

Deep learning-based decision support system for weeds detection in wheat fields

Brahim Jabir, Nouredine Falih

Computer and Science Department, LIMATI Laboratory, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco

Article Info

Article history:

Received Feb 7, 2021

Revised Jun 16, 2021

Accepted Jul 1, 2021

Keywords:

Convolutional neural network

Decision-making

Deep learning

Precision farming

Raspberry

Weeds detection

ABSTRACT

In precision farming, identifying weeds is an essential first step in planning an integrated pest management program in cereals. By knowing the species present, we can learn about the types of herbicides to use to control them, especially in non-weeding crops where mechanical methods that are not effective (tillage, hand weeding, and hoeing and mowing). Therefore, using the deep learning based on convolutional neural network (CNN) will help to automatically identify weeds and then an intelligent system comes to achieve a localized spraying of the herbicides avoiding their large-scale use, preserving the environment. In this article we propose a smart system based on object detection models, implemented on a Raspberry, seek to identify the presence of relevant objects (weeds) in an area (wheat crop) in real time and classify those objects for decision support including spot spray with a chosen herbicide in accordance to the weed detected.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Brahim Jabir

LIMATI Laboratory, University of Sultan Moulay Slimane Beni Mellal

Mghila, BP 592 Beni Mellal, Morocco

Email: ibra.jabir@gmail.com

1. INTRODUCTION

Cereal crops play an important economic and social role in Morocco, as they constitute between 10 and 20 percent of the gross domestic product (GDP). It represents about 40 percent of the nutrition budget of Moroccan families, and it covers about 25 percent of the total needs of the herd [1]. Weeds compete with cereals for the use of water and nutrients. Some weeds can also serve as secondary hosts for various pathogens, nematodes or pests [2]. Agricultural technologies is a chance that can advance great governance of the agricultural policy, it can give answers for these issues and lift the financial development of our nation [3].

Weed detection remains a difficult problem due to variation in plant appearance, changes in lighting, foliage occlusions and different stages of growth under field conditions. Monocotyledon and dicotyledon are the two groups into which all the flowering plants or angiosperms were formerly divided [4]. Cereals and especially wheat are monocotyledon plants, therefore in wheat fields, dicotyledon weeds will be controlled easily and more effectively by special herbicides for dicotyledon plants but monocotyledon weeds (same type as wheat) require a selective herbicide specific for this weed to eliminate them by targeting them and spraying them [5]. Current approaches for weed and crop recognition, segmentation, and detection rely primarily on conventional machine learning techniques, which require a large number of hand-designed features for modelling [6]. Several studies have been completed on this topic and many weed detection algorithms are developed to lead the techniques to obtain high-precision weed removal. These algorithms contributed to the emergence of many robotic weed control systems that focuses principally on single tactics [7]. Among the existing systems, we can cite electrical discharging [8], chemical spraying [9], flaming, and

mechanical weeding [10]. These systems, despite their development, still suffer from some drawbacks, the most important of which are: consuming a lot of time, pollution, harming the crop.

Intelligent weed removal machines depend on the performance of the machine vision system to detect weeds, these systems mainly depend on deep learning, that enables the classification and location of weeds, the most used technique for this goal is convolutional neural network (CNN). This article discusses a CNN model implemented on an intelligent system to detect and locate types of weeds known in wheat crops in the region of Beni Mellal-Khenifra, Morocco. This smart system will help farmers to identify the species of weeds earlier, and classified it into monocots and dicots, to achieve a localized spraying to eliminate them and allow crops to grow freely benefiting from mineral salts and water. all that for the purpose of preserving the environment and reducing the cost of the operation.

2. RESEARCH METHOD

In our experiment we used a professional Nikon 7000 camera, to collect 1318 real images of wheat fields, to train a CNN model, under different lighting conditions (from morning to afternoon in sunny and cloudy weather). We have chosen two types of weed (monocotyledon and Dicotyledon) the most known in our region of Beni Mellal-Khenifra Figures 1 and 2. We have added the technical options to the images. The technical options that add random variations to generate more training data and improve the performance of the model [11]. The model used is the you only look once (YOLO) algorithm which is an efficient choice when we need real-time detection [12], without loss of too much accuracy and able to predict class labels and detects locations of weeds in order to take the better decision Table 1 [13]. Due to its high-speed inference, this model has the potential to be deployed on a single-board platform like the Raspberry Pi for weed detection and control it in real time through a set of features included in it. (Camera, Wifi...) [14].



Figure 1. Monocotyledon weed (Convolvulus)

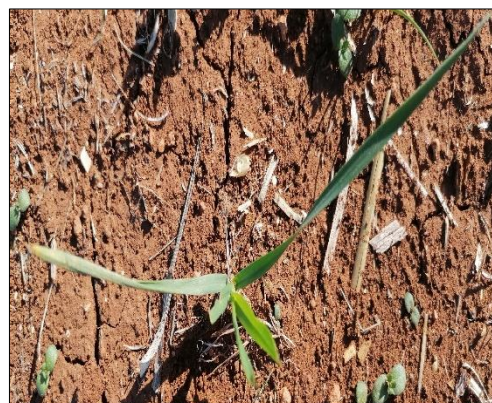


Figure 2. Dicotyledonous weed (Phalaris)

Table 1. Decision-making for controlling weeds

Crop affiliate group	Weed identified	Decision to make
Dicots	Monocotyledon	Spray a monocotyledon herbicide
	Dicotyledon (same type of culture; e.g convolovus)	Spray selective herbicide specifically targeting this dicotyledon weed.
Monocots	Monocotyledon (same type of culture; e.g phalaris in our case)	Spray selective herbicide specifically targeting this Monocotyledon weed.
	Dicotyledon	Spray dicotyledon herbicide

2.1. Object detection method

Object detection is one of the most popular computer vision models due to its versatility [15], it involves predicting the presence of one or more objects, along with their classes and bounding boxes [16]. YOLO is a state of art object detector which can perform object detection in real-time with good accuracy, YOLOv5 is a new release of the YOLO family of models that appeared on June 25, 2020. Used as one of the fastest algorithm that uses CNN for object detection combined bounding box prediction and object classification into a single end-to-end differentiable network, it can classify the image into a category, and can detect multiple objects within an image [17]. This algorithm applies a single neural network to the full image. It means that this network divides the image into regions and predicts bounding boxes and

probabilities for each region directly from full images in one evaluation. These bounding boxes are weighted by the predicted probabilities; YOLO is a faster and an accurate algorithm that is suitable for our operations. Therefore, we try to training the YOLO model and implement it on Raspberry Pi to guarantee a real time detection and identification, because this technological innovation opens up new possibilities for embedded internet of things (IoT) applications in fields such as the automatic weeding system that allows localized spraying with a chosen herbicide and with advanced analysis capability [18].

2.2. Loading and data preparation

As with any deep learning task, the first most important one is to prepare the dataset. Dataset is considered as the fuel, which runs any deep learning model [19]. The dataset for our experiment contains 1318 images of two different weed species well known in this region strained in wheat fields in this region known with its temperate climate. The first is the *Phalaris paradoxa*, the second is *Convolvulus* Table 2, these images are belonging to two different classes, and each class contains RGB images that show plants at different growth stages. The images are in various sizes and are in png format consequently, images had to be resized before being used as input to the CNN model to the shape 416×416 pixels [20].

Table 2. A list of the weeds classes available for the study

Scientific name	English name	Moroccan name	Affiliation group	Number of images
<i>Convolvulus arvensis</i> L	Convolvulus	Lwaya	Dicotyledon	904
<i>Phalaris paradoxa</i>	Phalaris	Zouane	Monocotolydone	401

The first thing we did is annotating those images which is a key technique used to create training data for computer vision. In order that the model perceives objects in their surroundings and automatically assign it a caption, annotated images are needed to train the model to learn to see an area full of objects as we do [21]. We used for that, bounding box technique that requires labellers to draw a box as close as possible to the edges of key objects within the image Figure 3, and stored the top-left and bottom-right points in the corresponding txt file respecting the syntax of *class_id x y width height* [22].



Figure 3. Bounding boxes example used

As the Figure 4 shows, there are three necessary sets of images to train the model, so we divided our images into the train, validation, and test splits to prevent the network model from overfitting issues and evaluate the model accurately [23]. The training set is the extensive part of our dataset (70% of our dataset) reserved for training the model. Inference resulted in these images after the training step will be taken to memorize the correct output. The validation set is a separate section of the dataset (20% of our dataset) that is used during training to evaluate the model performance reporting the validation metrics continually after each training epoch such as validation mean average precision (mAP), or validation loss. The test set (10% of our dataset) is used after the training experiments to get an idea of the final performance of the model.



Figure 4. The three basic sets for the model training

There are other preprocessing options, which are necessary steps to ensure that all our images are correctly formatted for the model and help it to detect edges better. This processing applies to all (train, valid, and test) images to reduce training time and improve inference speed [24], in our case we used a set of configuration as in the following examples: i) crop: crop each image to the specified section, such as the bottom third; ii) contrast: boosts contrast based on the image's histogram to improve normalization and line detection in varying lighting conditions; and iii) tile: split images into tiles to increase accuracy on small objects.

The last phase of data preparation is data-augmentation, which is a method applied to images in our training set to produce new and different training samples. This way applies domain-specific techniques to examples to generate more training data, and improve the model performance [25], below examples of some methods used in the experiment that gives us more than 3000 images after this operation Figure 5: i) exposure: this technique adds a degree of variability to image brightness to help the system be more resilient to lighting and camera setting changes; ii) flip: method that adds a vertical or horizontal flip to help the model be insensitive to subject orientation; iii) crop: method adds variability to positioning and size, helping the model be more resilient to subject translations and camera position.

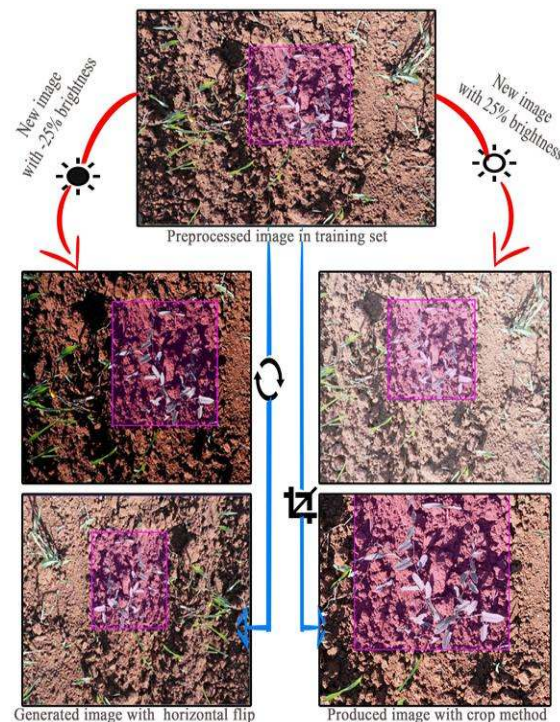


Figure 5. Set of data-augmentation methods applied

2.3. Deep neural network architecture

During our experiments, we used a YOLO model with some interventions on it. Thus, we reduced the number of detection scales to two scales, this model had in the scaling multipliers of the width 0.50 and depth 0.33 [26]. It is composed of a backbone part where our original image $416 \times 416 \times 3$ is input into the Focus structure. The slicing operation is used first to become a $208 \times 208 \times 12$ feature map, and then after a convolution operation of 32 convolution kernels the last change a feature map of $208 \times 208 \times 32$. Also, a cross-stage-partial (CSP) structure is designed in the backbone for feature extraction [27].

Furthermore, a neck part uses feature pyramid network (FPN) and path aggregation network (PAN) to make object detection predictions at different scale levels [28]. Finally, the model uses GIOU_Loss as the loss function of the bounding box in output [29]. Regarding the various parameters of different layers (283 layers) they have the same structure as the small YOLOv5 as it is mentioned in the following schema Figures 6 and 7.

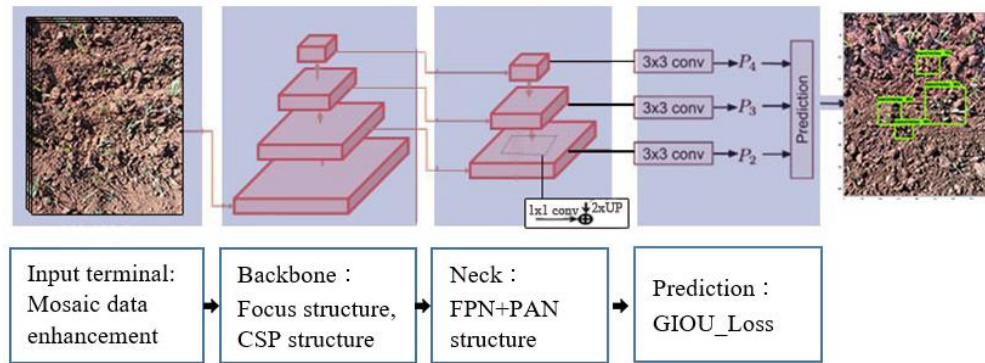


Figure 6. Deep neural network architecture

```

depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, BottleneckCSP, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, BottleneckCSP, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, BottleneckCSP, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]],
  [-1, 3, BottleneckCSP, [1024, False]], # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, BottleneckCSP, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, BottleneckCSP, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, BottleneckCSP, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, BottleneckCSP, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]

```

Figure 7. Model configuration

3. RESULT AND DISCUSSION

With prepared custom data, we trained the model configured above for 200 epochs which are hyperparameters of gradient descent that control the number of complete passes through the training dataset, that generate 7255094 parameters through 283 layers Figure 8. For evaluating the performance of the model, we employed several parameters and mean average precision (mAP) as one of the popular metrics in measuring the accuracy of object detectors defined with the formula [30]:

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q}$$

Q : The number of queries in the set.

AveP(q) : The average precision (AP) for a given query, q.

To fully evaluate the performance of the model, we need to analyze both precision and recall. The precision presents a proportion of positive identifications that were actually identified as correct (1), the recall is the measure of the model correctly identifying true positives (TP) (2). Precision and recall are

calculated using true positives (TP), false positives (FP) and false negatives (FN) [31], calculating those metrics for all the objects presented in the images helps to get mAP. Also getting the result on test set once a model is completely trained to help to check the performance of the model [32].

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	1	161152	models.common.BottleneckCSP	[128, 128, 3]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	1	641792	models.common.BottleneckCSP	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	656896	models.common.SPP	[512, 512, [5, 9, 13]]
9	-1	1	1248768	models.common.BottleneckCSP	[512, 512, 1, False]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	378624	models.common.BottleneckCSP	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	95104	models.common.BottleneckCSP	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	313088	models.common.BottleneckCSP	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1248768	models.common.BottleneckCSP	[512, 512, 1, False]
24	[17, 20, 23]	1	16182	models.yolo.Detect	[1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116,

Model Summary: 283 layers, 7255094 parameters, 7255094 gradients

Figure 8. Model summary

To display the results of the training, we used the visualization tool called TensorBoard. It allows us to visualize our TensorFlow graphs, to plot quantitative metrics on the execution of our graphs. The next section illustrates the results obtained in terms of precision recall and mean average precision.

As shown in the results, we tried 200 epochs, and we got good outcomes, the accuracy of the learning increases with the number of periods; this reflects that the model learns more information in each period. The model has arrived at 83% accuracy as shown in Figure 9, which means that the model achieves high precision. Moreover, the recall reaches 93% means that the model is efficient in identifying the relevant data as shown in Figure 10, this training returns a mAP of 94.4% at 0.5 shown in Figure 11. In Figure 12, the old school graphs also showed the same result in terms of recall, precision and mAP. The label predictions on the testing set images also show precise detection shown in Figure 13. Therefore, these experiments have demonstrated good achievement in training the CNN model to address weed classification in real-time.

Although we can be satisfied with the results to evaluate the model, we showed other the same metrics on some old school graphs plotted by python as shown in Figure 11. That showed the same results that we obtained on Tansorboard. In the Figure 12, we show model inference on the test images, where this evaluation is also commonly used to see the performance of the model.

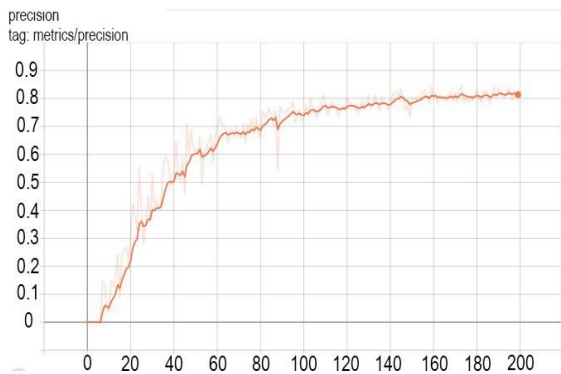


Figure 9. Model precision

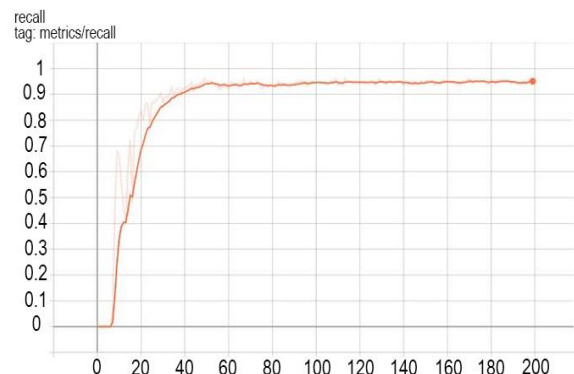


Figure 10. Model recall

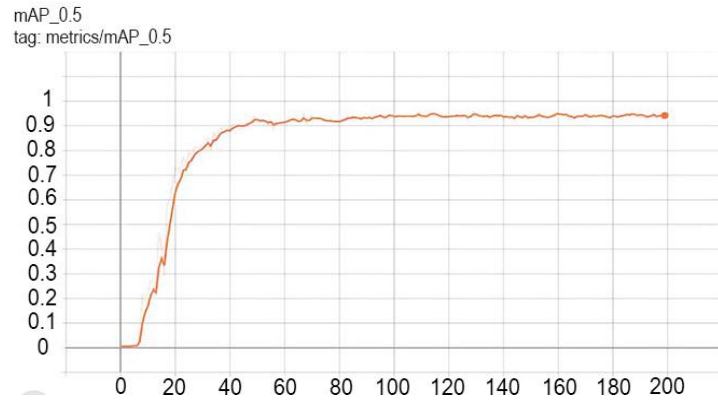


Figure 11. The mAP measured at 0.5

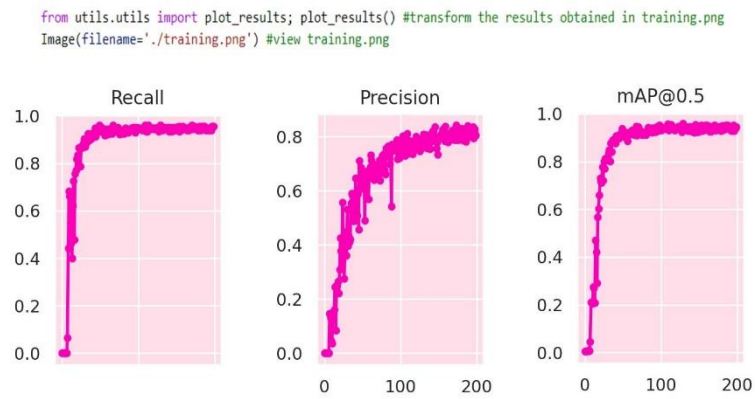


Figure 12. Model performance in our dataset in some older school graphs



Figure 13. Label predictions on test images

The main result of this study demonstrate the efficacy of YOLO and its high ability to identify weeds in a short and real time, the obtained accuracy above is very interesting, show that this model is accurate and fast. Which makes us adopted it in an intelligent automated real-time weed detection system in wheat crops to achieve localized spraying of these weeds with the correct herbicide instead of spraying the entire surface. The idea consists of deploying this deep learning model trained and saved in .h5 format on a Raspberry equipped with a camera which scan the field in motion to detect weeds in real-time [33]. The Raspberry sends a start order to a 12 V pump [34], which works alternately with other pumps, depends on the species of the weed detected. This system is also configured in a way that allows it to be controlled remotely from another computer or mobile device [35]. The Figure 14 clearly shows the synoptic of this spraying system.

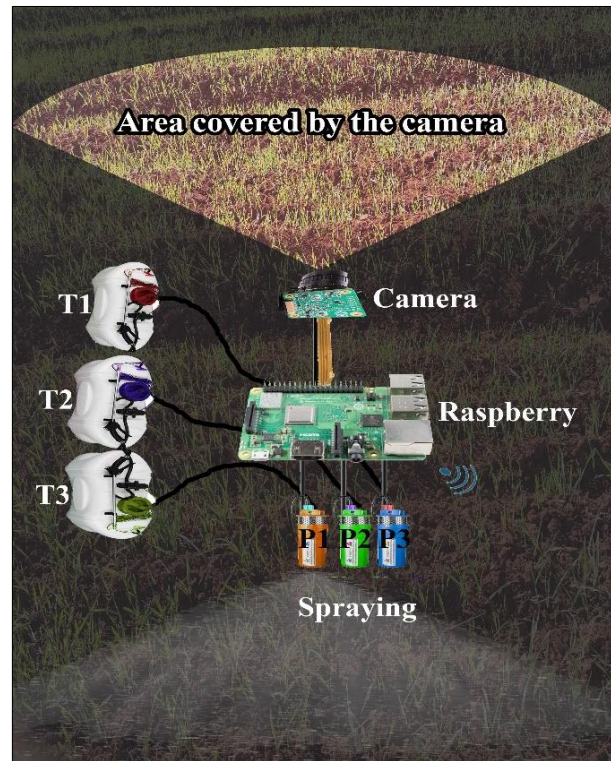


Figure 14. Overview of the spraying system

Where:

T1, T2, and T3 : Tanks of the herbicide A, B, C... depends on existing weed species in the field.

P1, P2, P3 : Pumps of the herbicide A, B, C.

The added value of the system we described above is that it can be considered as a smart system that will contribute to preserving the environment. Also the economical use of the herbicide, Quickly identify the type of weeds and determine the type of the appropriate herbicide in real-time, effectiveness in spraying the affected areas, it will produce a set of data that can be used in tracking and decision-making [36], it can be controlled remotely. In the end, we can also expand the list of unwanted weeds by reprogramming it after it becomes clear to us that this system is effective in the desired goal. We conclude with a brief comparison of our proposed weed control system with the existing systems mentioned in the literature review in the Table 3.

Table 2. The comparison of the proposed system with other existing methods

Criterion	Proposed weed control system	Existing systems [37]-[39]
Time	Real-time detection and control (fast)	Consume much time
Cost	Economical use of the herbicide	Too much herbicide is consumed
Control	Economical to establish	Mechanical control
Data	Controlled remotely	No data generated
Environment	Produce a set of data	Large scale spraying (affect the environment and crops)
Technology	Only spray the affected areas (Environmental protection)	Sometimes complex
	Simple	

4. CONCLUSION

We planned to work with a fast deep learning algorithm that allows us to identify weeds in real-time in a cultivated area and implemented an smart weeding system. In our experimentation, we implemented an open-source object detection model based on the latest version of YOLO, and we saw how it performs with the weed detection problem. The results showed better accuracy, this who encouraged us to adopt this model and implement it in a Raspberry based system which allows an automatic spray decision to be made. Instead of spraying one type of herbicide all over the plot, the system chooses the right herbicide depending on the weed detected by spraying only the areas infected with weeds, and this makes weeding more effective and saves herbicide and protects the environment. In the future, further steps may be taken to improve the performance of the system adjusting other parameters to be as a powerful real-time weeding system.




REFERENCES

- [1] T. Benabdelouahab, Y. Lebrini, A. Boudhar, R. Hadria, A. Htioui, and H. Lionboui, "Monitoring spatial variability and trends of wheat grain yield over the main cereal regions in Morocco: a remote-based tool for planning and adjusting policies," *Geocarto International*, pp. 1-20, 2019, doi: 10.1080/10106049.2019.1695960.
- [2] K. Singh, R. Rawat, and A. Ashu, "Image segmentation in agriculture crop and weed detection using image processing and deep learning techniques," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 5, pp. 235-238, 2021.
- [3] B. Jabir and N. Falih, "Digital agriculture in Morocco, opportunities and challenges," *2020 IEEE 6th International Conference on Optimization and Applications (ICOA)*, Beni Mellal, Morocco, 2020, pp. 1-5. doi: 10.1109/ICOA49421.2020.9094450.
- [4] V. de la Fuente, L. Rufo, N. Rodríguez, E. Ramírez, I. Sánchez-Gavilán, and R. Amils, "Differential iron management in monocotyledon and dicotyledon plants from the Río Tinto basin," *Protoplasma*, vol. 257, no. 3, pp. 889-900, 2020, doi: 10.1007/s00709-019-01476-x.
- [5] D. Todorova, I. Sergiev, Z. Katerova, E. Shopova, L. Dimitrova, and L. Brankova, "Assessment of the biochemical responses of wheat seedlings to soil drought after application of selective herbicide," *Plants*, vol. 10, no. 4, 2021, Art. no. 733, doi: 10.3390/plants10040733.
- [6] M. Rashid *et al.*, "A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection," *Sustainability*, vol. 12, no. 12, 2020, Art. no. 5037, doi: 10.3390/su12125037.
- [7] R. Raja, T. T. Nguyen, D. C. Slaughter, and S. A. Fennimore, "Real-time weed-crop classification and localisation technique for robotic weed control in lettuce," *Biosystems Engineering*, vol. 192, pp. 257-274, 2020, doi: 10.1016/j.biosystemseng.2020.02.002.
- [8] H. Sahin, "Comparative analysis of microwave and electric arc as a new method in weed control," *European Journal of Engineering and Technology Research*, vol. 6, no. 2, pp. 132-135, 2021, doi: 10.24018/ejers.2021.6.2.2369.
- [9] N. M. Devi, B. Mayanglambam, and B. Thangjam, "Biological control of weed," *Biotica Research Today*, vol. 2, no. 5, pp. 341-344, 2020.
- [10] M. Mainardis, F. Boscutti, M. del M. R. Cebolla, and G. Pergher, "Comparison between flaming, mowing and tillage weed control in the vineyard: Effects on plant community, diversity and abundance," *PLoS ONE*, vol. 15, no. 8, 2020, Art. no. e0238396, doi: 10.1371/journal.pone.0238396.
- [11] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2226-2234, doi: 10.1109/CVPR.2018.00237.
- [12] M. Y. Kamil, "A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 844-850, 2021, doi: 10.11591/ijece.v11i1.pp844-850.
- [13] P. M. Blok, F. K. Evert, A. P. M. Tielen, E. J. Henten, and G. Kootstra, "The effect of data augmentation and network simplification on the image-based detection of broccoli heads with Mask R-CNN," *Journal of Field Robotics*, vol. 38, no. 1, pp. 85-104, 2021, doi: 10.1002/rob.21975.
- [14] V. Czymmek, C. Möller, L. O. Harders, and S. Hussmann, "Deep learning approach for high energy efficient real-time detection of weeds in organic farming," *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2021, pp. 1-6, doi: 10.1109/I2MTC50364.2021.9459943.
- [15] N. V. Rao, D. V. Prasad, and M. Sugumaran, "Real-time video object detection and classification using hybrid texture feature extraction," *International Journal of Computers and Applications*, vol. 43, no. 2, pp. 119-126, 2021, doi: 10.1080/1206212X.2018.1525929.
- [16] F. Sun, H. Li, Z. Liu, X. Li, and Z. Wu, "Arbitrary-angle bounding box based location for object detection in remote sensing image," *European Journal of Remote Sensing*, vol. 54, no. 1, pp. 102-116, 2021, doi: 10.1080/22797254.2021.1880975.
- [17] M. I. Mardiyah and T. Purwaningsih, "Developing deep learning architecture for image classification using convolutional neural network (CNN) algorithm in forest and field images," *Science in Information Technology Letters (SITECH)*, vol. 1, no. 2, pp. 83-91, 2020, doi: 10.31763/sitech.v1i2.160.
- [18] N. L. Aravinda, M. Jabirullah, and D. Kirtana, "An Intelligent helmet system using IoT and Raspberry Pi," *IOP Conference Series: Materials Science and Engineering*, vol. 981, 2020, Art. no. 042076, doi: 10.1088/1757-899x/981/4/042076.
- [19] B. Jabir, N. Falih, A. Sarih, and A. Tannouche, "A strategic analytics using convolutional neural networks for weed identification in sugar beet fields," *AGRIS on-line Papers in Economics and Informatics*, vol. 13, no. 1, pp. 49-57, 2021, doi: 10.7160/aol.2021.130104.
- [20] M. Burić, M. Pobar, and M. Ivašić-Kos, "Adapting YOLO network for ball and player detection," *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, Prague, Czech Republic, 2019, pp. 845-851, doi: 10.5220/0007582008450851.
- [21] Md. A. Jishan, K. R. Mahmud, and A. K. A. Azad, "Natural language description of images using hybrid recurrent neural network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2932-2940, 2019, doi: 10.11591/ijece.v9i4.pp2932-2940.
- [22] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157-173, 2008, doi: 10.1007/s11263-007-0090-8.




- [23] A. R. Luaibi, T. M. Salman, and A. H. Miry, "Detection of citrus leaf diseases using a deep learning technique," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1719-1727, 2021, doi: 10.11591/ijece.v11i2.pp1719-1727.
- [24] H. Jeong, K. Park, and Y. Ha, "Image preprocessing for efficient training of YOLO deep learning networks," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 635-637, doi: 10.1109/BigComp.2018.00113.
- [25] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *2018 International Interdisciplinary PhD Workshop (IIPHDW)*, 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338.
- [26] F. Noort, C. H. Vaart, A. T. M. Grob, M. K. Waarsenburg, C. H. Slump, and M. Stralen, "Deep learning enables automatic quantitative assessment of puborectalis muscle and urogenital hiatus in plane of minimal hiatal dimensions," *Ultrasound Obstet Gynecol*, vol. 54, no. 2, pp. 270-275, 2019, doi: 10.1002/uog.20181.
- [27] I. Pacal and D. Karaboga, "A robust real-time deep learning based automatic polyp detection system," *Computers in Biology and Medicine*, vol. 134, 2021, Art. no. 104519, doi: 10.1016/j.compbiomed.2021.104519.
- [28] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask R-CNN With pyramid attention network for scene text detection," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 764-772, doi: 10.1109/WACV.2019.00086.
- [29] H. Rezaatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658-666, doi: 10.1109/CVPR.2019.00075.
- [30] S. Robertson, "A new interpretation of average precision," *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, Singapore, 2008, pp. 689-690, doi: 10.1145/1390334.1390453.
- [31] B. Juba and H. S. Le, "Precision-recall versus accuracy and the role of large data sets," *AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4039-4048, 2019, doi: 10.1609/aaai.v33i01.33014039.
- [32] A. C. Obula Reddy and K. Madhavi, "Convolutional recurrent neural network with template based representation for complex question answering," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2710-2718, Ju2020, doi: 10.11591/ijece.v10i3.pp2710-2718.
- [33] H.-Q. Nguyen, T. T. Kim Loan, B. Di. Mao, and E.-N. Huh, "Low cost real-time system monitoring using Raspberry Pi," *2015 Seventh International Conference on Ubiquitous and Future Networks*, Sapporo, Japan, 2015, pp. 857-859, doi: 10.1109/ICUFN.2015.7182665.
- [34] A. Imteaj, T. Rahman, M. K. Hossain, and S. Zaman, "IoT based autonomous percipient irrigation system using raspberry Pi," *2016 19th International Conference on Computer and Information Technology (ICCIT)*, 2016, pp. 563-568, doi: 10.1109/ICCITECHN.2016.7860260.
- [35] R. B. Anire, F. R. G. Cruz, and I. C. Agulto, "Environmental wireless sensor network using raspberry Pi 3 for greenhouse monitoring system," *2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2017, pp. 1-5, doi: 10.1109/HNICEM.2017.8269426.
- [36] B. Jabir, N. Falih, and K. Rahmani, "HR analytics a roadmap for decision making: case study," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 15, no. 2, pp. 979-990, 2019, doi: 10.11591/ijeecs.v15.i2.pp979-990.
- [37] S. A. Fennimore and D. J. Doohan, "The challenges of specialty crop weed control, future directions," *Weed Technology*, vol. 22, no 2, pp. 364-372, 2008, doi: 10.1614/WT-07-102.1.
- [38] B. Jabir, N. Falih, and K. Rahmani, "Accuracy and efficiency comparison of object detection open-source models," *International Journal of Online and Biomedical Engineering*, vol. 17, no. 05, 2021, Art. no. 165, doi: 10.3991/ijoe.v17i05.21833.
- [39] Y. Xiong, Y. Ge, Y. Liang, and S. Blackmore, "Development of a prototype robot and fast path-planning algorithm for static laser weeding," *Computers and Electronics in Agriculture*, vol. 142, pp. 494-503, 2017, doi: 10.1016/j.compag.2017.11.023.

BIOGRAPHIES OF AUTHORS



Brahim Jabir    received his Master degree in 2015 in computer engineering and systems at the Polydisciplinary Faculty of Sultan Moulay Slimane University in Beni Mellal, Morocco. Currently, He is a Ph.D. student in the Faculty of Sciences and Technics of the same University and he is working as a teacher of computer science in the regional centers of education and training professions in Beni Mellal, Morocco. His research interest is about Digital Agriculture, Deep learning, Strategic Analytics and Information Systems. He can be contacted at email: ibra.jabir@gmail.com.



Nouredine Falih    is PhD on Computer Science from Faculty of Sciences and Technologies of Mohammedia, Morocco in 2013. He is an associate professor in Polydisciplinary Faculty of Sultan Moulay Slimane University at Beni Mellal, Morocco since 2014. He has 18 years of professional experience in several renowned companies. His research topics are about Information System Governance, Business Intelligence, Big Data Analytics and Digital Agriculture. He can be contacted at email: nourfald@yahoo.fr.