

Enhancement of YOLOv5 for automatic weed detection through backbone optimization

Mohammed Habib¹, Salma Sekhra¹, Adil Tannouche², Youssef Ounejjar¹

¹Laboratory of Spectrometry, Materials and Archaeomaterials "LASMAR", Faculty of Sciences, Moulay Ismail University, Meknes, Morocco

²Laboratory of Engineering and Applied Technology "LITA", High School of Technology, Université Sultan Moulay Slimane, Beni Mellal, Morocco

Article Info

Article history:

Received Mar 15, 2024

Revised Jul 6, 2024

Accepted Oct 18, 2024

Keywords:

Computer vision
Convolutional neural networks backbone
Deep learning
Object detectors
Smart farming
Weed detection

ABSTRACT

In the context of our research project, which involves developing a robotic system capable of eliminating weeds using deep learning techniques, the selection of powerful object detection model is essential. Object detectors typically consist of three components: backbone, neck, and prediction head. In this study, we propose an enhancement to the you only look once version 5 (YOLOv5) network by using the most popular convolutional neural networks (CNN) networks (such as DarkNet and MobileNet) as backbones. The objective of this study is to identify the best backbone that can improve YOLOv5's performance while preserving its other layers (neck and head). In terms of detecting and ultra-localizing pea crops. Additionally, we compared their results with those of the most commonly used object detectors. Our findings indicate that the fastest models among the networks studied were MobileNet, YOLO-tiny, and YOLOv5, with speeds ranging from 5 to 14 milliseconds per image. Among these models, MobileNetv1 demonstrated the highest accuracy, achieving average precision (AP) score of 89.3% for intersection over union (IoU) threshold of 0.5. However, the accuracy of this model decreased when we increased the threshold, suggesting that it does not provide perfect crop delineation. On the other hand, while YOLOv5 had a lower AP score than MobileNetv1 at an IoU threshold of 0.5, it exhibited greater stability when faced with variations in this threshold.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Mohammed Habib
Laboratory of Spectrometry, Materials and Archaeomaterials "LASMAR", Faculty of Sciences
Moulay Ismail University
Meknes, Morocco
Email: habibmohamedtsei@gmail.com

1. INTRODUCTION

With the growing global population and the need to produce more food sustainably, farmers and researchers are turning to advanced technologies, including machine learning to optimize agricultural processes. This technology helps solve various agricultural issues, such as weed control. Machine learning methods can aid in the precise identification and management of weeds, thereby reducing the need for chemical products and improving crop yields.

In this context, our research team has already made significant progress in solving problems related to weed control. In their work, Tannouche *et al.* [1] utilized a method based on a set of discriminant classifiers constructed using Haar-like features. They achieved a notable milestone by developing an efficient shape

descriptor specifically designed to distinguish between two types of weed species: monocots and dicots [2]. Both studies employed machine learning techniques with the aim of optimizing herbicide usage.

In the context of site-specific weed management (SSWM), which aims to target and precisely control weeds, we have developed research to meet this need. Our studies focused on using deep learning models for detecting and identifying crops among weeds. In our initial study, we trained the Faster region-based convolutional neural network (RCNN) residual networks (ResNet)-50 model to detect and locate crops with high precision [3]. In a subsequent project, we evaluated the performance of the you only look once version 5 (YOLOv5) object detector by testing different sizes and learning methods, including training from scratch and transfer learning. This allowed us to achieve high accuracy and fast localization speeds for crops. Furthermore, we explored various segmentation networks to effectively distinguish crops from weeds [4]. This analysis enabled us to accurately determine the positions of weeds by eliminating the crop regions identified by YOLOv5. In another study, we proposed a classification model based on ResNet and MobileNet for identifying and differentiating between crops and weeds, and compared its performance with popular convolutional neural networks (CNNs) [5].

In the same context, Wang *et al.* [6] summarized the progress made in weed detection using computer vision and image processing techniques. All these techniques have focused on direct detection of weed features, a delicate task in view of the wide variety of weed species. Our weed detection method involves a two-step approach: vegetation/ground discrimination (segmentation) followed by crop/weed discrimination (object detection), which allows us to extract the weeds alone.

Object detection continues to be a significant area of research in deep learning, with various applications showcasing the capabilities of object detectors in solving computer vision problems. These include facial recognition [7], pedestrian detection [8], video analysis [9], and logo detection [10]. Currently, there are two major categories of object detectors: two-stage detectors, such as RCNN and its variants, and single-stage detectors, which are generally more powerful than the former, like the YOLO model and its versions.

The remarkable success of YOLO's architecture has led to numerous research efforts aimed at its improvement. In [11], [12], the authors optimized the number and size of bounding boxes generated by YOLOv3 and YOLOv5 during training using K-means and K-means++ clustering techniques. A feature fusion method called PB-FPN was proposed, building upon path aggregation network (PANet) and bidirectional feature pyramid network (BiFPN) techniques [13]. Another study focused on optimizing the YOLOv5 model specifically for plant disease identification [14]. Additionally, researchers proposed a dilation technique for the spatial pyramid module (SPP) to incorporate multi-scale information and address scale variation issues in YOLOv3 [15]. In the domain of kiwi agriculture fault detection, the authors of [16] introduced several enhancements to YOLOv5. These improvements included the addition of a small target detection layer and a SELayer, as well as modifications to the loss function from distance-intersection over union (DioU) to complete-intersection over union (CioU).

When it comes to enhancing the accuracy of object detection networks, architecture plays a vital role among various parameters. Typically, object detectors consist of three main components: the backbone, the neck, and the prediction head. In this research, we propose an enhancement to the YOLOv5 network by using the most popular CNN networks (such as DarkNet, and MobileNet) as backbones. The objective of this study is to identify the best backbone that can improve YOLOv5's performance while preserving its other layers (neck and head). We focus on evaluating their performance in the accurate detection and localization of pea crops. Additionally, we will compare the results obtained by these networks with those of other commonly used object detectors.

2. METHOD

In this section we will show you the methods and materials we used to carry out this study. A modern object detector consists of 3 main parts, a backbone, a neck, and a head (Figure 1). The backbone component of an object detection model consists of a set of layers responsible for extracting features from input images. These layers aim to capture detailed information from the images. The neck component, on the other hand, is typically used to gather and merge the output features obtained from the backbone. It then sends feature maps of different sizes back to the detection heads. This process enables the model to detect objects of varying sizes within the image. Finally, the detection head performs calculations for bounding box regression and probability estimation. In this study we propose to make a modification to the architecture of YOLOv5 by replacing its backbone with the popular CNNs: DarkNet, MobileNet, ResNet and visual geometry group (VGG), as shown in Figure 1, and to evaluate the performance of each model.

2.1. YOLOv3

YOLO, or "you only look once," is a well-known one-stage object detector that predicts the location and class of objects in a single pass, making it generally faster than detectors using region proposal networks (RPN) [17]. YOLOv3, one of the most popular versions, enhances the YOLO architecture, resulting in

improved accuracy and speed. YOLOv3's architecture consists of three main components: the backbone, neck, and head. It uses Darknet53 as the backbone, comprising 53 convolutional layers inspired by Darknet19, including Bottleneck modules with shortcut connections for efficient feature extraction.

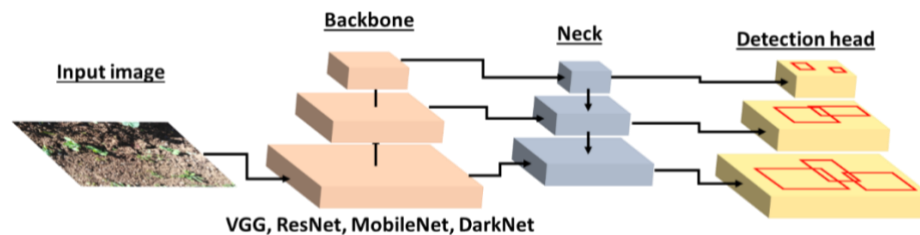


Figure 1. The structure of a modern object detector network

The neck incorporates the feature pyramid network (FPN) method to fuse low and high-level features, upsampling the backbone's output and combining it with low-level outputs. The head uses k-means clustering to determine bounding box coordinates, distributing nine clusters among three output scales [18], [19]. In a separate study, researchers proposed YOLOv3-tiny, a compact version using Darknet19 as the backbone and a simplified neck with a single concatenation. YOLOv3-tiny employs two feature maps of different scales for detection predictions, which are then processed by a similar sensing head as in YOLOv3 [20]. This compact version balances speed and accuracy, making it suitable for embedded systems with limited computational resources.

2.2. YOLOv4

YOLOv4 is a recent iteration of YOLO detectors, featuring a modern architecture with CSPDarkNet53 as its backbone. Built upon the DarkNet 53 architecture used in YOLOv3, CSPDarkNet53 includes Conv modules and a new BottleNeckCSP module inspired by the CSPNet technique [21]. After numerous experiments, the authors determined that CSPDarkNet53 is the optimal model compared to other networks tested. YOLOv4's neck incorporates an additional SPP module to generate representations from images of arbitrary sizes [22]. It also uses PANet in the neck, improving object localization accuracy through enhanced feature fusion from bottom to top and top to bottom. The head of YOLOv4 is similar to that of YOLOv3. Another study introduced YOLOv4-tiny, a smaller version with a reduced CSPDarkNet53 backbone featuring 3 BottleNeckCSP modules instead of 27. The extracted features are processed by a reduced neck based on the FPN technique, similar to YOLOv3-tiny, before being sent to the YOLOv3 head for predicting location and class [23].

2.3. YOLOv5

YOLOv5, a recent YOLO version, is known for its impressive accuracy and speed. Its architecture closely resembles YOLOv4, using the CSPDarkNet53 backbone with modifications. YOLOv5 includes unique modules like C3, comprising three Conv modules and a BottleNeckCSP, which mitigates gradient information duplication. It also features the SPPF module for improved feature expression, inspired by SPP networks but faster. YOLOv5's neck uses PANet for feature fusion, and the outputs are sent to the detection head, similar to YOLOv3, for predicting object location and class [13].

2.4. The MobileNet networks

MobileNet is renowned for its speed and lightweight architecture, ideal for mobile applications. It utilizes depthwise separable convolutions (DWConv) for efficient feature extraction, consisting of a depthwise convolution layer followed by batch normalization (BatchNorm) and ReLU activation. MobileNet v1 uses DWConv-sep blocks, which include DWConv modules followed by Conv modules [24]. MobileNet v2 introduces BottleNeck-Mob blocks, featuring convolution layers with an expansion layer at the input and a projection layer at the output, incorporating shortcut connections and the DWConv module at their core [25].

2.5. The ResNet models

ResNet was introduced to tackle the vanishing gradient problem in training deep neural networks. ResNet models are similar to VGG networks but include shortcut connections that link the input and output of each module, ensuring smooth gradient flow. These connections help mitigate the vanishing gradient issue.

Depending on the model size, ResNet can contain different numbers of ResConv modules, such as 34, 50, or 101 [26].

2.6. The VGG models

VGG networks are recognized as fundamental convolutional networks known for their straightforward architecture. They consist of a sequence of convolutional layers (Conv) stacked on top of each other. The various versions of VGG are distinguished by the number of convolutional layers employed, such as VGG16 and VGG19 [27].

2.7. Generalized intersection over union loss

When training object detectors, the loss is determined by two functions, the classification loss and the bounding box regression loss. The most commonly used loss function for bounding box regression is the intersection over union (IoU) and its derivations. The IoU is defined by (1).

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (1)$$

where A is the prediction and B is the true bounding box.

The loss function is expressed as a distance, $Loss_{IoU} = 1 - IoU$. There are other functions derived from IoU that enable the evaluation of the dissimilarity between the predicted and true bounding boxes, even in cases where they do not overlap. One such function is the generalized intersection over union (GIoU), which addresses the issue of disjoint bounding boxes (A and B). GIoU possesses the same scale invariance properties as IoU and is defined by the (2) and (3).

$$GIoU(A, B) = IoU(A, B) - \frac{|C| - |A \cup B|}{|C|} \quad (2)$$

$$Loss_{GIoU} = 1 - GIoU(A, B) = 1 - IoU(A, B) + \frac{|C| - |A \cup B|}{|C|} \quad (3)$$

with C is the small box that encloses the boxes A and B [13].

In this study, we will utilize the GIoU function to calculate the bounding box regression loss at each iteration. For classification, we used the binary cross entropy with logits loss function from the YOLOv5 model training. The BCEWithLogits loss combines the functionalities of two Sigmoid functions and the BCELoss by using the log-sum-exp method. The BCELossWithLogits is defined as (4).

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^T, l_n = -w_n [y_n \cdot \log(\sigma(x_n)) + (1 - y_n) \cdot \log(1 - \sigma(x_n))] \quad (4)$$

N is the batch size. The result is a more numerically stable function compared to using Sigmoid followed by BCELoss separately [28].

2.8. Dataset acquisition

DataSet acquisition still remains one of the great challenges of DeepLearning, a great DataSet means a good learning. In our study, we utilized a previously collected and prepared DataSet from our earlier work [3] for model training. The images we have collected contain the pea crop and the weeds with these different species. Figure 2 illustrates the techniques employed for image acquisition. To further enhance the DataSet, we employed image processing tools for data augmentation.

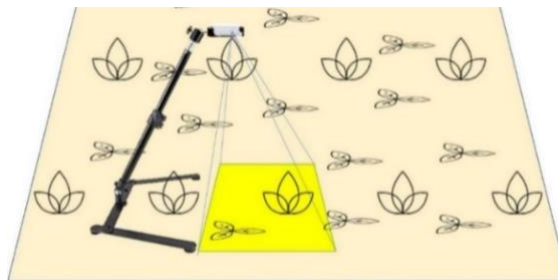


Figure 2. The materials used for the acquisition of dataset a digital camera was fixed in 40 cm from the ground

For image labeling, we utilized the YOLO models' specific annotation format. Each bounding box is characterized by the coordinates of its center point (x, y), its length, width, and the corresponding object class. These coordinates are normalized and saved in a text file. The DataSet was divided into three subsets: 7,360 images for training, 1,840 images for validation, and 54 images for testing.

3. RESULTS AND DISCUSSION

In this section, we will present the results of training and validation of object detectors using the previously discussed backbone architectures. We will evaluate their performance in crop identification and ultra-localization. The models we will examine are derived from the neural networks already studied. Table 1 provides an overview of the constructed models, including their size and computational requirements. It summarizes the characteristics of each model.

Table 1. The properties of the trained models

Model Name	Backbone	Neck	Layers	Parameters	Gflops
YOLOv5	CSPDarkNet53 (mod)	PANet	214	7235389	16.6
YOLOv4	CSPDarkNet53	PANet	383	40020445	102.1
YOLOv3	DarkNet53	FPN	262	61949149	156.6
YOLOv4-Tiny	CSPDarkNet19	FPN-tiny	91	3105526	6.5
YOLOv3-Tiny	DarkNet19	FPN-tiny	49	8852366	13.3
VGG16	VGG16	FPN-tiny	68	25996222	272.1
VGG19	VGG19	FPN-tiny	77	31307198	340.1
Mobnetv1	Mobilenetv1	FPN-tiny	105	16973630	13.3
Mobnetv2	Mobilenetv2	FPN-tiny	200	16774670	18.9
Resnet50	ResNet50	FPN-tiny	284	61283390	89.5

The Gflops refers to the number of floating-point operations per second, expressed in Giga. As shown in the Table 1, YOLOv4, YOLOv3, and VGG models have high computational requirements, indicating that their execution and training times will be relatively long. On the other hand, YOLO-tiny and MobileNet models have lower computational demands, allowing them to be executed efficiently even on low-performance processors.

3.1. Training preparation

We built these models with the functionality of the Pytorch library with which the YOLOv5 model was written. We trained, validated and tested all the created models under the same conditions in terms of dataset, pre-processing techniques, and using the Tesla T4 graphics processor. The hyperparameters for model training are listed in Table 2.

Table 2. The hyper-parameters of the training of the models

Epochs	Warmup epochs	Batch Size	Img. Size	lr0	Optimizer	Momentum	Loss function
50	3	16	320	0.01	SGD	0.937	BCEWithLogitsLoss, GIoU

3.2. Validation results

After training our models, we validated their performance by introducing the validation images and comparing the predicted bounding boxes with the true bounding boxes. We varied the size of the input images from 320 to 640 to observe its impact on the results. Precision and recall are the most commonly used metrics for evaluating the performance of a CNN during validation. They are defined as (5) and (6):

$$p = \frac{TP}{TP+FP} \quad (5)$$

$$r = \frac{TP}{TP+FN} \quad (6)$$

TP representing true positive, TN for true negative, FP for false positive, and FN for false negative. In our case, the positive class corresponds to peas, while the negative class represents the background [5].

Precision recall (PR) curves are used to represent precision versus recall, where precision measures result accuracy and recall measures the relevance of results. The x-axis of PR curves shows recall, and the

y-axis shows precision. Figure 3 displays PR curves for model validation across various IoU thresholds. PR curves are common in binary classification tasks, like distinguishing between Pea and background, to evaluate model performance [29]. A higher area under the curve (AUC) indicates better accuracy and recall, with high precision signifying fewer false positives and high recall indicating fewer false negatives.

From Figures 3(a) to 3(d), it is evident that the VGG, ResNet50, and YOLOv3 models exhibited superior AUC values compared to the other models across all IoU thresholds. Conversely, the MobileNets, YOLOv5, and YOLOv4-tiny models achieved the minimum AUC values. It is worth noting that the YOLOv3 model remained relatively stable in terms of AUC as the IoU threshold varied. However, the MobileNets models demonstrated significant changes, particularly in precision, with varying IoU thresholds. Regarding recall, there were no significant differences observed among the models or across the different IoU thresholds.

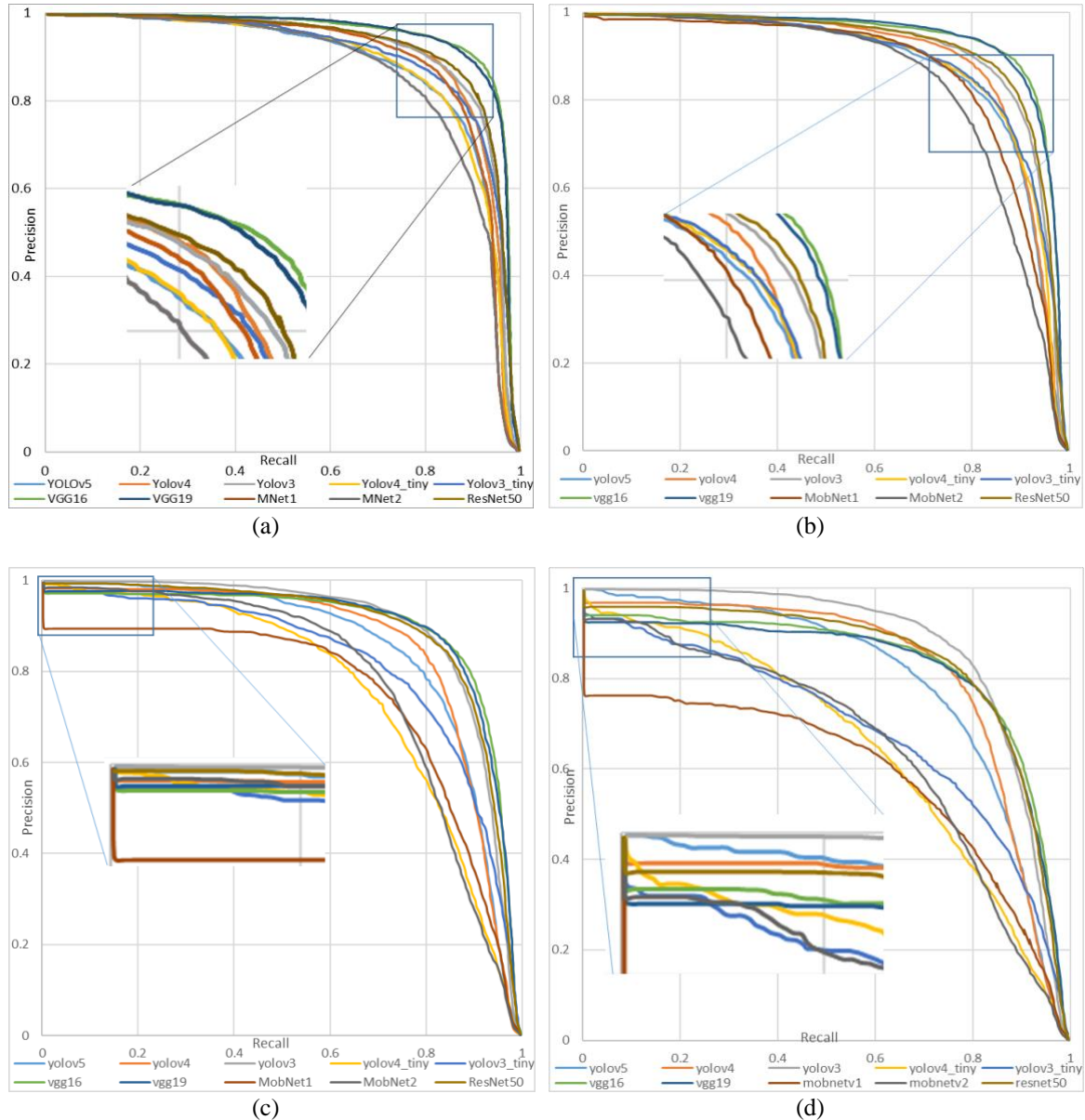


Figure 3. The PR_curves of the validation of the studied models with an IoU threshold: (a) 0.5, (b) 0.75, (c) 0.85, and (d) 0.9

In order to clarify the results of model validation, we used one of the metrics that better reflects the AUC is the average precision (AP). It is defined by (7) [3].

$$AP = \int_0^1 p(r) dr \quad (7)$$

The results of AP validation of the models studied in function of IoU threshold as well as the speed of inference (detection) for a single image are shown in Table 3.

The Table 3 further supports the findings depicted in the PR curves. The VGG, ResNet, and YOLO models exhibit the highest precision values and demonstrate stability in precision as the IoU threshold varies. However, these models tend to be slower in terms of execution time. On the other hand, YOLOv5 stands out as a fast model while maintaining good precision. The YOLO-tiny, MobileNets, and YOLOv5 models are identified as the fastest models, prioritizing computational efficiency.

Table 3. The validation results of the models studied as a function of IoU threshold

The model	AP ₅₀	AP ₇₅	AP ₈₅	AP ₉₀	Speed (ms)
YOLOv5	0.881	0.874	0.852	0.798	10
YOLOv4	0.906	0.890	0.859	0.818	29
YOLOv3	0.915	0.912	0.898	0.871	28.4
YOLOv4-Tiny	0.881	0.851	0.769	0.654	4.7
YOLOv3-Tiny	0.891	0.877	0.806	0.663	5.9
VGG16	0.946	0.935	0.898	0.830	32.5
VGG19	0.945	0.937	0.900	0.822	40.4
MobileNetv1	0.893	0.862	0.752	0.593	10.1
MobileNetv2	0.865	0.842	0.788	0.657	13.9
ResNet50	0.923	0.917	0.896	0.841	27.7
Faster RCNN ResNet 50	0.957	0.956	-	-	136
SSD-MobileNetv2	0.956	0.899	-	-	17

4. DISCUSSION

This section evaluates the results and limitations of our methods, focusing on identifying the best backbone to enhance YOLOv5's performance while preserving its other parameters (neck and head). We discovered that VGG, ResNet50, and YOLOv3 models, which demonstrated high precision and recall, are effective for achieving accurate results and capturing most positive instances. However, considering the constraints of low-capacity computing systems, prioritizing faster models like MobileNet, YOLO-tiny, and YOLOv5 is reasonable due to their good balance of speed and accuracy, making them suitable for deployment in resource-limited environments. For an IoU threshold of 0.5, MobileNetv1 achieved the highest AUC, reflecting strong performance in precision and recall. Yet, increasing the IoU threshold causes a drop in precision, as illustrated in Figures 4(a) and 4(b), where MobileNetv1's bounding box fails to accurately enclose the object compared to YOLOv5 [29].



Figure 4. Comparison between the prediction of (a) Mobilenetv1 and (b) YOLOv5 models

YOLOv5 stands out as the most stable among fast models, maintaining consistent average precision (AP) across varying IoU thresholds and accurately surrounding the pea crop. In contrast, MobileNetv1 excels at an IoU threshold of 0.5 but struggles with precise localization and suffers from occlusion errors, which can lead to parts of the crop being excluded from the bounding box and misclassified as weeds in subsequent segmentation. Therefore, when selecting a model, it is important to consider both detection capabilities and

stability across different IoU thresholds. Solutions to improve performance include expanding the bounding box to ensure complete coverage or modifying the model architecture to enhance results.

5. CONCLUSION

This study successfully utilized established CNNs to develop an object detector capable of accurately localizing crops amidst weeds in real-time images. Initial hypotheses were validated, and findings underscore the superiority of models based on VGG, ResNet, YOLOv3, and YOLOv4 in terms of accuracy, despite their high computational demands. However, for our application aiming to integrate this detector into an embedded system capable of simulating manual weed removal, speed is crucial. The fastest models like MobileNet, YOLO-tiny, and YOLOv5 performed well, with speeds ranging from 5 to 14 milliseconds per image. MobileNetv1 showed the best performance, achieving an AP of 89.3% for an IoU threshold of 0.5, though its performance decreases with higher IoU thresholds. Looking ahead, our focus will be on optimizing model architectures, particularly enhancing MobileNetv1 with advanced techniques to achieve even better results in terms of accuracy and speed.




REFERENCES

- [1] A. Tannouche, K. Sbair, M. Rahmoune, R. Agounoune, and A. Rahmani, "Real time weed detection using a boosted cascade of simple features," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 6, pp. 2755–2765, 2016, doi: 10.11591/ijece.v6i6.11878.
- [2] A. Tannouche *et al.*, "A fast and efficient shape descriptor for an advanced weed type classification approach," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 3, pp. 1168–1175, 2016, doi: 10.11591/ijece.v6i3.9978.
- [3] H. Mohammed, A. Tannouche, and Y. Ounejjar, "Weed detection in pea cultivation with the faster RCNN ResNet 50 convolutional neural network," *Revue d'Intelligence Artificielle*, vol. 36, no. 1, pp. 13–18, 2022, doi: 10.18280/ria.360102.
- [4] Y. Ounejjar, A. Tannouche, S. Sekhra, and M. Habib, "Optimisation of weed management by image segmentation in precision agriculture," *International Journal of Computational Vision and Robotics*, vol. 1, no. 1, 2024, doi: 10.1504/ijcvr.2024.10062366.
- [5] M. Habib, S. Sekhra, A. Tannouche, and Y. Ounejjar, "The identification of weeds and crops using the popular convolutional neural networks," *Digital Technologies and Applications*, pp. 484–493, 2023, doi: 10.1007/978-3-031-29857-8_49.
- [6] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Computers and Electronics in Agriculture*, vol. 158, pp. 226–240, 2019, doi: 10.1016/j.compag.2019.02.005.
- [7] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6738–6746, doi: 10.1109/CVPR.2017.713.
- [8] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018, doi: 10.1109/TMM.2017.2759508.
- [9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014, doi: 10.1109/CVPR.2014.223.
- [10] H. Su, X. Zhu, and S. Gong, "Deep learning logo detection with data expansion by synthesising context," *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pp. 530–539, 2017, doi: 10.1109/WACV.2017.65.
- [11] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-YOLO: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, 2020, doi: 10.3390/s20082238.
- [12] Z. Wang, L. Wu, T. Li, and P. Shi, "A smoke detection model based on improved YOLOv5," *Mathematics*, vol. 10, no. 7, 2022, doi: 10.3390/math10071190.
- [13] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: a lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155817.
- [14] H. Wang, S. Shang, D. Wang, X. He, K. Feng, and H. Zhu, "Plant disease detection and classification method based on the optimized lightweight YOLOv5 model," *Agriculture*, vol. 12, no. 7, 2022, doi: 10.3390/agriculture12070931.
- [15] X. Zhang, Y. Gao, H. Wang, and Q. Wang, "Improve YOLOv3 using dilated spatial pyramid module for multi-scale object detection," *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, 2020, doi: 10.1177/1729881420936062.
- [16] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for kiwifruit defects based on YOLOv5," *Electronics*, vol. 10, no. 14, 2021, doi: 10.3390/electronics10141711.
- [17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proceedings - 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking, ISPA/BDCLOUD/SustainCom/SocialCom 2019*, pp. 1500–1504, Dec. 2016, doi: 10.1109/ISPA-BDCLOUD-SustainCom-SocialCom48970.2019.00217.
- [19] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *arXiv-Computer Science*, pp. 1–6, 2018.
- [20] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: object detection and recognition using one stage improved model," in *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 2020, pp. 687–694, doi: 10.1109/ICACCS48705.2020.9074315.
- [21] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Computer Vision – ECCV 2014*, pp. 346–361, 2014, doi: 10.1007/978-3-319-10578-9_23.




- [23] Z. Jiang, L. Zhao, S. Li, and Y. Jia, "Real-time object detection method based on improved YOLOv4-tiny," *arXiv-Computer Science*, pp. 1-11.
- [24] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," *arXiv-Computer Science*, pp. 1-9, 2017.
- [25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [28] "BCE with logits loss," *PyTorch*, 2023. Accessed: Mar. 04, 2023. [Online]. Available: <https://pytorch.org/docs/master/generated/torch.nn.BCEWithLogitsLoss.html>
- [29] "Precision-recall," *Scikit Learn*, 2023. Accessed: Mar. 04, 2023. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

BIOGRAPHIES OF AUTHORS






Mohammed Habib    is currently a Ph.D. student at the Spectrometry, Materials and Archeomaterials Laboratory (LASMAR) Moulay Ismail University, Faculty of Sciences, Meknes, Morocco. His research interests are focused in computer vision, deep learning, and robotics and their application in agriculture. He can be contacted at email: moh.habib@edu.umi.ac.ma.






Salma Sekhra    is currently a Ph.D. student at the Spectrometry, Materials and Archeomaterials Laboratory (LASMAR), Moulay Ismail University, Faculty of Sciences, Meknes, Morocco. Her research interests are focused in computer vision, deep learning and their application in agri-food field. She can be contacted at email: sekhrasalma3@gmail.com.



Pr. Adil Tannouche    is a professor at the Higher School of Technology, Béni Mellal, Sultan Moulay Slimane University, Morocco. He received his Ph.D. in electronics and embedded systems from Moulay Ismail University, Meknes, Morocco. He is member of Laboratory of Engineering and Applied Technologies and his research area includes machine vision, artificial intelligence, and applications in the field of precision agriculture and agroindustry. He can be contacted at email: tannouche@gmail.com.



Pr. Youssef Ounejjar    was born in Meknes, Morocco, in 1971. He received the B.Eng. and M.S. degrees in electrical engineering from the Ecole Nationale d'Ingénieurs de Sfax, Sfax, Tunisia, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from the École de Technologie Supérieure, Montréal, QC, Canada, in 2011. His current research interests are the multilevel power converters. He is Associate Professor at the Moulay Ismail University, Meknes, Morocco. He can be contacted at email: ounejjar@gmail.com.