen, N., Yuan, H. 2023. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. Neural Comput. Appl. 35:13895-13906.

Gallo, I., Rehman, A.U., Dehkordi, R. H., Landro, N., La Grassa, R., Boschetti, M. 2023. Deep object detection of crop weeds: Performance of YOLOv7 on a real case dataset from UAV images. Remote Sens. (Basel) 15:539.

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J. 2021. Yolox: Exceeding yolo series in 2021. arXiv: 2107.08430.

Girshick, R. 2015. Fast r-cnn. IEEE Int. Conf. on Computer Vision, Santiago. pp. 1440-1448,

Gremes, M.F., Fermo, I.R., Krummenauer, R., Flores, F.C., Gonçalves Andrade, C.M., da Motta Lima, O.C. 2023. System of counting green oranges directly from trees using artificial intelligence. AgriEngineering (Basel) 5:1813-1831.

Hamidisepehr, A., Mirnezami, S.V., Ward, J.K. 2020. Comparison of object detection methods for corn damage assessment using deep learning. T. ASABE 63:1969-1980.

Haque, M.E., Rahman, A., Junaeid, I., Hoque, S.U., Paul, M. 2022. Rice leaf disease classification and detection using yolov5. arXiv: 2209.01579.

Hobbs, J., Khachatryan, V., Anandan, B.S., Hovhannisyan, H., Wilson, D. 2021. Broad dataset and methods for counting and localization of on-ear corn kernels. Front. Robot. AI 8:627009.

Hosang, J., Benenson, R., Schiele, B. 2017. Learning non-maximum suppression. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu. pp. 6469-6477.

Hsu, C.-C., Hsu, K.-J., Tsai, C.-C., Lin, Y.-Y., Chuang, Y.-Y. 2019. Weakly supervised instance segmentation using the bounding box tightness prior. 33rd Conf. Neural Information Processing Systems (NeurIPS 2019), Vancouver.

Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B. 2022. A review of Yolo algorithm developments. Procedia Comput. Sci. 199:1066-1073.

Jintasuttisak, T., Edirisinghe, E., Elbattay, A. 2022. Deep neural network based date palm tree detection in drone imagery. Comput. Electron. Agr. 192:106560.

Kulkarni, A., Chong, D., Batarseh, F.A. 2020. Foundations of data imbalance and solutions for a data democracy. In: Feras A. Batarseh, Yang R. (eds.), Data democracy. Cambridge, Academic Press. pp. 83-106.

Kumar, P., Kumar, N. 2023. Drone-based apple detection: Finding the depth of apples using YOLOv7 architecture with multi-head attention mechanism. Smart Agr. Technol. 5:100311.

Lawal, M.O. 2021. Tomato detection based on modified YOLOv3 framework. Sic. Rep. 11:1477.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. 2022. YOLOv6: A single-stage object detection framework for industrial applications. arXiv: 2209.02976.

Li, M., Zhang, Z., Lei, L., Wang, X., Guo, X. 2020. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, YOLO v3 and SSD. Sensors (Basel) 20:4938.

Lippi, M., Bonucci, N., Carpio, R. F., Contarini, M., Speranza, S., Gasparri, A. 2021. A yolo-based pest detection system for precision agriculture. 29th Mediterranean Conf. Control and Automation (MED). pp. 342-347.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. 2018. Path aggregation network for instance segmentation. IEEE Conf. Computer Vision and Pattern Recognition, Salt Lake City. pp. 8759-8768.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C. 2016. Ssd: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. Lecture Notes in Computer Science vol 9905. Cham, Springer.

Liu, Y., Shao, Z., Hoffmann, N. 2021. Global attention mechanism: Retain information to enhance channel-spatial interactions. arXiv: 2112.05561.

Lu, D., Ye, J., Wang, Y., Yu, Z. 2023. Plant detection and counting: Enhancing precision agriculture in UAV and general scenes. IEEE Access.

Madhurya, C., and Jubilson, E. A. (2023). YR2S: efficient deep learning technique for detecting and classifying plant leaf diseases. IEEE Access 11:116196-116205

Mamalis, M., Kalampokis, E., Kalfas, I., Tarabanis, K. 2023. Deep learning for detecting verticillium fungus in olive trees: Using YOLO in UAV imagery. Algorithms (Basel) 16:343.

Mathew, M.P., Mahesh, T.Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. Signal Image Video P. 16:841-847.

Narayana, C.L., Ramana, K.V. 2023. An efficient real-time weed detection technique using YOLOv7. Int. J. Adv. Comput. Sci. Appl. 14:550-556.

Nath, C.P., Singh, R.G., Choudhary, V.K., Datta, D., Nandan, R., Singh, S.S. (2024). Challenges and alternatives of herbicide-based weed management. Agronomy (Basel) 14:126.

Nugroho, D.P., Widiyanto, S., Wardani, D.T. 2022. Comparison of deep learning-based object classification methods for detecting tomato ripeness. Int. J. Fuzzy Logic Intell. Syst. 22:223-232.

Nurhabib, I., Seminar, K. 2022. Recognition and counting of oil palm tree with deep learning using satellite image. IOP Conf. Ser. Earth Environ. Sci. 974:012058.

Ohnemüller, L., Briassouli, A. 2021. Improving accuracy and efficiency in plant detection on a novel, benchmarking real-world dataset. IEEE Int.Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento-Bolzano. pp. 172-176.

Özer, T., Akdoğan, C., Cengız, E., Kelek, M.M., Yildirim, K., Oğuz, Y., Akkoç, H. 2022. Cherry tree detection with deep learning. IEEE Conf. on Innovations in Intelligent Systems and Applications (ASYU), Antalya. pp. 1-4.

Papageorgiou, C.P., Oren, M., Poggio, T. 1998. A general framework for object detection. IEEE 6th Int. Conf. on Computer Vision, Bombay. pp. 555-562.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. 2011. Scikit-learn: Machine learning in Python. J. Machin. Learning Res. 12:2825-2830.

Pu, H., Chen, X., Yang, Y., Tang, R., Luo, J., Wang, Y., Mu, J. 2023. Tassel-YOLO: A new high-precision and real-time method for maize tassel detection and counting based on UAV aerial images. Drones 7:492.

Qin, Z., Wang, W., Dammer, K.-H., Guo, L., Cao, Z. 2021. Ag-YOLO: A real-time low-cost detector for precise spraying with case study of palms. Front. Plant Sci. 12:753603.

Qing, Y., Liu, W., Feng, L., Gao, W. 2021. Improved Yolo network for free-angle remote sensing target detection. Remote Sens. (Basel) 13:2171.

Quach, L.-D., Quoc, K.N., Quynh, A.N., Ngoc, H.T., Nghe, N.T. 2024. Tomato health monitoring system: tomato classification, detection, and counting system based on YOLOv8 model with explainable MobileNet models using Grad-CAM++. IEEE Access 12:9719-9737.

Rajamohanan, R., Latha, B.C. 2023. An optimized YOLO v5 model for tomato leaf disease classification with field dataset. Eng. Technol. Appl. Sci. Res. 13:12033-12038.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. 2016. You only look once: Unified, real-time object detection. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas. pp. 779-788.

Redmon, J., Farhadi, A. 2017. YOLO9000: better, faster, stronger. IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu. pp. 6517-6525.

Redmon, J., Farhadi, A. 2018. Yolov3: An incremental improvement. arXiv: 1804.02767.

Roy, A.M., Bhaduri, J., Kumar, T., Raj, K. 2023. WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. Ecol. Inform. 75:101919.

Sneha, N., Sundaram, M., Ranjan, R. 2024. Acre-scale grape bunch detection and predict grape harvest using YOLO deep learning network. SN Comput. Sci. 5:250.

Sohan, M., Sai Ram, T., Reddy, R., Venkata, C. 2024. A review on YOLOv8 and its advancements. Int. Conf. on Data Intelligence and Cognitive Informatics. pp 529-545

Song, C., Wang, C., Yang, Y. 2020. Automatic detection and image recognition of precision agriculture for citrus diseases. IEEE Eurasia Confe. on IOT, Communication and Engineering, Yunlin, Taiwan. pp. 187-190.

Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S. 2011. Contextualizing object detection and classification. IEEE T. Pattern Anal. 37:13-27.

Sportelli, M., Apolo-Apolo, O.E., Fontanelli, M., Frasconi, C., Raffaelli, M., Peruzzi, A., Perez-Ruiz, M. 2023. Evaluation of YOLO object detectors for weed detection in different turfgrass scenarios. Appl. Sci. 13:8502.

Štancel, M., Hulič, M. 2019. An introduction to image classification and object detection using YOLO detector. Proc. CEUR Workshop.

Straker, A., Puliti, S., Breidenbach, J., Kleinn, C., Pearse, G., Astrup, R., Magdon, P. 2023. Instance segmentation of individual tree crowns with YOLOv5: A comparison of approaches using the ForInstance benchmark LiDAR dataset. ISPRS Open J. Photogramm. Remote Sens. 9:100045.

Subramanyam, V.S. 2021. Non Max Suppression (NMS). Available from: https://medium.com/analytics-vidhya/non-max-suppression-nms-6623e6572536

Sulemane, S., Matos-Carvalho, J.P., Pedro, D., Moutinho, F., Correia, S.D. 2022. Vineyard gap detection by convolutional neural networks fed by multi-spectral images. Algorithms 15:440.

Teng, L., Li, H., Karim, S. 2019. DMCNN: a deep multiscale convolutional neural network model for medical image segmentation. J. Healthc. Eng. 2019:8597606.

Terven, J., Cordova-Esparza, D. 2023. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv: 2304.00501.

Thomson, S.J., Sullivan, D.G. 2006. Crop status monitoring using multispectral and thermal imaging systems for accessible aerial platforms. 2006 ASAE Annual Meeting 061179.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z. 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agr. 157:417-426.

Tishby, N., Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. IEEE Information Theory Workshop, Jerusalem. pp. 1-5

Tundia, C., Tank, P., Damani, O.P. 2020. Aiding irrigation census in developing countries by detecting minor irrigation structures from satellite imagery. Proc. 6th Int. Conf. on Geographical Information Systems Theory, Applications and Management. pp. 208-215.

Tzutalin, D. 2015. tzutalin/labelImg. Available from: https://github.com/tzutalin/labelImg

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, Vancouver. pp. 7464-7475.

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H. 2020. CSPNet: A new backbone that can enhance learning capability of CNN. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, Seattle. pp. 1571-1580

Wang, C.-Y., Yeh, I.-H., Liao, H.-Y. M. 2024. YOLOv9: learning what you want to learn using programmable gradient information. arXiv: 2402.13616.

Wang, C., Wang, C., Wang, L., Wang, J., Liao, J., Li, Y., Lan, Y. 2023. A lightweight cherry tomato maturity real-time detection algorithm based on improved YOLOV5n. Agronomy (Basel) 13:2106.

Wang, H., Fan, Y., Wang, Z., Jiao, L., Schiele, B. 2018. Parameter-free spatial attention network for person re-identification. arXiv: 1811.12150.

Wang, Y., Fu, Q., Ma, Z., Tian, X., Ji, Z., Yuan, W., et al. 2023. YOLOv5-AC: a method of uncrewed rice transplanter working quality detection. Agronomy (Basel) 13:2279.

Wang, Z., Hua, Z., Wen, Y., Zhang, S., Xu, X., Song, H. 2024. E-YOLO: Recognition of estrus cow based on improved YOLOv8n model. Expert Syst. Appl. 238:122212.

Wiggers, K.L., Pohlod, C.D., Orlovski, R., Ferreira, R., Santos, T.A. 2022. Detection and counting of plants via deep learningusing images collected by RPA. Rev. Bras. Cien. Agr. 17:1.

Wu, D., Lv, S., Jiang, M., Song, H. 2020. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput. Electron. Agr. 178:105742.

Wu, Y., Yang, H., Mao, Y. 2024. Detection of the pine wilt disease using a joint deep object detection model based on drone remote sensing data. Forests (Basel) 15:869.

Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., Lan, X. 2020. A review of object detection based on deep learning. Multim. Tools Appl. 79:23729-23791.

Xu, S., Wang, R., Shi, W., Wang, X. 2023. Classification of tree species in transmission line corridors based on YOLO v7. Forests (Basel) 15:61.

Yao, J., Song, B., Chen, X., Zhang, M., Dong, X., Liu, H., et al. 2024. Pine-YOLO: a method for detecting pine wilt disease in unmanned aerial vehicle remote sensing images. Forests (Basel) 15:737.

Yeh, J.-F., Lin, K.-M., Yuan, L.-C., Hsu, J.-M. (2024). Automatic counting and location labeling of rice seedlings from unmanned aerial vehicle images. Electronics (Basel) 13:273.

Yin, S., Zhang, Y., Karim, S. 2018. Large scale remote sensing image segmentation based on fuzzy region competition and Gaussian mixture model. IEEE Access 6:26069-26080.

Yin, S., Zhang, Y., Karim, S. 2019. Region search based on hybrid convolutional neural network in optical remote sensing images. Int. J. Distrib. Sensor N. 15:1550147719852036.

Yu, J., Zhang, C., Wang, J., Zhang, M., Zhang, X., Li, X. 2023. Research on asparagus recognition based on deep learning. IEEE Access 11:117362-117367.

Zakria, Z., Deng, J., Kumar, R., Khokhar, M.S., Cai, J., Kumar, J. 2022. Multiscale and direction target detecting in remote sensing images via modified YOLO-v4. IEEE J. Sel. Top. Appl. 15:1039-1048.

Zhang, H., Cloutier, R.S. 2021. Review on one-stage object detection based on deep learning. EAI Endor. T. e-Learning 7:e5.

Zhang, L., Lin, L., Liang, X., He, K. 2016. Is faster R-CNN doing well for pedestrian detection? ECCV 2016. Lecture Notes in Computer Science, vol 9906. Springer, Cham. pp 443-457.

Zhang, P., Li, D. 2022. YOLO-VOLO-LS: a novel method for variety identification of early lettuce seedlings. Front. Plant Sci. 13:806878.

Zhong, Z., Jin, L., Xie, Z. 2015. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. 3rd Int. Conf. on Document Analysis and Recognition, Tunis. pp. 846-850.

Zhou, P., Ni, B., Geng, C., Hu, J., Xu, Y. 2018. Scale-transferrable object detection. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, Salt Lake City. pp. 528-537.

Zhu, S., Ma, W., Wang, J., Yang, M., Wang, Y., Wang, C. 2023. EADD-YOLO: An efficient and accurate disease detector for apple leaf using improved lightweight YOLOv5. Front. Plant Sci. 14:1120724.

Zhu, X., Wang, R., Shi, W., Liu, X., Ren, Y., Xu, S., Wang, X. 2024. Detection of pine-wilt-disease-affected trees based on improved YOLO v7. Forests (Basel) 15:691.