Soft Computing (2024) 28:5029–5044
https://doi.org/10.1007/s00500-023-09212-7

APPLICATION OF SOFT COMPUTING

Automatic detection of weeds: synergy between EfficientNet and transfer learning to enhance the prediction accuracy

Linh T. Duong[1] · Toan B. Tran[1] · Nhi H. Le[1] · Vuong M. Ngo[2] · Phuong T. Nguyen[3]

Accepted: 5 September 2023 / Published online: 29 September 2023

Abstract

The application of digital technologies to facilitate farming activities has been on the rise in recent years. Among different tasks, the classification of weeds is a prerequisite for smart farming, and various techniques have been proposed to automatically detect weeds from images. However, many studies deal with weed images collected in the laboratory settings, and this might not be applicable to real-world scenarios. In this sense, there is still the need for robust classification systems that can be deployed in the field. In this work, we propose a practical solution to recognition of weeds exploiting two versions of EfficientNet as the recommendation engine. More importantly, to make the learning more effective, we also utilize different transfer learning strategies. The final aim is to build an expert system capable of accurately detecting weeds from lively captured images. We evaluate the approach's performance using DeepWeeds, a real-world dataset with 17,509 images. The experimental results show that the application of EfficientNet and transfer learning on the considered dataset substantially improves the overall prediction accuracy in various settings. Through the evaluation, we also demonstrate that the conceived tool outperforms various state-of-the-art baselines. We expect that the proposed framework can be installed in robots to work on rice fields in Vietnam, allowing farmers to find and eliminate weeds in an automatic manner.

1 Introduction

In recent years, there has been an increasing number of computer-aided applications deployed to help farmers perform their daily work (Rehman et al. 2019), attempting to deal with challenges caused by climate change. Robots equipped with object detection techniques have been used in various tasks (Dhaya et al. 2021), such as plant disease detection (Ampatzidis et al. 2017), pest classification (Thenmozhi and Reddy 2019), managing water resources (Reis et al. 2019), to name a few. Such tools help improve both effectiveness and efficiency of farming work (Pilarski et al. 2002; Lehnert et al. 2017; Bargoti and Underwood 2017) as they can fulfill these tasks in a short time and obtain a high accuracy (Deepa and Ganesan 2019).

Vietnam is a developing country, and despite the recent structural transformation process, agriculture still remains a staple element of the country's economy. Nevertheless, there is a worryingly increasing lack of labor in the agriculture sector (Sabzi et al. 2017; Liu et al. 2020), as descendants of farmers prefer to seek their fortune in big cities, rather than continuing the tradition. In fact, Vietnam is among the world's major rice exporters, and rice accounts for around 90% of the food production (Tam and Shimada 2019). However, due to the country's geographical position and weather conditions, rice fields in Vietnam severely suffer from the invasion of pets and, in particular, several types of wild weeds. The country has sought different countermeasures to improve crops and combat pets and weeds, including fertilizers and herbicides. Still, the effectiveness of herbicides is far from optimal as weeds quickly adapt and become resilient to

✉ Phuong T. Nguyen
phuong.nguyen@univaq.it
https://www.disim.univaq.it/ThanhPhuong

Linh T. Duong
duongtuanlinh@duytan.edu.vn

Toan B. Tran
tranbaotoan@duytan.edu.vn

Nhi H. Le
lehoangnhi@duytan.edu.vn

Vuong M. Ngo
vuong.nm@ou.edu.vn

[1] Institute of Research and Development, Duy Tan University, Danang, Vietnam

[2] Ho Chi Minh City Open University, Ho Chi Minh, Vietnam

[3] Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, L'Aquila, Italy

these means. Moreover, an abusive use of herbicides would cause contamination and harm to the environment. Given the circumstances, there is a need for automatic techniques and tools to deal with the lack of labor, as well as to support the detection and elimination of weeds. Having a sound technical background to recognize weeds is an important prerequisite to deploy in-field automatic robots.

The classification of weeds under realistic conditions turns out to be a daunting task. Although various studies have been conducted to recognize weeds and good results have been obtained, most of the existing studies deal with detection of weeds in a laboratory setting. As a matter of fact, the classification of weeds in rangeland environments has not received adequate attention (Olsen et al. 2019a). In reality, images captured from fields are heterogeneous and they contain noise in the background. In this respect, there is the need for robust mechanisms for detecting weeds *in situ*.

Machine learning (ML) has made profound progress in the past decade, thanks to the proliferation of several disruptive deep learning algorithms (Duong et al. 2023). There is a rise of applications exploiting ML across several domains. Among others, ML techniques have been used to solve different issues in the agriculture sector (Espejo-Garcia et al. 2021; Duong et al. 2020).

In this work, we propose a practical solution to weeds recognition, in terms of efficiency and effectiveness. We made use of two variants of the EfficientNet family (Tan and Le 2019) as the classification engine. More importantly, we incorporated different optimization functions and transfer learning strategies, aiming to find the best configuration. The performance of our approach has been evaluated using a real weed dataset, i.e., DeepWeeds (Olsen 2020). The results we got so far are promising: for all configurations, the obtained accuracy is always larger than 97%, with 99.62% being the maximum accuracy. Compared to two state-of-the-art baselines, our approach achieves a better prediction performance with respect to various quality metrics. The aim of our work is to build an expert system, which paves the way for a machine that is able to automatically detect and eliminate weeds from rice fields.

In this sense, our paper makes the following contributions:

- A practical solution to weed classification adopting cutting-edge deep neural network and transfer learning techniques.
- A comprehensive evaluation of the conceived framework on a real dataset. This also aims to compare it with two well-established baselines, namely ResNet-50 (Olsen et al. 2019a) and Inception-V3 (dos Santos Ferreira et al. 2019).
- A software prototype provided as a mobile app is ready for download.[1]

The paper is organized as follows. In Sect. 2, we provide a literature review on the related topics. Afterward, Sect. 3 presents in detail the proposed approach. In Sect. 4, we explain the dataset and metrics used for evaluation. Section 5 presents the main results and discussions, as well as the probable threats to validity of the outcome. Finally, we present future work and conclude the paper in Sect. 6.

2 Related work

In this section, we present a literature review on the related topics. In particular, we review work for automation in agriculture, and notable studies on weed classification.

Recently, different techniques have been conceived to solve issues in agriculture (Deepa and Ganesan 2019). A survey on studies exploiting Deep Learning in agriculture and food production has been recently conducted (Kamilaris et al. 2018). By means of a detailed examination on various agricultural problems, it has been shown that Deep Learning algorithms help obtain a better prediction accuracy and they outperform conventional image processing techniques. Similarly, Rehman et al. (2019) present a survey on machine learning techniques for various agricultural areas. The work summarizes the pros and cons of statistical machine learning techniques for certain purposes. Furthermore, it also provides a discussions on future trends of statistical machine learning technology applications.

Convolutional neural network models (Ferentinos 2018) have been applied to help farmers detect plant diseases. The approach has been evaluated using an open database with 25 different plants. The framework achieved a high prediction accuracy, and this suggests that the model can be used to support an identification system working in real-world scenarios. Our approach presented in this paper is relevant to this work, as we also attempt to assist farmers in their daily tasks, using Deep Learning techniques. We suppose that EfficientNet can also be used to automatically recognize plant diseases.

A survey (Zhao et al. 2016) presented the major techniques used in fruit or vegetable harvesting robots, and provided discussions on the challenges and trends of deploying different automatic techniques in robots. The exists an approach to identification of soybean leaves and herbivorous pest from images captured by unmanned aerial vehicles (Amorim et al. 2019). The system aims to support specialists and farmers in pest control management in soybean fields, especially when there is a limited amount of labeled instances. We assume that our approach build on top of EfficientNet and transfer learning can be used to tackle the issue (Amorim et al. 2019).

The automatic identification of tree species have great potential in agriculture, and there is a tree species recognition method based on the fusion of multiple deep learning

[1] https://github.com/linhduongtuan/DeepWeeds_Classifier_WebApp.

models (Hu et al. 2018). The dataset was built based on the published image datasets on the Internet and autonomous photography. The experimental results showed that the recognition accuracy of the tree species in the complex background with the proposed method reached 93.75%. A recent work (Sharpe et al. 2020) leveraged the advantages of tiny-You Only Look Once 3 (YOLOv3-tiny) as a potential detector to aid goosegrass identification and spraying in situ. The approach was evaluated by two annotation techniques, i.e., annotation of the entire plant (EP) and annotation of partial sections of the leaf blade. The reported performance is as follows: F_1-score 0.75 and 0.85 for the EP and LB of goosegrass detection in strawberry.

To assist farmers in their daily tasks, a number of approaches have been conceived to classify fruits classification, making use of various machine learning techniques. For instance, a feature learning-based algorithm has been used to build a systems for classifying fruits (Hung et al. 2015). The algorithm extracts the most representative features of fruit images by means of pixel classification. Similarly, a deep neural network to detect fruit was conceived (Sa et al. 2016), aiming to support yield estimation and automated harvesting. A multi-modal Faster R-CNN model (Ren et al. 2015) was designed and implemented, and it achieves improvement in accuracy compared to various state-of-the-art approaches. Furthermore, the approach is also timing efficient, as it requires bounding box annotation instead of pixel-level annotation. The model was retrained to detect seven fruits, and the entire process takes 4 h to annotate and train the new model for a fruit. In our recent work (Duong et al. 2020), we proposed a workable solution to fruit classification using EfficientNet and MixNet . The experimental results show that our proposed approach obtains a high prediction accuracy, thereby outperforming a well-established baseline.

Recently, several studies have focused on boosting productivity in agriculture. In a recent work (Olsen et al. 2019b), the authors proposed an approach to recognize various weed species in the complicated range land environment. The work introduced a baseline for weeds classification task using deep learning model, Inception-v3 and ResNet-50 on the DeepWeeds dataset including 17,509 images of eight weed species across northern Australia. The proposed framework gained an average classification accuracy of 95.1% and 95.7%.

A deep learning architecture named Graph Weeds Net (GWN) (Hu et al. 2020) has been recently introduced to detect multiple types of weeds from images. The proposed method improved the performance of weed identification tasks by establishing fine-grained level deep representation and specified the high probability of the true-positive weed rather than the others. The problem of plant recognition is still challenging due to background noise from the living environment. The proposed technique (Zhu et al. 2019) is able to recognize plants on four datasets, i.e., Malayakew, ICL, Flowers 102, and CFH plant, by exploiting the two-way attention model with deep convolutional neural network. The approach obtained accuracy of 99.8%, 99.9%, 97.2%, and 79.5% for the aforementioned datasets. The first attention way is based on taxonomy of plants to recognize plant's family, whereas the second attention way focuses on the discriminative features of plants image based on the feature maps generated by convolutional neural networks. Due to the compatibility of two-way attention, the discriminative feature learning and part-based attention are combined and they obtained promising results.

3 A practical solution to weeds classification

This section introduces the proposed approach based on EfficientNet (Tan and Le 2019), EfficientNet -Lite4, and transfer learning (Huang et al. 2017; Weiss et al. 2016). We systematically present the related technologies, i.e., EfficientNet in Sect. 3.1, and transfer learning in Sect. 3.2. The architecture conceptualized to build an expert system to automatically recognize weeds is presented in Sect. 3.3.

3.1 EfficientNet

EfficientNet (Tan and Le 2019) is a recently developed family of deep neural network, with the aim of transcending the main limitations of the existing CNN technologies related to prediction accuracy. EfficientNet imposes a balance between all network dimensions, i.e., width, depth, and resolution by means of a set of fixed scaling coefficients that meet some specific constraints (Tan and Le 2019). The EfficientNet family is made of different versions, and the most simple one is EfficientNet -B0. The other EfficientNet configurations are generated from EfficientNet -B0 with different scaling values. For illustration purposes only, the most compact configuration, i.e., EfficientNet -B0, is shown in Fig. 1: It consists of 18 convolution layers in total and each of them uses a kernel either k(3,3) or k(5,5). Input images are made of three color channels, i.e., R, G, B, and each of them is scaled to the size of 224 × 224. The next layers are reduced in resolution, but increased in width to enhance accuracy. For example, the second convolutional layer is equipped with **W**=16 filters, and by its next layer, the number of filters is **W**=24. The maximum number of filters is **D**=1, 280 by the last layer which is fed to the fully connected layer.

Similar to the EfficientNet backbone, the EfficientNet -Lite models[2] are designed for working on mobile CPU, GPU, and EdgeTPU, but still maintain a comparable accuracy compared to quantized version of some popular image classification models. EfficientNet-Lite architectures do not apply

[2] https://bit.ly/3nBUzGo.

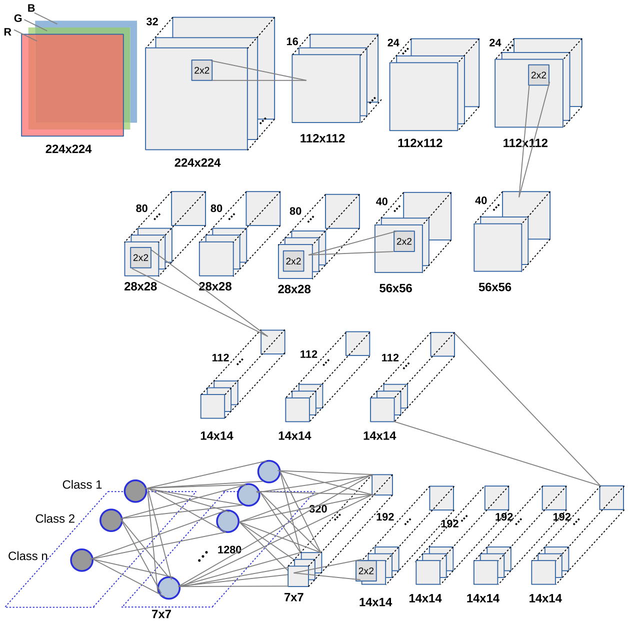


Fig. 1 The efficientNet-B0 architecture

squeeze-and-excite function like the original backbone, but use the modification of Rectified Linear Unit, namely RELU6 function, instead of Sigmoid Linear Units (SiLU). We opted for EfficientNet -Lite4, the largest variant, which achieved 80.4% ImageNet top-1 accuracy, while still running in real time, e.g., 30ms/image on a Pixel 4 CPU. Within a CNN, different optimization functions can be used to speed up the learning process. In the scope of this paper, three different optimization functions, namely, SGD, ADAM, and SLS, are incorporated into our evaluation.

3.2 Transfer learning

An important requirement for deep neural networks is to acquire enough data for the training process. Nevertheless, labeled data can be obtained by manual annotation, which requires time and human labor. Thus, transfer learning has been adopted as a practical solution to overcome the lack of data, since it allows for the re-use of weights and biases trained using large datasets. With respect to using only random weights, transfer learning brings a better performance in terms of effectiveness and efficiency, even when the target domain is quite different from the original one where the weights are obtained (Huang et al. 2017). In the scope of this work, we consider three learning strategies as follows.

- **ImageNet**: We import pre-trained weights from the ImageNet dataset (Russakovsky et al. 2015), which consists of more than 14 million images, covering various categories;

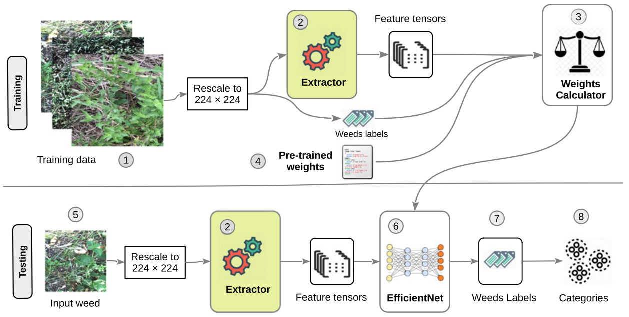


Fig. 2 System architecture

- **Adversarial propagation (AdvProp)** (Xie et al. 2019): Adversarial propagation has been proposed as an enhanced training scheme, and it treats adversarial examples as additional examples, with the ultimate aim of minimizing overfitting;
- **Noisy Student (NS)** (Xie et al. 2019): This aims at boosting up ImageNet classification Noisy Student training by: (i) extending the trainee/student equal to or larger than the trainer/teacher, so as to force the trainee learn better on a large dataset, and (ii) adding noise to the student, thus enabling it to learn more.

3.3 Architecture

The conceived system is shown in Fig. 2 and there are two main phases, namely training and testing. The architecture allows for the inclusion of weights pre-trained using other datasets, e.g., ImageNet (Russakovsky et al. 2015).

Input images ① are already labeled and they are transformed into a feature vector, which is then handled by the `Extractor` component ②. During training, an input image is augmented with rotation and one resized crop, as well as horizontal flipping with a random change. The rotation is done to change at least 30% of the input image. The resulting image is then rescaled to fit into a frame of size 224 × 224, and fed as training data. The feature vectors and their labels are used to train the system by means of the `Weights Calculator` component ③. Pre-trained weights ④ from other datasets can be imported to deploy transfer learning.

The resulting trained parameters can then be used to classify unlabeled weeds images. The testing phase (or deployment) can be performed on lightweight devices, e.g., laptops or smartphones. Each time, when an unknown input weed image ⑤ is put into the system, it will be extracted to build a feature vector using `Feature Extractor` ②. The feature tensor is then fed to the neural networks ⑥ which run the classification engine to label the input image ⑦. The final result is a label that classifies the input weed image into a certain category ⑧.

4 Evaluation materials and methods

In this section, we explain in detail the experiments to evaluate the proposed approach as well as to compare it with two existing studies. First, the research questions to study the systems' characteristics are enumerated in Sect. 4.1. Section 4.2 gives an overview of the DeepWeeds dataset (Olsen 2020), which has been exploited as input data for the evaluation. The experimental settings are described in Sect. 4.3, while Sect. 4.4 presents the metrics utilized to measure the prediction performance.

4.1 Research questions

By performing a series of experiments on the given dataset, we attempt to answer three research questions pertinent to this work:

- **RQ**$_1$: *Which optimization function brings a better classification performance for EfficientNet -B4?* We examine different EfficientNet -B4 configurations to find the one that obtains the best prediction accuracy.
- **RQ**$_2$: *Which optimization function brings a better classification performance for EfficientNet -B4 Lite?* Similarly, we perform experiments on the EfficientNet -Lite4 network, aiming to determine the setting that brings the best prediction accuracy on the given dataset.
- **RQ**$_3$: *How does the proposed approach perform compared to the baselines?* Finally, we are interested in understanding if our proposed approach achieves a better prediction performance compared to two state-of-the-art baselines (Olsen et al. 2019a; dos Santos Ferreira et al. 2019).

4.2 Dataset and baselines

The DeepWeeds dataset was collected using a ground-based weed control robot (Olsen 2020). There are 17,509 images in total with nine categories of weeds, which are summarized in Fig. 3. Among them, eight categories are real weeds, while the Negative category is made of different images rather than weeds, i.e., soil and vegetation. Figure 4a–i depicts some representative examples of the categories from the original dataset. By plain eyes, it is difficult to distinguish the weeds from each other. In this respect, an automatic mechanism to detect weeds is highly desirable.

To better study the performance of our proposed approach in relation to state-of-the-art studies, we compare EfficientNet -B4 and EfficientNet -Lite with two baselines as follows. The first one is built based on ResNet-50 , a convolutional neural network originally designed and implemented by the creators of the DeepWeeds dataset to articulate their contributions. The tool works on top of the TensorFlow framework,[3] and its source code is available online (Olsen 2020). The second baseline is a tool developed by *dos Santos Ferreira et al.* (dos Santos Ferreira et al. 2019), which is based on the Inception-V3 deep neural network.

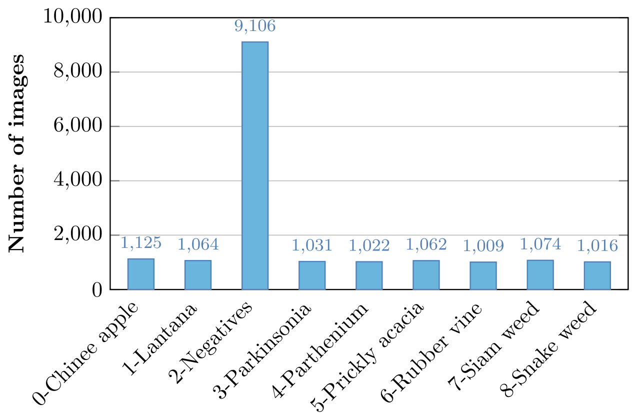


Fig. 3 A summary of the DeepWeeds dataset (Olsen 2020)

4.3 Experimental settings

For all the experiments, we split the original dataset into three independent parts, i.e., 80% for training, 10% for validation, and 10% for testing. Moreover, we applied various data augmentation techniques as follows. Images are transformed using rotate, blur, random noise, horizontal flip, and vertical flip exploiting an existing tool.[4] The parameters and their corresponding values are listed in Table 1. The ultimate aim of the augmentation process is to enrich the dataset to feed the recommendation engine.

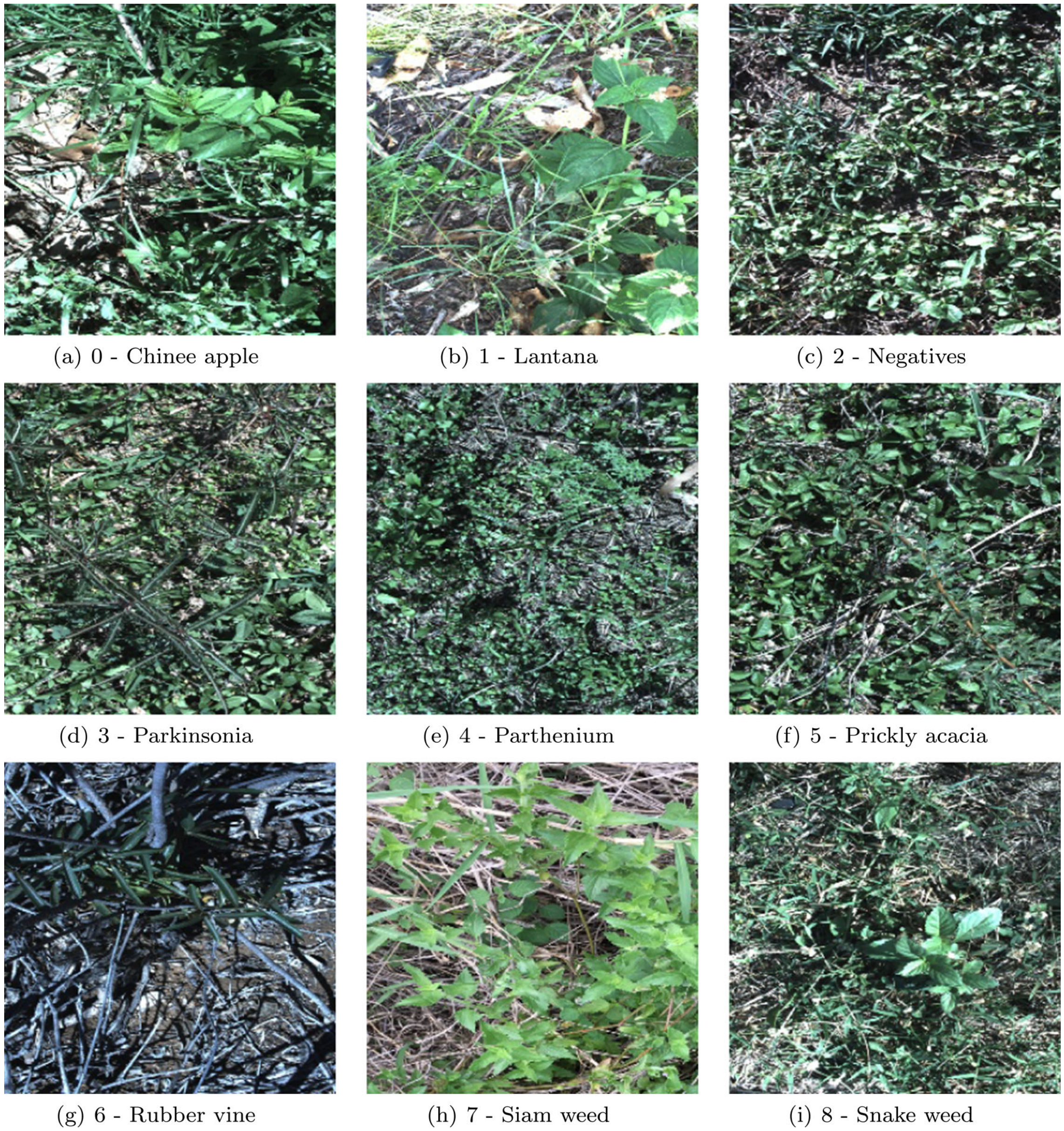
We adopted a recent implementation (Wightman 2019a) of EfficientNet and EfficientNet -Lite4 which was built on top of the PyTorch framework.[5] We also imported pre-trained weights from the ImageNet dataset (Russakovsky et al. 2015) as well as using NS and AdvProp (cf. Sect. 3.2), with the final aim of speeding up and increasing the effectiveness of the learning process. We trained on a server with the following configurations: Intel® Xeon® CPU E5-2678v3 @ 2.50GHz × 12 cores, 96GiB RAM, NVIDIA GeForce GTX 1080Ti, Operating System Ubuntu 20.04.1 LTS.

Table 2 depicts the experimental configurations for EfficientNet -B4, EfficientNet-Lite4 as well as for the two baselines, i.e., ResNet-50 and Inception-V3 . The *# Params* column corresponds to the number of parameters needed to store the weights and biases of the network. The *# MAC* column measures the computational complexity, counted as

[3] https://www.tensorflow.org/.

[4] https://github.com/tomahim/py-image-dataset-generator.

[5] https://pytorch.org.



(a) 0 - Chinee apple

(b) 1 - Lantana

(c) 2 - Negatives

(d) 3 - Parkinsonia

(e) 4 - Parthenium

(f) 5 - Prickly acacia

(g) 6 - Rubber vine

(h) 7 - Siam weed

(i) 8 - Snake weed

Fig. 4 Real weed images (extracted from the original dataset (Olsen 2020))

Table 1 Augmentation parameters

No	Parameter	Description	Value
0	DEFAULT_ROTATE_PROBABILITY	The probability to rotate	0.5
1	DEFAULT_ROTATE_MAX_LEFT_DEGREE	The maximum left rotating degree	25
2	DEFAULT_ROTATE_MAX_RIGHT_DEGREE	The maximum right rotating degree	25
3	DEFAULT_BLUR_PROBABILITY	The probability to blur an image	0.1
4	DEFAULT_RANDOM_NOISE_PROBABILITY	The probability to add random noise	0.5
5	DEFAULT_HORIZONTAL_FLIP_PROBABILITY	The probability to flip horizontally	0.3
6	DEFAULT_VERTICAL_FLIP_PROBABILITY	The probability to flip vertically	0.3

Table 2 Experimental configurations

	No	# Params (M)	# MAC (B)	File size (MB)	Optimizer	Learning
EfficientNet -B4	1	17.57	2.10	141.4	SGD	ImageNet
	2					Noisy student
	3					AdvProp
	4			211.8	ADAM	ImageNet
	5					Noisy student
	6					AdvProp
	7			71.0	SLS	ImageNet
	8					Noisy student
	9					AdvProp
Lite4	10	11.74	1.32	94.6	SGD	ImageNet
	11			141.5	ADAM	
	12			47.6	SLS	
RN-50		23.61	3.88	277.2	ADAM	ImageNet
IN-V3		21.8	2.85	256.8	ADAM	ImageNet

the number of multiply-accumulate operation. While both EfficientNet -B4 and EfficientNet -Lite4 need less than 20 millions of parameters and 2.10G MAC, ResNet-50 and Inception-V3 are larger in size, i.e., they have more than 21.80 millions of parameters and 2.85G MAC. Correspondingly, the *File size* used to store the parameters (in MB) varies depending on the number of parameters. We also use three different optimizers in our evaluation, i.e., SGD, ADAM, and SLS. For EfficientNet -B4, we combine three transfer learning strategies introduce in Sect. 3.2, i.e., ImageNet, NS, and AdvProp. While for EfficientNet -Lite4, we can use only weights pre-trained from the ImageNet dataset.

4.4 Evaluation metrics

Given a set of weeds images, we have their corresponding labels, i.e., $G = (G_1, G_2, .., G_N)$. By running a classifier, we obtain a set of predicted labels, i.e., $C = (C_1, C_2, .., C_N)$. We compare the two sets to measure the prediction performance, exploiting the following metrics: *accuracy*, *precision*, *recall*, and F_1 *score*, which are defined as follows.

Accuracy: The metric is measured as the ratio of number of correct prediction to the total number items

$$accuracy = \frac{\sum_i^N match_i}{\sum_i^N |G_i|} \times 100\%. \tag{1}$$

Precision and Recall: *Precision* asserts the number correctly predicted instances, whereas *recall* expresses the ability to find all relevant instances in the dataset. The metrics are computed using the following formula:

$$precision_i = \frac{match_i}{|C_i|} \tag{2}$$

$$recall_i = \frac{match_i}{|G_i|}. \tag{3}$$

F_1score (F-Measure): F_1-score is computed as the harmonic mean of Precision and Recall as follows:

$$F_1 = \frac{2 \cdot precision_i \cdot recall_i}{precision_i + recall_i}. \tag{4}$$

In the next section, we analyze the experimental results by answering the research questions in Sect. 4.1.

5 Experimental results

We report and analyze the results obtained by conducting a series of experiments on the given dataset in Sects. 5.1, 5.2, and 5.3. Afterward, Sect. 5.5 discusses the probable threats that may adversely impact on the validity of our findings.

5.1 RQ$_1$: *Which optimization function brings a better classification performance for EfficientNet -B4?*

We study the performance by calculating precision, recall, and F_1 scores for EfficientNet -B4, exploiting three optimization functions, i.e., SGD, ADAM, and SLS. The obtained results are shown in Table 3. To facilitate the reading, we compare the performance category-wise, and use different colors to mark the maximum values for each metric. In particular, the light red, light gray, and light green colors are used to highlight the maximum precision, maximum recall, and maximum F_1, respectively.

The EfficientNet family uses either SGD, ADAM, or SLS (Vaswani et al. 2019) as the loss function. The number of epochs for the training phase is 120 iterations. Moreover,

Table 3 Precision, Recall, and F1-score using EfficientNet -B4 with different optimizers

			Categories								
			Chinee apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
SGD	**P**	IM	0.978	0.981	0.998	0.990	0.990	0.959	0.952	0.934	0.976
		AP	0.982	0.981	1.000	0.990	0.995	0.977	0.975	0.986	0.990
		NS	0.969	0.986	1.000	0.995	0.995	0.972	0.975	0.995	0.990
	R	IM	0.986	0.995	0.973	1.000	0.995	1.000	1.000	0.995	1.000
		AP	1.000	1.000	0.985	1.000	1.000	1.000	1.000	1.000	1.000
		NS	1.000	1.000	0.985	1.000	1.000	1.000	1.000	1.000	1.000
	F_1	IM	0.982	0.988	0.986	0.995	0.992	0.979	0.975	0.964	0.987
		AP	0.991	0.990	0.992	0.995	0.997	0.988	0.987	0.993	0.995
		NS	0.984	0.993	0.992	0.997	0.997	0.986	0.987	0.997	0.995
ADAM	**P**	IM	0.965	0.981	0.996	0.985	0.980	0.950	0.957	0.934	0.957
		NS	0.944	0.981	0.996	0.981	0.990	0.968	0.966	0.950	0.971
		AP	0.924	0.968	0.996	0.981	0.976	0.958	0.966	0.968	0.943
	R	IM	0.986	0.976	0.968	1.000	1.000	0.995	0.995	1.000	0.995
		NS	0.991	0.990	0.970	1.000	1.000	0.995	1.000	0.990	0.995
		AP	0.982	1.000	0.964	1.000	1.000	0.985	1.000	0.990	0.995
	F_1	IM	0.975	0.978	0.982	0.992	0.990	0.972	0.975	0.966	0.975
		AP	0.952	0.983	0.980	0.990	0.987	0.972	0.983	0.979	0.968
		NS	0.967	0.986	0.983	0.990	0.995	0.981	0.983	0.970	0.983
SLS	**P**	IM	0.945	0.990	0.998	1.000	0.995	0.955	0.975	0.995	0.985
		AP	0.965	0.986	0.998	0.995	0.995	0.963	0.957	0.981	0.990
		NS	0.970	1.000	0.999	0.985	0.985	0.959	0.980	0.977	0.985
	R	IM	1.000	0.990	0.981	1.000	1.000	1.000	1.000	0.995	0.995
		AP	1.000	1.000	0.980	1.000	1.000	0.990	1.000	0.995	1.000
		NS	1.000	1.000	0.981	1.000	1.000	1.000	0.995	1.000	1.000
	F_1	IM	0.971	0.990	0.990	1.000	0.997	0.977	0.987	0.995	0.990
		AP	0.982	0.993	0.989	0.997	0.997	0.976	0.978	0.988	0.995
		NS	0.986	1.000	0.990	0.992	0.992	0.979	0.987	0.988	0.992

a start learning rate of 0.001 with rate decay of 10^{-1} after every 30 epochs is set for SGD and ADAM. In fact, SLS does not use learning rate, and its number of batch sizes is smaller than that of SGD and ADAM. Although SLS helps to converge faster in terms of numbers of training epochs, the total time for training is not significantly reduced due to the number of batch size settings and SLS seems not to achieve a saturated prediction performance on the independent test set. Due to these reasons, the file sizes of trained weights using SLS optimizer are smaller than those of using either SGD or ADAM (see Table 2). Though SLS optimization does not obtain the best performance, it is still useful to build a baseline for further experimental benchmark.

Overall, the table demonstrates that using the ADAM optimization function yields a mediocre performance, compared to its counterparts. To be concrete, we can see that ADAM enables to achieve the maximum recall score only for some categories, e.g., R=1.00 for Category Parkinsonia, Parthenium, Prickly Acacia; Meanwhile, it does not help bring any maximum values for Precision and F_1. In contrast, compared to ADAM, the SLS optimizer contributes to a much better prediction, i.e., it allows EfficientNet -B4 to yield a good performance in terms of Recall for many categories. Moreover, using SLS also is also beneficial to the Precision and F_1 scores.

In comparison to the others, the SGD optimization function brings the best performance with respect to all the evaluation metrics. As we can see in the table, most of the maximum Precision scores are obtained using SGD, i.e., there are several cells marked with the light red color, demonstrating the most superior values. Similarly, for other metrics, i.e., Recall and F_1, SGD demonstrates its superiority by helping EfficientNet -B4 achieve the best predictions, compared to ADAM and SLS. Especially, by Recall, we can see that SGD enables EfficientNet -B4 to yield a maximum performance by many categories.

Altogether, the results in RQ_1 reveals an interesting outcome as follows. The ADAM optimizer has been widely used in various classification tasks; however, as we show in this evaluation, at least for weed classification, it cannot gain the upper hand compared to the other functions.

> **Answer to RQ_1.** Training EfficientNet -B4 with the SGD optimizer brings the best prediction performance on the given dataset.

5.2 RQ_2: *Which optimization function brings a better classification performance for EfficientNet -B4 Lite?*

The precision, recall, and F1 scores obtained using EfficientNet -Lite4 are reported in Table 4. Similar to RQ_1, we also use the same set of colors to highlight the maximum values for the quality metrics, i.e., Precision, Recall, and F_1.

The table shows that training EfficientNet -Lite4 with the ADAM optimization function brings a mediocre performance, compared to using SGD and SLS. Running EfficientNet -Lite 4 with SLS is beneficial to the final classification, as this configuration brings a good performance in terms of Precision and Recall for various categories. Overall, since most of the colored cells are related to SGD, we conclude that it is the most suitable optimization for this network configuration, since it brings the maximum values for all the evaluation metrics. In particular, with respect to recall, the combination of EfficientNet -Lite4 and SGD yields the maximum score, i.e., 1.00 for eight among nine weed categories. Concerning the F_1 metric, SGD also helps EfficientNet -Lite4 get the maximum value by all the categories.

To further study the performance of the SGD function, we show in Fig. 5a–d the confusion matrices for running EfficientNet and EfficientNet -Lite4 with various transfer learning strategies. The figures show that training EfficientNet -B4 with NS and AdvProp brings a superior performance, compared to other configurations. To be concrete, by both configurations, we obtain correct predictions for all the weed categories, i.e., accuracy is equal to 1.00. Only by the Negatives category, there are some miss predictions. Altogether, this demonstrates that using the SGD optimizer is beneficial to the final outcome.

> **Answer to RQ_2.** The SGD optimization function is also beneficial to weed classification as it helps EfficientNet -Lite4 obtain the best prediction performance on the given dataset.

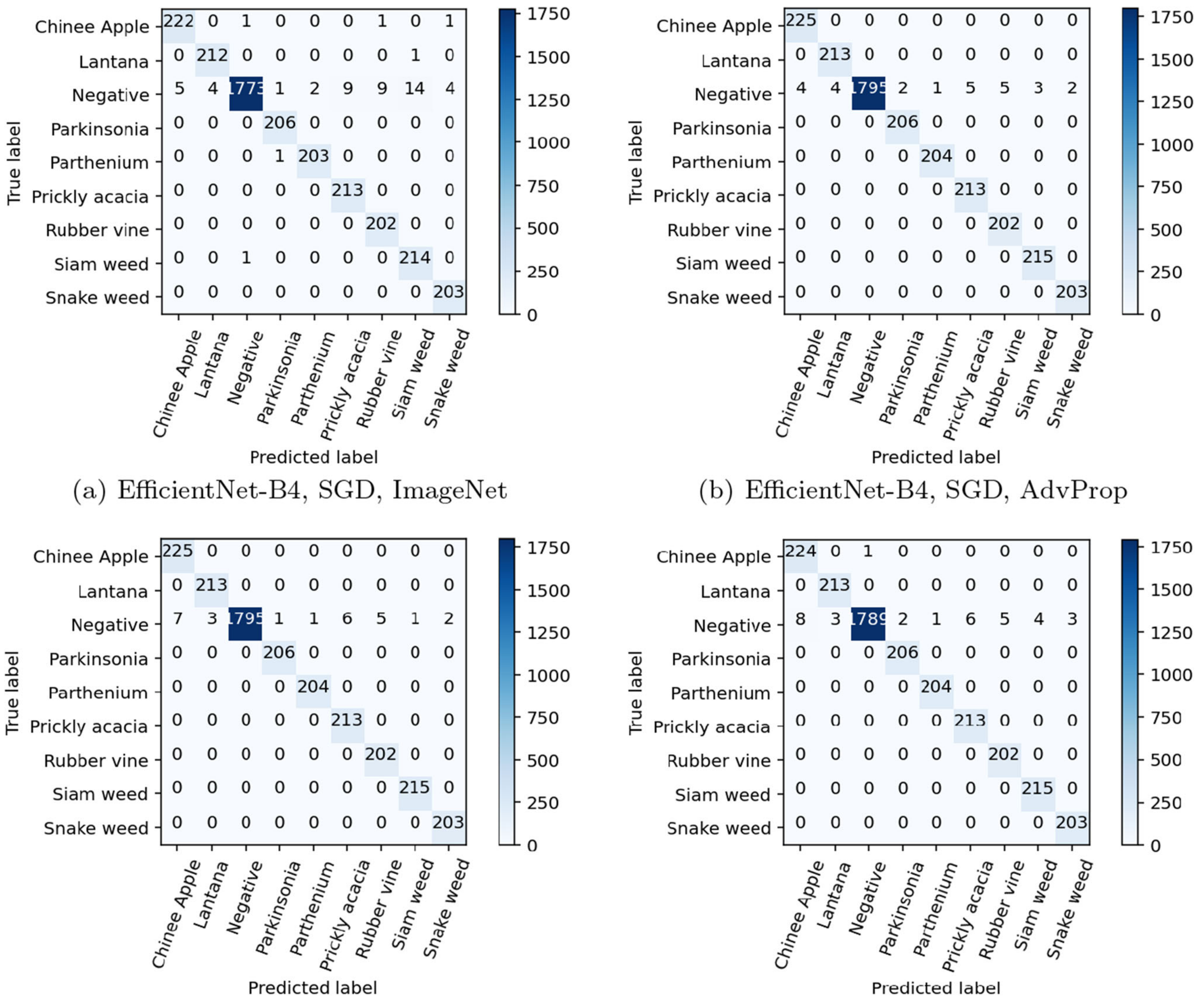
5.3 RQ_3: *How does the proposed approach perform compared to the baselines?*

Transfer learning has shown to bring benefits to the classification with deep neural networks (Thenmozhi and Reddy 2019; Duong et al. 2020). In this work, we investigate if the weights pre-trained obtained by various learning algorithms are beneficial to the recognition of weeds. We ran the networks, i.e., EfficientNet -B4, EfficientNet -Lite4, ResNet-50 and Inception-V3 on the given dataset and obtained a set of categories as results. We computed the evaluation metrics for each of the systems following the descriptions in Sect. 4.4. Table 5 reports the accuracy obtained by all the systems.

In relation to the two baselines, our approach gains a much better performance in terms of accuracy. To be concrete, the first one (Olsen et al. 2019a) obtained 96.08% with the best configuration using ResNet-50 , while the second baseline (dos Santos Ferreira et al. 2019) obtained a maximum accuracy of 94.96% using the VGG16 deep neural network. In particular, using EfficientNet -B4 with the SGD function and weights pre-trained with AdvProp and Noisy Student, we get 99.26% as the prediction accuracy, the maximum score compared to others.

Table 4 Precision, Recall, and F1-score using EfficientNet -Lite4 with different optimizers

		Categories								
		Chinee apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
SGD	**P**	0.965	0.986	0.999	0.990	0.995	0.972	0.975	0.981	0. 985
	R	0.995	1.000	0.982	1.000	1.000	1.000	1.000	1.000	1.000
	F_1	0.980	0.993	0.990	0.995	0.997	0.986	0.987	0.990	0.992
ADAM	**P**	0.902	0.963	0.994	0.976	0.975	0.950	0.935	0.972	0.961
	R	0.986	0.990	0.958	**1.000**	0.990	0.981	**1.000**	0.995	0.985
	F_1	0.942	0.976	0.976	0.988	0.983	0.965	0.966	0.983	0.973
SLS	**P**	0.965	0.986	0.997	0.990	0.995	0.963	0.975	0.986	0.976
	R	0.986	1.000	0.980	1.000	0.995	1.000	1.000	1.000	1.000
	F_1	0.9758	0.993	0.989	0.995	0.995	0.981	0.987	0.993	0.987



True label \ Predicted label	Chinee Apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
Chinee Apple	222	0	1	0	0	0	1	0	1
Lantana	0	212	0	0	0	0	0	1	0
Negative	5	4	1773	1	2	9	9	14	4
Parkinsonia	0	0	0	206	0	0	0	0	0
Parthenium	0	0	0	1	203	0	0	0	0
Prickly acacia	0	0	0	0	0	213	0	0	0
Rubber vine	0	0	0	0	0	0	202	0	0
Siam weed	0	0	1	0	0	0	0	214	0
Snake weed	0	0	0	0	0	0	0	0	203

(a) EfficientNet-B4, SGD, ImageNet

True label \ Predicted label	Chinee Apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
Chinee Apple	225	0	0	0	0	0	0	0	0
Lantana	0	213	0	0	0	0	0	0	0
Negative	4	4	1795	2	1	5	5	3	2
Parkinsonia	0	0	0	206	0	0	0	0	0
Parthenium	0	0	0	0	204	0	0	0	0
Prickly acacia	0	0	0	0	0	213	0	0	0
Rubber vine	0	0	0	0	0	0	202	0	0
Siam weed	0	0	0	0	0	0	0	215	0
Snake weed	0	0	0	0	0	0	0	0	203

(b) EfficientNet-B4, SGD, AdvProp

True label \ Predicted label	Chinee Apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
Chinee Apple	225	0	0	0	0	0	0	0	0
Lantana	0	213	0	0	0	0	0	0	0
Negative	7	3	1795	1	1	6	5	1	2
Parkinsonia	0	0	0	206	0	0	0	0	0
Parthenium	0	0	0	0	204	0	0	0	0
Prickly acacia	0	0	0	0	0	213	0	0	0
Rubber vine	0	0	0	0	0	0	202	0	0
Siam weed	0	0	0	0	0	0	0	215	0
Snake weed	0	0	0	0	0	0	0	0	203

(c) EfficientNet-B4, SGD, NS

True label \ Predicted label	Chinee Apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
Chinee Apple	224	0	1	0	0	0	0	0	0
Lantana	0	213	0	0	0	0	0	0	0
Negative	8	3	1789	2	1	6	5	4	3
Parkinsonia	0	0	0	206	0	0	0	0	0
Parthenium	0	0	0	0	204	0	0	0	0
Prickly acacia	0	0	0	0	0	213	0	0	0
Rubber vine	0	0	0	0	0	0	202	0	0
Siam weed	0	0	0	0	0	0	0	215	0
Snake weed	0	0	0	0	0	0	0	0	203

(d) Lite4, SGD, ImageNet

Fig. 5 Confusion matrices EfficientNet -B4 and EfficientNet -Lite4 with SGD

Table 5 Accuracy for EfficientNet -B4, EfficientNet -Lite4, ResNet-50 , and Inception-V3 (%)

Network	Optimizer	Transfer learning	Accuracy (%)
EfficientNet -B4	SGD	ImageNet	98.46
EfficientNet -B4	SGD	AdvProp	**99.26**
EfficientNet -B4	SGD	Noisy student	**99.26**
EfficientNet -B4	ADAM	ImageNet	98.03
EfficientNet -B4	ADAM	AdvProp	97.86
EfficientNet -B4	ADAM	Noisy student	98.26
EfficientNet -B4	SLS	ImageNet	98.94
EfficientNet -B4	SLS	AdvProp	98.89
EfficientNet -B4	SLS	Noisy student	99.03
EfficientNet -Lite4	SGD	ImageNet	99.06
EfficientNet -Lite4	ADAM	ImageNet	97.40
EfficientNet -Lite4	SLS	ImageNet	98.89
RN-50 (Olsen et al. 2019a)	ADAM	ImageNet	96.08
IN-V3 (dos Santos Ferreira et al. 2019)	ADAM	ImageNet	94.96

Table 6 Precision, Recall, and F1-score (using the ADAM optimization function)

		Categories								
		Chinee apple	Lantana	Negative	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	Siam weed	Snake weed
B4	**P**	0.965	0.981	0.996	0.985	0.980	0.950	0.957	0.934	0.957
	R	0.986	0.976	0.968	1.000	1.000	0.995	0.995	1.000	0.995
	F_1	0.975	0.978	0.982	0.992	0.990	0.972	0.975	0.966	0.975
Lite4	**P**	0.902	0.963	0.994	0.976	0.975	0.950	0.935	0.972	0.961
	R	0.986	0.990	0.958	1.000	0.990	0.981	1.000	0.995	0.985
	F_1	0.942	0.976	0.976	0.988	0.983	0.965	0.966	0.983	0.973
RN-50	**P**	0.932	0.954	0.973	0.979	0.972	0.896	0.985	0.936	0.923
	R	0.863	0.973	0.971	0.986	0.972	0.967	0.951	0.967	0.917
	F_1	0.896	0.964	0.972	0.983	0.972	0.930	0.968	0.951	0.920
IN-V3	**P**	0.950	0.958	0.972	0.993	0.916	0.916	1.000	0.838	0.864
	R	0.832	0.921	0.966	0.959	0.972	0.934	0.923	0.986	0.924
	F_1	0.887	0.939	0.969	0.975	0.943	0.925	0.960	0.906	0.893

Table 7 Timing performance

Configuration	# of images	Inference time (milliseconds)	Speed (images/second)
EfficientNet-B4	21,375	33,668	635
EfficientNet_Lite4	21,375	27,321	782
ConvNext_Atto	21,375	21,495	**994**
ConvNext_Femto	21,375	21,517	**993**
ConvNext_Pico	21,375	21,559	991
ConvNext_Nano	21,375	21,668	987
CoatNet_Nano_RW_224	21,375	27,338	782
GCViT_XXTiny	21,375	30,828	693
ResNet-50	21,375	25,581	836
Inception-V3	21,375	21,928	975

Table 8 Summary of recent state-of-the-art algorithms

Configuration	# of Param. (M)	# of MAC	Trained weight size (MB)	Acc. on ImageNet-1K (%)	Acc. on DeepWeeds test set (%)
ConvNeXt_Atto	3.70	551.16M	13.5	75.7	98.66
ConvNeXt_Femto	5.22	784.22M	19.4	77.5	99.11
ConvNeXt_Pico	9.05	1.37G	32.4	79.5	99.00
ConvNeXt_Nano	15.59	2.45G	59.9	80.9	99.11
CoAtNet_Nano_RW_224	15.00	2.28G	58.6	81.7	97.72
GCViT_XXTiny	12	1.94G	46.0	79.8	99.34

To further compare the approaches, from the predicted categories, we calculated the precision, recall, and F_1 scores following Eqs. (2), (3), and (4), respectively. The final results are shown in Table 6 and we also use the same colors in RQ_1 to mark the maximum values. From the table, it is evident that EfficientNet -B4 is the best classifier among others, as most of the colored cells are concentrated on the rows representing the evaluation metrics for this configuration. Both baselines obtain an inferior performance compared to EfficientNet -B4 and EfficientNet -Lite4 with respect to precision, recall, and F_1.

For instance, Inception-V3 gets two maximum precision scores for Category 3 and 6, and ResNet-50 gains only one maximum recall value. In contrast, we can see that most of the best predictions are achieved using EfficientNet -B4. Also, by EfficientNet -Lite4, there are maximum scores for precision, recall, and F_1.

Table 7 depicts the timing performance of the proposed approach in comparison with the baselines. Among others, ResNet-50 is far from optimal as it can predict the results for 836 images per second. Inception-V3 is quite efficient, as it can generate predictions for 975 images in 1 s. However, ConvNext_Femto and ConvNext_Atto are more efficient compared to Inception-V3 as they return 993 and 994 predictions within a second, respectively.

Answer to RQ_3. Our proposed approach considerably outperforms the two baselines in all the considered test configurations with respect to both prediction accuracy and timing efficiency.

5.4 Ablation study

To study the generalizability of our approach, we conduct more experiments using state-of-the-art backbones including the ConvNeXt family (Liu et al. 2022), CoAtNet (Dai et al. 2021), and GCViT (Hatamizadeh et al. 2022) on the DeepWeeds dataset. An Nvidia 1080Ti is not suitable for real deployment because of its energy consumption. We carried out more experiments on edge and very light architectures, including ConvNeXt_Atto, ConvNeXt_Femto, and this allows us to bring more choices to deploy the research result on devices with limited computational resources.

The supplement experiments are implemented using the latest version of the timm codebase (Wightman 2019b).[6] These configurations are summarized in Table 8. For all the model settings, we use a transfer learning strategy with trained weights from ImageNet-1K and the SGD optimization function. The experimental results of the models on the independent test set are shown in Fig. 6. Considering the ConvNeXt family, the most optimistic performance belongs to the ConvNeXt_Femto (see Fig. 6b), while it obtains the accuracy of 99.11%, which is equal to an accuracy of ConvNetXt_Nano (see Fig. 6d), but smaller than the ConvNeXt_Nano in terms of hyperparameter numbers, MAC, file size of trained weight. The equal accuracy seems to be saturated prediction performance using ConvNeXt variant.

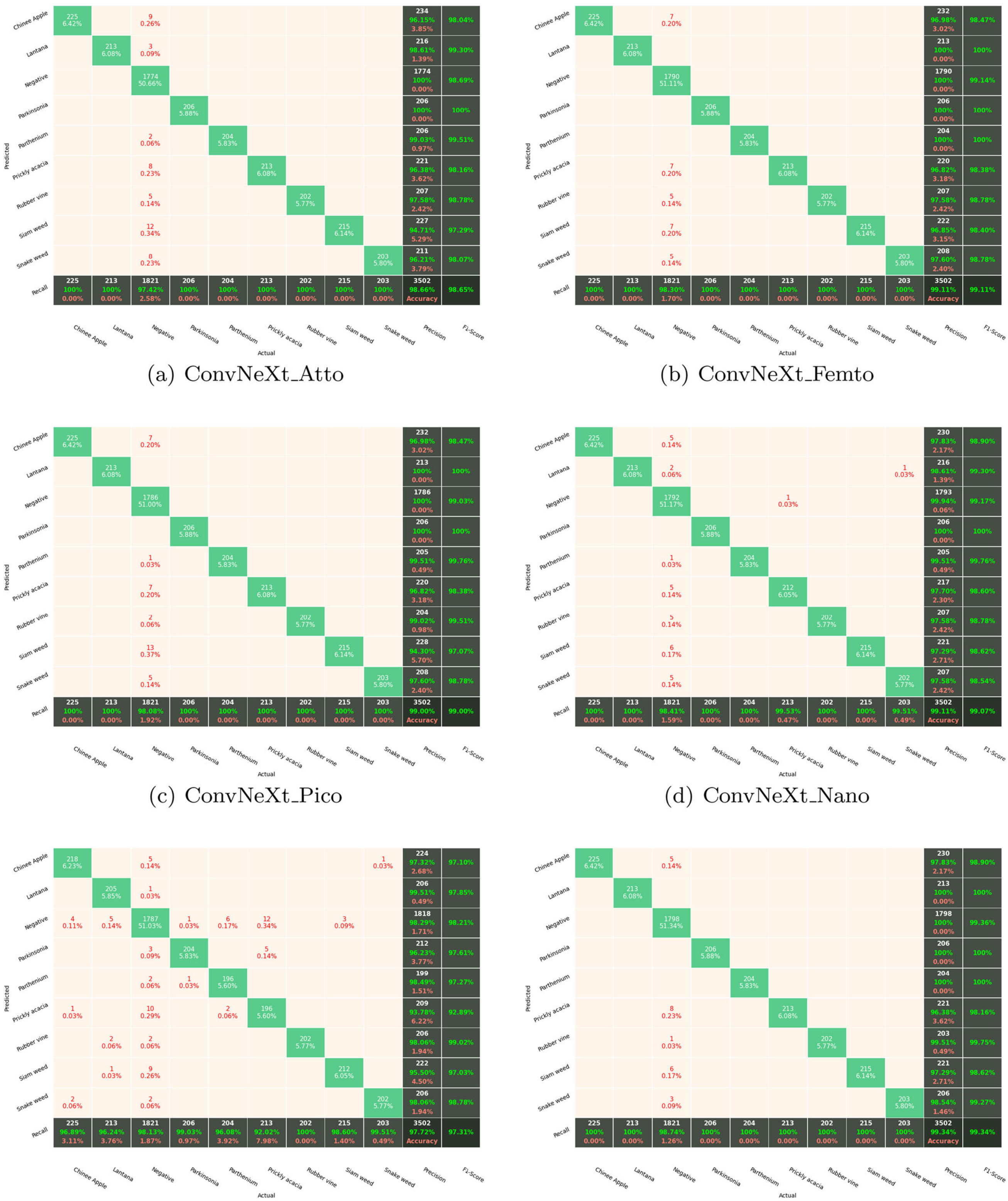
Obviously, training and inference cost for ConvNeXt_Femto are cheaper than that of ConvNeXt_Nano. ConvNeXt_Pico obtains the accuracy of 99.00%, while the smallest architecture ConvNeXt_Atto brings an accuracy of 98.66% (see Fig. 6a). Furthermore, GCViT_XXTiny achieves the maximum accuracy of 99.34% on the independent test set (see Fig. 6f. As shown in Fig. 6e, the second biggest model CoAtNet_Nano_RW_224 gives a baseline accuracy of 97.72%. This suggests that the synergy of architectures and typical datasets needs to be further investigated in future work.

5.5 Threats to validity

This section explains the probable threats to internal and external validity of our evaluation as follows.

Internal validity. These are internal factors that could influence the evaluation. A probable threat is the comparison with the baselines. Such a threat is minimized, since we ran experiments using the original implementations, as well as executed

[6] We gratefully acknowledge the models on top of original backbones, including ConvNeXt_Atto, ConvNeXt_Femto, ConvNeXt_Pico, and ConvNeXt_Nano, CoAtNet_Nano_RW_224, and GCViT_XXTiny built by Ross Wightman.



(a) ConvNeXt_Atto

(b) ConvNeXt_Femto

(c) ConvNeXt_Pico

(d) ConvNeXt_Nano

(e) CoAtNet_Nano_RW_224

(f) GCViT_XXTiny

Fig. 6 Confusion matrices of recently state-of-the-art models on the independent test set of DeepWeeds

the three systems on the same dataset, and compare them using the same set of metrics.

External validity. The threat to *external validity* of the approach is related to the generalizability of our findings, i.e., whether they would still be valid outside the scope of this study. We moderated the threat by evaluating EfficientNet and EfficientNet -Lite4 using a dataset collected in situ, which covers different categories of weeds. This aims at evaluating the feasibility of the approach in real-world settings.

Construct validity. The threats are pertinent to the experimental configurations presented in the paper, with respect to the simulated scenario to evaluate the system. We conducted the evaluation using a training set and a test set, which might not reflect well a real-world usage. To aim for a reliable comparison, we used the same settings to evaluate and compare the systems.

Conclusion validity. This is related to the factors that may influence the obtained outcome. The evaluation metrics, namely accuracy, precision, recall, and F_1 score, might pose a threat to conclusion validity. To tackle the issue, we used the same metrics for comparing our proposed approach with the two baselines.

6 Conclusions and future work

In this paper, we proposed a solution to weeds classification exploiting EfficientNet -B4 and EfficientNet -Lite4 as the engine, together with various transfer learning techniques. Our proposed approach has been validated on a real-world dataset. Through the empirical evaluation, we see that the conceived framework outperforms two well-established baselines in terms of various quality metrics. In this respect, we concluded that the combination of EfficientNet and transfer learning brings a substantial improvement in performance compared to using ResNet-50 and Inception-V3. Interestingly, we found out that the ADAM optimizer function, which has been widely used in deep neural networks, does not gain the upper hand compared to the SLS and SGD functions. We come to the conclusion that the SGD optimizer function is the most suitable one for EfficientNet, when it comes to weed classification. Altogether, we contribute to an advancement in the classification of weeds in real-world scenarios.

We plan to deploy the system on a robot to detect weeds from lively captured images. Such a robot is beneficial to farmers in Vietnam as it helps them automatically find and eliminate weeds on rice fields. We are also going to extend the evaluation by considering additional weed datasets, as well as compare EfficientNet and EfficientNet -Lite4 with other neural network approaches. Last but not least, we will select the most network model and export to Android, making it an independent tool suitable to work on a smartphone.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped us improve the paper.

Funding Open access funding provided by Universitá degli Studi dell'Aquila within the CRUI-CARE Agreement. The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest All the authors declare that they have no conflict of interest. Furthermore, they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Amorim W P, Tetila E C, Pistori H, Papa J P (2019) Semi-supervised learning with convolutional neural networks for uav images automatic recognition. Comput Electron Agric 164:104932, 2019. ISSN 0168-1699. https://doi.org/10.1016/j.compag.2019.104932. http://www.sciencedirect.com/science/article/pii/S0168169919305137

Ampatzidis Y, De Bellis L, Luvisi A (2017) ipathology: Robotic applications and management of plants and plant diseases. Sustainability 9(6):2071–1050. https://doi.org/10.3390/su9061010. **(ISSN)**

Bargoti S, Underwood J (2017) Deep fruit detection in orchards. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 3626–3633, Singapore, Singapore, May 2017. IEEE. ISBN 978-1-5090-4633-1. https://doi.org/10.1109/ICRA.2017.7989417

Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: Marrying convolution and attention for all data sizes. Adv Neural Inf Process Syst 34:3965–3977

Deepa N, Ganesan K (2019) Hybrid rough fuzzy soft classifier based multi-class classification model for agriculture crop selection. Soft Comput 23(21):10793–10809. https://doi.org/10.1007/s00500-018-3633-8. **(1433-7479. ISSN)**

Dhaya R, Kanthavel R, Ahilan A (2021) Developing an energy-efficient ubiquitous agriculture mobile sensor network-based threshold built-in MAC routing protocol (TBMP). Soft Comput 25(18):12333–12342. https://doi.org/10.1007/s00500-021-05927-7. **(ISSN 1433-7479.)**

dos Santos Ferreira A, Freitas D M, da Silva G G, Pistori H, Folhes M. T. (2019) Unsupervised deep learning and semi-automatic data labeling in weed discrimination. Comput Electron Agric 165:104963, 0168-1699. ISSN https://doi.org/10.1016/j.compag.2019.104963. URL http://www.sciencedirect.com/science/article/pii/S0168169919313237

Duong L T, Nguyen P T, Di Sipio C, Di Ruscio D (2020) Automated fruit recognition using efficientnet and mixnet. Comput Electron Agric 171:105326, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2020.105326. URL http://www.sciencedirect.com/science/article/pii/S0168169919319787

Duong L T, Doan T T, Chu C Q, Nguyen P T (2023) Fusion of edge detection and graph neural networks to classifying electrocardiogram signals. Expert Syst Appl 225:120107, . ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2023.120107. URL https://www.sciencedirect.com/science/article/pii/S0957417423006097

Espejo-Garcia B, Mylonas N, Athanasakos L, Vali E, Fountas S (2021) Combining generative adversarial networks and agricultural transfer learning for weeds identification. Biosyst Eng 204:79–89, ISSN 1537-5110. https://doi.org/10.1016/j.biosystemseng.2021.01.014. URL https://www.sciencedirect.com/science/article/pii/S1537511021000155

Ferentinos K P (2018) Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2018.01.009. URL http://www.sciencedirect.com/science/article/pii/S0168169917311742

Hatamizadeh A, Yin H, Kautz J, Molchanov P (2022) Global context vision transformers, URL arXiv:2206.09959

Hu K, Coleman GRY, Zeng S, Wang Z, Walsh M (2020) Graph weeds net: a graph-based deep learning method for weed recognition. Comput Electron Agric 174:105520

Hu M, Fen H, Yang Y, Xia K, Ren L (2018) Tree species identification based on the fusion of multiple deep learning models transfer learning. 2018 Chinese Automation Congress (CAC), pages 2135–2140,

Huang Z, Pan Z, Lei B (2017) Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. Remote Sens 9(9):2072–4292. https://doi.org/10.3390/rs9090907. (**ISSN**)

Hung C, Underwood J, Nieto J, Sukkarieh S (2015) A feature learning based approach for automated fruit yield estimation, pages 485–498. Springer International Publishing, Cham, ISBN 978-3-319-07488-7. https://doi.org/10.1007/978-3-319-07488-7_33

Kamilaris A, Prenafeta-Boldú F X (2018) Deep learning in agriculture: a survey. Comput Electron Agric 147:70–90, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2018.02.016. URL http://www.sciencedirect.com/science/article/pii/S0168169917308803

Lehnert C, English A, McCool C, Tow AW, Perez T (2017) Autonomous sweet pepper harvesting for protected cropping systems. IEEE Robot Autom Lett 2(2):872–879. https://doi.org/10.1109/LRA.2017.2655622. (**ISSN 2377-3766.**)

Liu Y, Barrett CB, Pham T, Violette W (2020) The intertemporal evolution of agriculture and labor over a rapid structural transformation: Lessons from vietnam. Food Policy 94:101913, ISSN 0306-9192. https://doi.org/10.1016/j.foodpol.2020.101913. URL http://www.sciencedirect.com/science/article/pii/S0306919220301172. Understanding Agricultural Development and Change: Learning from Vietnam

Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Olsen A (2020) Deepweeds dataset and tool, July . URL https://github.com/AlexOlsen/DeepWeeds/

Olsen A, Konovalov D, Philippa B, Ridd P, Wood J, Johns J, Banks W, Girgenti B, Kenny O, Whinney J, Calvert B, Rahimi Azghadi M, White R (2019) DeepWeeds: a multiclass weed species image dataset for deep learning. Sci Rep 9:12. https://doi.org/10.1038/s41598-018-38343-3

Olsen A, Konovalov D, Philippa B, Ridd P, Wood JC, Johns J, Banks W, Girgenti B, Kenny O, Whinney J, Calvert B, Azghadi M, White Deepweeds R (2019) A multiclass weed species image dataset for deep learning. Sci Rep 9:2

Pilarski T, Happold M, Pangels H, Ollis M, Fitzpatrick K, Stentz A (2002) The demeter system for automated harvesting. Auton Robot 13(1):9–20. https://doi.org/10.1023/A:1015622020131. (**ISSN 1573-7527.**)

Rehman TU, Mahmud MS, Chang YK, Jin J, Shin J (2019) Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. Comput Electron Agric, 156:585–605, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2018.12.006. http://www.sciencedirect.com/science/article/pii/S0168169918304289

Reis MM, da Silva AJ, Junior JZ, Santos LDT, Azevedo AM, Gonçalves Lopes Érika M (2019) Empirical and learning machine approaches to estimating reference evapotranspiration based on temperature data. Comput Electron Agric 165:104937, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2019.104937. URL http://www.sciencedirect.com/science/article/pii/S0168169919306787

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems 28, pages 91–99. Curran Associates, Inc.,

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y. (**ISSN 0920-5691.**)

Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C (2016) Deep-Fruits: a fruit detection system using deep neural networks. Sensors 16(8):1222 (http://dblp.uni-trier.de/db/journals/sensors/sensors16.html#SaGDUPM16d)

Sabzi S, Abbaspour-Gilandeh Y, Javadikia H (2017) The use of soft computing to classification of some weeds based on video processing. Appl Soft Comput 56:107–123,. ISSN 1568-4946. https://doi.org/10.1016/j.asoc.2017.03.006. URL https://www.sciencedirect.com/science/article/pii/S1568494617301291

Sharpe SM, Schumann A, Boyd N (2020) Goosegrass detection in strawberry and tomato using a convolutional neural network. Sci Rep 10:15

Tam H T, Shimada K (2019) The effects of climate smart agriculture and climate change adaptation on the technical efficiency of rice farming-an empirical study in the mekong delta of vietnam. Agriculture 9(5):99. https://doi.org/10.3390/agriculture9050099. (**5**)

Tan M, Le Q (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/tan19a.html

Thenmozhi K, Reddy US (2019) Crop pest classification based on deep convolutional neural network and transfer learning. Comput Electron Agric 164:104906,ISSN 0168-1699. https://doi.org/10.1016/j.compag.2019.104906. URL http://www.sciencedirect.com/science/article/pii/S0168169919310695

Vaswani S, Mishkin A, Laradji I, Schmidt M, Gidel G, Lacoste-Julien S (2019) Painless stochastic gradient: interpolation, line-search, and convergence rates. Adv Neural Inf Process Syst 32:25

Weiss K, Khoshgoftaar T, Wang D (2016) A survey of transfer learning. J Big Data 3:12. https://doi.org/10.1186/s40537-016-0043-6

Wightman R (2019a) Pytorch image models. https://github.com/rwightman/pytorch-image-models,

Wightman R (2019b) Pytorch image models. https://github.com/rwightman/pytorch-image-models,

Xie C, Tan M, Gong B, Wang J, Yuille A, Le QV (2019) Adversarial examples improve image recognition. arXiv e-prints, art. arXiv:1911.09665,

Xie Q, Luong M-T, Hovy E, Le QV (2019) Self-training with noisy student improves imagenet classification

Zhao Y, Gong L, Huang Y, Liu C (2016) A review of key techniques of vision-based control for harvesting robot. Comput Electron Agric 127:311–323, 2016. ISSN 0168-1699. https://doi.org/10.1016/j.compag.2016.06.022. URL http://www.sciencedirect.com/science/article/pii/S0168169916304227

Zhu Y, Sun W, Cao X, Wang C, Wu D, . Yang Y, Ye N (2019) Ta-cnn: Two-way attention models in deep convolutional neural network for plant recognition. Neurocomputing 365:191–200, . ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2019.07.016. URL http://www.sciencedirect.com/science/article/pii/S0925231219309440

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.