IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2022, accepted 17 August 2022, date of publication 22 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3200603

RESEARCH ARTICLE

Hybrid CNN Model for Classification of Rumex Obtusifolius in Grassland

AHMED HUSHAM AL-BADRI[1], **NOR AZMAN ISMAIL**[1], **KHAMAEL AL-DULAIMI**[2,3], **AMJAD REHMAN**[4], **IBRAHIM ABUNADI**[4], **AND SAEED ALI BAHAJ**[5,6]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[2]Computer Science Department, College of Science, Al-Nahrain University, Baghdad 10072, Iraq
[3]School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, QLD 4059, Australia
[4]Artificial Intelligence and Data Analytics Laboratory, CCIS, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5]MIS Department, College of Business Administration, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[6]Department of Computer Engineering, College of Engineering and Petroleum, Hadhramaut University, Mukalla 50511, Yemen

Corresponding authors: Amjad Rehman (drrehman70@gmail.com) and Saeed Ali Bahaj (saeedalibahaj@gmail.com)

ABSTRACT *Rumex obtusifolius* Linnaeus (*R. obtu.* L.) is one of the vital broad-leaved weeds in grassland that needs removal. It affects dairy products and reduces their quality. Hand-removal methods are costly and time-consuming. Chemical treatment using herbicides has a negative impact on crops and causes environmental pollution. In smart farming, weeding is performed by using computer vision to recognize the weeds efficiently and effectively. Conventional machine learning (ML)-based algorithms face challenges, especially in identifying the weeds in real-world data due to a lack of features. Deep learning (DL) approaches use self-learning to extract all potential features that assist in classifying malignant weed species accurately. Recently, single deep learning methods achieved high performance in identifying well-separated and illumination but suffered from misclassification in more sophisticated cases such as overlapping and partial occlusion leaves. This paper presents a hybrid Convolutional Neural Network (CNN) model of three state-of-the-art CNNs to classify *Rumex obtusifolius*. The proposed model utilizes convolutional neural networks to extract features and classify images. The framework of the proposed method comprises three paramount stages to accomplish the classification key idea, including the data preparation phase, pre-processing phase, and classification phase. A hybrid model of three CNN extractor networks is used as the backbone in the classification stage. Our tested data is real-world data that includes multi-circumstances (overlap, occlusion, various illuminations, etc.) acquired from nature. The first extractor is the Visual Graphics Group-16 (VGG-16) for well-separated leaves and non-complicated issues. The second extractor is Residential Energy Services Network-50 (ResNet-50), to overcome complex real-world issues. The third extractor is Inception-v3 to solve the illumination problem. Therefore, combining three networks into one model improves the discriminatory ability to extract additional useful features. The proposed model has been tested using two benchmark datasets for *Rumex* weed plants. Both of these datasets were captured in real-world environments. The first dataset consists of 900 samples, while the second dataset consists of 677 samples. Each dataset is individually tested in the proposed model to evaluate the classification accuracy using a set of standard evaluation metrics including accuracy, precession, recall, True-Positive Rate (TPR), False-Positive Rate (FPR), and F1-score. The total averages of the proposed model on both datasets are 97.51%, 97.4%, 94.45%, and 95.9% on the accuracy, recall, precision, and F1-score, respectively.

INDEX TERMS CNN networks, ensemble models, real-world data, weed classification, economic growth.

NOMENCLATURE

CNN	Convolutional Neural Network.
Mask R-CNN	Mask region-convolutional neural network.
E-RCNN	Ensemble-region convolutional neural network.
ML	Machine learning.

The associate editor coordinating the review of this manuscript and approving it for publication was Liandong Zhu.

DL	Deep learning.
R. obtu. L.	*Rumex obtusifolius* linnaeus.
VGG	Visual graphics group.
ResNet	Residential energy services network.
SIFT	Scale invariant feature transform.
SURF	Speed-up robust feature.
KNN	K-nearest neighbor.
SVM	Support vector machine.
L2regL2lossSVCp	Linear2-regularized with Linear2-loss logistic regression model using primal computation.
L2regLogReg	L2-regularized with L2-loss logistic regression.
RF	Random forest.
LBP	Local binary patterns.
YOLO	You only look once.
NDVI	Normalized difference vegetation index.
FPR	False-positive rate.
FNR	False-negative rate.
TPR	True-positive rate.
TNR	True-negative rate.
UAV	Unmanned aerial vehicle.
FCL	Fully connected layers.
ReLU	Rectified linear unit.
RBF	Radial basis function.
RoI	Region of interest.
DoI	Domain of interest.

I. INTRODUCTION

In recent years, continuous development towards weed control within planted crops has been offered. *Rumex obtusifolius (R. obtu.),* or dock broad-leaved, is considered an undesirable weed plant in agriculture that necessitates removal. The harmful effects of this weed have spread around the world, particularly in Europe. In Germany, 85% of organic farms encounter broad-leaved dock issues. It diminishes the grass's productivity by 10%–40% [1]. Due to the *Rumex's* widespread nature, livestock gormandize it readily and intensively. Therefore, it has a substantial impact on dairy and productivity [2]. It causes animal health issues due to high oxalic acid, which hinders the quality of products due to low nutritional value. In addition, it significantly affects the economic growth of countries. Progress in the accurate classification of *Rumex* is, however, restricted by the demand for physical removal or chemical treatments. Therefore, these issues triggered this study to design a robust weed classification model that can be utilized in an automatic weed control system to classify this harmful species of weed. Due to the vast range of weed species in nature and working conditions, this research issue is fraught with challenges.

Manual or hand-weeding is one of the well-known techniques to eliminate weeds. The farmer scans the entire farm for undesirable or unusual plants, plugging them out using his hands or simple tools. Their technique faces numerous challenges, such as lengthy-time of completion, difficulty of detection, and labor-cost. Another method for malignant weed removal is chemical treatment, which targets numerous weeds using herbicides sprayed on large-scale farms [3]. Farmers or machines perform this process. The problem with such a technique is the treatment cost and environmental pollution issues. In addition, this adversely affects animal and human health [4]. Thus, both hand-removing and chemical treatment techniques are time-consuming, costly, and can result in environmental issues [5].

Nowadays, precision farming or smart farming approaches utilize computer vision as an alternate technique to determine the Region of Interest (RoI) in the pasture [6]. These approaches are more robust in terms of efficiency and effectiveness. Some studies focused on detecting various weed species over the last three decades by discriminating these weeds from crop plants, as in Binch and Fox [7]. In their work, Jia *et al.* [8] examined the identification of the plant on the farm using thresholding. They located the root position by computing the cross points of major veins in corn leaf images captured from the top scene. In the same year, Franz *et al.* [9] used the curvature technique to detect partially occluded leaves on different seedlings plants at late growth stage. By aligning the resampled curvatures for each genus, the author revealed the significance of identifying a leaf that was not entirely occluded. The shortage in their approach was related to the accuracy of curvature to identify various shape of serration. Tian [10] utilized spatial features to determine certain locations of cotyledon crop plants. They obtained the location information by calculating the center point of the stem during the early growth stage. Woebbecke *et al.* [11] proposed a method for differentiating monocot and dicot weed plants, representing two weed species that exist in the United States. They claimed that the optimum period to address these harmful plants would be from the 14th to the 23rd day of their growth. In the last three decades, real-world data has remained a challenging task for the computer vision scientific community. Occlusion, overlapping, different illumination, and various growth stages conditions are common issues in real-world data [12]. Hand-engineering features are basic Machine Learning (ML) methods to extract features manually. These methods achieved satisfying results with artificial data under controlled conditions. Deep Learning (DL)-based approaches are an extension of ML approaches to achieve encouraging results with real-world data using self-extracting features [13].

The key limitation of the previous methods is how to accurately classify the *Rumex* under real-world conditions in the case of insufficient training images. Due to the mundane nature of annotating a huge number of images, the motivation of designing a model to work with a reasonable number of images that contain various real-world conditions is desirable. Therefore, the main drive of this work is to investigate the issues of *Rumex* classification, including heavy occurrence, various growth stages, overlapping with plants, and adverse environmental-agricultural impact. In addition, using mechanical or chemical actuation methods to control

Rumex species impacts human and animal life and reduces the plants' amount and quality. This is the first time that a hybrid Convolutional Neural Network (CNN) has been used to improve the accuracy of classifying *Rumex* weed plants in a complicated scenario. The purpose of using CNN networks in classification is to provide a robust method for self-extracting features [14].

Our main contribution focuses on designing a new ensemble model of three CNN networks at its backbone base. The framework of this model is adaptable to numerous weed control applications to address various weed species. This ensemble uses voting majority rule to decide whether the plant is considered by the model as a weed or not. This combination has not previously been used in the *Rumex* classification model to the best of our knowledge. The Ensemble-Region Convolutional Neural Network (E-RCNN) network is proposed for its novelty in using ensemble classifiers at its backbone base. The second contribution is using the new proposed model to address data challenges under real-world conditions such as occlusion, overlapping, various image resolutions, various growth stages, and different illumination conditions. Combining three extractors into one model provides the following expected benefits: i) enhancing classification accuracy, ii) reducing the illumination effect, iii) controlling the occluded and overlapped issues, and iv) enhancing the capability of feature extraction and representation. This study uses benchmark datasets from Kounalakis *et al.* [15] and Van Evert *et al.* [1]. Both sets are real-world of the actual farm that were captured under challenging conditions, such as various illuminations, occlusions, and overlapping conditions.

The remainder of this paper is organized as follows: The related work is presented in Section II; the proposed model is outlined in Section III; the materials and methods are explained in Section IV; Section V is the conclusion; and finally, limitations and future trends are elucidated in Section VI.

II. RELATED WORK

This section discusses most related works that explored ML and DL techniques to address broad-leaved weed plants. Dürr *et al.* [16] utilized the Local Binary Patterns (LBP) with C-histograms to extract the size and spectral features of the *Rumex* weeds. Then, they eliminated the detected regions using a heating oven at 1200 KW. The problem with their method is the high error rate of misclassified regions, which reached 35%. Van Evert *et al.* [17] found that texture is a significant feature for identifying broad-leaved weeds like *Urtica* and *Rumex*. Binch and Fox [7] compared different ML algorithms using real data. Their comparison demonstrated that the best results were obtained by combining LBP with Support Vector Machine (SVM) for *Rumex* classification. Unfortunately, the LBP method relies heavily on texture features and ignores beneficial information such as shape, and color. This dependence restricts the method's performance, making it unable to mitigate the error rates.

Gao *et al.* [18] used the Normalized Difference Vegetation Index (NDVI) color index with Random Forest (RF) to classify *Rumex* and two additional species of weeds, *Convolvulus arvensis* and *Cirsium arvense,* from the maize crop. The mean classification rate of *Rumex* is 69.1%, which is better than the K-Nearest Neighbor (KNN). However, they depend on a specific number of extracted features from 8 different bands. In addition, their method is costly due to the multispectral camera. Kounalakis *et al.* [19] proposed Speed-Up Robust Feature (SURF) features with Linear2-regularized with Linear2-loss logistic regression model using primal computation (L2regL2lossSVCp) to recognize the *Rumex*. They captured 100 images of *Rumex* in a real field using multiple high-resolution cameras. Then, each image is segmented into 9 patches to yield 900 images. Dividing an image into muti-regions has adversely affected the quality of the features represented in the image. Moreover, their method is based on hand-crafted features that represent image content. Thus, the classification results of such a method record 89.09% accuracy with a 4.38% False-Positive Rate (FPR).

Zhang *et al.* [20] utilized a single CNN approach to recognize *Rumex obtusifolius* in various illumination conditions. They achieved 96.88%. The problem with such a method is that resizing the input image size to 64 × 64 pixels causes a loss of useful information that can assist in solving challenging cases. Valente *et al.* [21] used AlexNet transfer learning to classify *Rumex obtusifolius* in grassland. They generated high-resolution data using a small Unmanned Aerial Vehicle (UAV). They scored 91.9% accuracy when the *Rumex* in moved and cut-off cases. The drawback of such a method is that the images are not tested under various illumination cases. In addition, they captured their images from the same level at 10 meters in height. Such image types restrict the method from obtaining sufficient information about the leaves and the entire object. This limits the method's performance to identifying *Rumex* in various real-world conditions. Lam *et al.* [22] used the Visual Graphics Group (VGG) method to classify the early growth of *Rumex* weeds using UAV. One of the limitations of their method is focusing on limited cases of *Rumex* that are found on one field site and ignoring other cases such as the different growth stages and other real conditions. The results of their proposed method are 92.1% and 78.7% on the accuracy and F1-score, respectively. In this work, we utilized the data collected by Kounalakis *et al.* [23] and Van Evert *et al.* [1] to estimate the performance of the proposed model. Furthermore, four extracted features comprise visual texture features, spatial context features, spectral features, and biology morphology features. Besides, their study supported the idea that the sophisticated system is a trade-off between accuracy and efficiency. Finally, Kounalakis *et al.* [23] applied the transfer learning technique to recognize *Rumex* in grassland.

The significant contributions of this research are designing a new ensemble model of three CNN architectures to enhance the classification accuracy of *Rumex*. To the best of our knowledge, the three DL networks were not previously

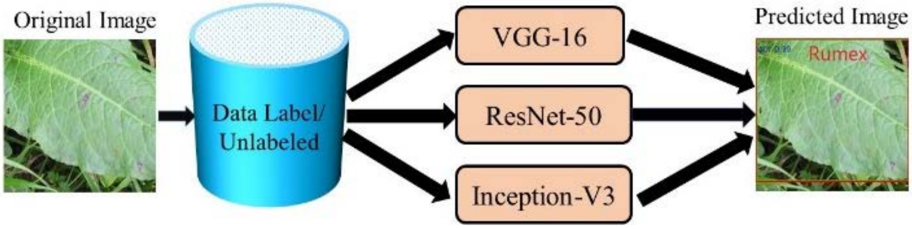


FIGURE 1. Ensemble diagram.

combined into one model to be implemented in the agricultural field, especially with *Rumex*. In addition, this work improves the classification accuracy for *Rumex* and reduces the error rate. This improvement leads the scientist to utilize this model to be generalized to various weed plants.

III. PROPOSED METHOD

In this paper, the proposed method has been discussed thoroughly. Three main stages are identified and proposed to accomplish the classification objectives, including the data preparation stage, pre-processing stage, and classification stage. The stages of the methodology are thoroughly discussed in the following subsections. This paper highlights the generic framework of this research and introduces the required steps to implement the research systematically. Ensemble learning is the aggregation of numerous models, such as extractors and classifiers, to tackle a specific computational intelligence issue. The key idea of ensemble learning is to enhance classification accuracy and prediction. The structure of the E-RCNN network consists of two parts. The first one is the features extractor, and the second part is the classifier network. Each extractor consists of convolutional layers, dropouts, and max-pooling layers in between. Three CNN models are adapted in their structure to fit the data requirements. This data suffers from illumination, overlapping, and occlusion. These three extractors are merged to form a hybrid model. The ensemble model requires an odd number of methods for voting purposes, such as three, five or seven and upwards. Therefore, determining the number of elements (e.g., methods) in an ensemble is critical [24]. Three selected methods are combined to design our proposed model in this case. Using more than three architectures in one model increases the memory space and reduces efficiency. Regarding using more than five networks, the model complexity is also increased; as a result, the model will be complicated, which negatively affects the system's performance. Fig. 1 illustrates the mechanism of the ensemble. First, each variable is passed through the three extractors to be processed. Then, these individual extractors' outputs contain the predicted label. Hence, the predicted outputs attained from the three extractor backbones are passed through the ensemble model as inputs to vote for one classified label in each process. Each classified object is selected to have a low error rate with a high probability. The formula of the ensemble is depicted in the below Equation.

Given some training data:

$$\mathcal{D}_{\text{train}} = \mathbf{x}_n, y_n; \quad n1, \ldots, N_{\text{train}} \tag{1}$$

where:

D: represents the classifier model.
n: is the number of classes.
Inductive learning:

$$\mathcal{L} : \mathcal{D}_{\text{train}} \rightarrow h(\cdot), \quad \text{where } h(\cdot) : \chi \rightarrow \mathcal{Y} \tag{2}$$

Ensemble learning:

$$\mathcal{L}_T : \mathcal{D}_{\text{train}} \rightarrow h_T(\cdot) \Rightarrow \{h_I(\cdot), h_2(\cdot), \ldots, h_T(\cdot)\} \tag{3}$$

The ensemble model yields optimum performance when there is critical diversity in the output results of the composing methods [25]. The first layers of the feature extractor network extract useful features such as color identification, edges, and curves of the objects in the image. Then, the annotated images were divided to the ratio of 80:20 into a training set and a testing set, respectively. All this data is with RGB color images of various sizes. After that, the prepared data becomes ready to feed the proposed model. Generally, DL models require a small square image to reduce the time and memory-constrains. In addition, DL networks require a fixed resolution of training images to feed the network [26]. Furthermore, data augmentation is utilized to improve generalization [27], [28], [29]. In this regard, the image resizing technique is applied to decline the input image size to the standard size of 224 × 224 × 3 pixels [30], [31], [32]. Furthermore, DL methods necessitate a vast dataset to increase accuracy and hinder overfitting.

The outputs of these methods are grouped to produce the final predictions. Each extractor uses the mean subtraction algorithm located in the data loader. This technique assists in accumulating the data around the mean where the helpful features exist. The benefit of such a technique is that it reduces the effects of outliers and illumination issues in some cases. Due to the high performance of the three networks in the ImageNet competition [14], they are selected to combine the proposed model.

1. The first extractor model is VGG-16 [33], which is the basis of our hybrid model. It is efficient and accurate [34] to handle well-separated and some partially occluded leaves.

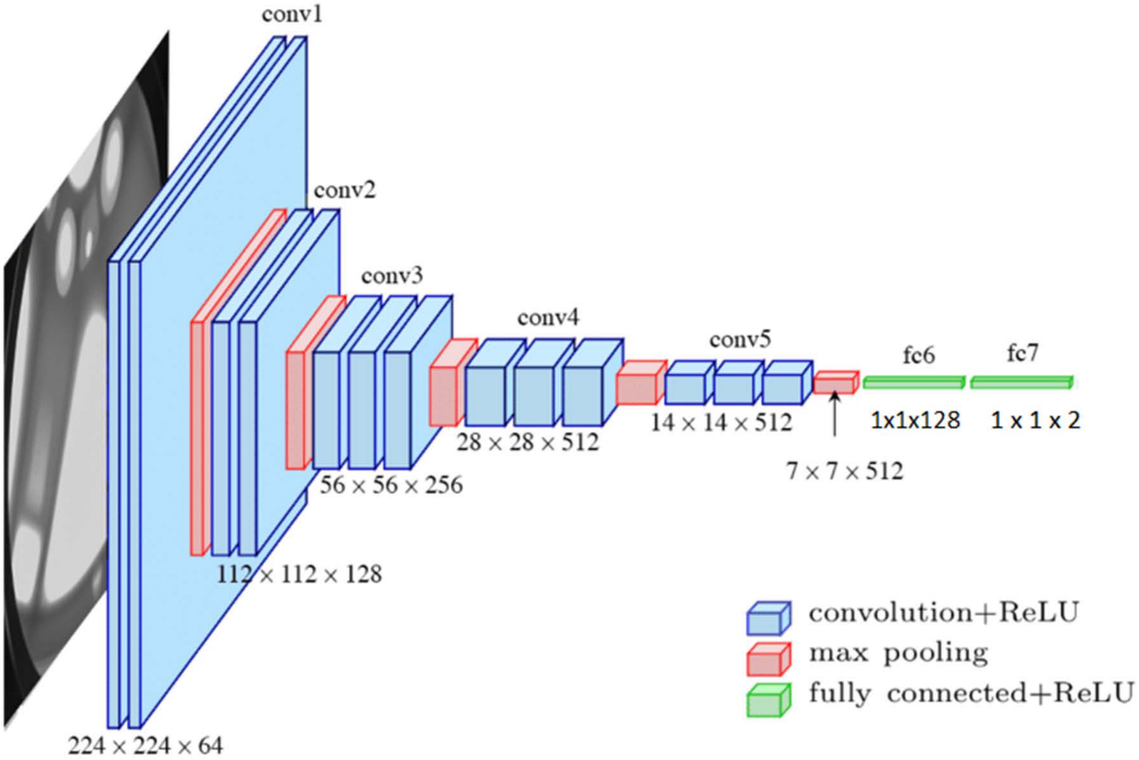


FIGURE 2. The architecture of VGG-16.

2. Residential Energy Services Network-50 (ResNet-50) [35] is the second extractor model, which has a more dense convoluted architecture than VGG-16 [33] due to its further dense layers. Nevertheless, it handles the overfitting problem of VGG-16 and deals with more sophisticated issues such as overlapping and occlusion.
3. The final extractor is the Inception-V3 model [36], which is more robust than the VGG-16 and ResNet-50 to overcome the illumination issue not solved in the preprocessing step.

The E-RCNN utilizes a hybrid model composed of three backbone networks, including VGG-16, ResNet-50, and Inception-V3 for feature extraction and classification. Further explanation can be found in [33] and [35].

A. VGG-16 EXTRACTOR

Fig. 2 illustrates the sixteen layers of the VGG-16 network architecture. Some of these layers include trainable parameters, while some do not, like the Max pool layer. The key idea of the VGG depth group was to investigate how the depth of convolutional networks influences the accuracy of models for wide-range image recognition and classification. All of VGG's architectures have many Fully Connected Layers (FCLs) with various convolutional layers. The more depth, the more convolutional layers. Fig. 2 shows thirteen blue rectangles related to the hidden layers and the non-linear activation function represented by the Rectified Linear Unit (ReLU). The five red rectangles are related to the max-pooling layers. In addition, two green rectangles represent two FCLs. Therefore, the total number of layers with adjustable parameters is 15, including 13 convolution layers and 2 FCL layers. The proposed method fine-tunes the last two layers, the SoftMax layers, to fit our dataset. In this work, the SoftMax function is re-initialized to carry the appropriate number of classes for the samples to decide whether the plant is *Rumex* or non-*Rumex*. In this design, VGG-16 commenced with a relatively small channel capacity of 64 and rose by a scale factor after each max-pooling layer till it reached 512. Fig. 3 shows the flattened architecture of VGG-16.

The structure consists of five blocks. The first two adjacent blocks are composed of pair-convolution layers and then max-pooling. The last three contiguous blocks have three convolution layers followed by max-pooling. Finally, the last three dense layers represent the FCL, or as they are known, the classification layers [22]. The first two FCLs are flattened, consisting of 512 depths, while the last FCL includes 128 depths. The size is reduced by half after every max-pooling. Table 1 displays the VGG-16's overall network configurations.

These are the characteristics of the VGG-16 network:

1. Input Layer: It accepts 224 × 224 color images with three channels as input.

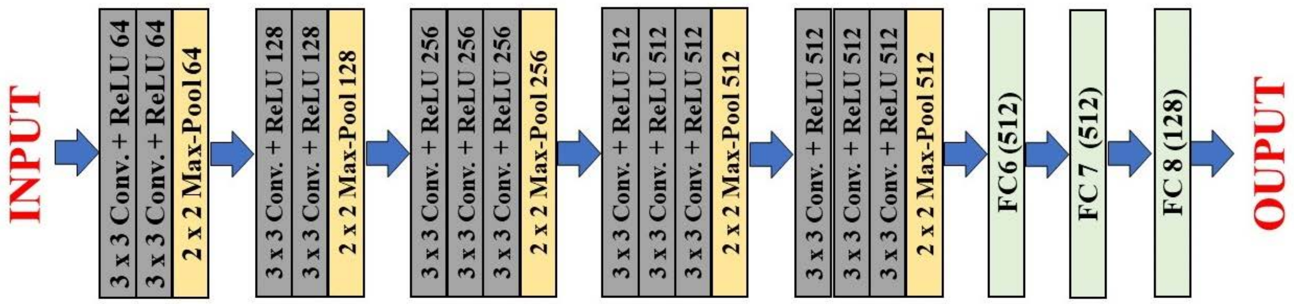


FIGURE 3. VGG-16's flattened architectural design.

TABLE 1. The configuration summary of the VGG-16.

Layer (type)	Output Shape	No. of Parameters	Pool-Size	Kernel-Size	Float
inpt_1 (InpLyr)	[(Nil, 224, 224, 3)]	0	-	-	-
blk1_conv1 (Conv_2D)	(Nil, 224, 224, 64)	1792	-	(3, 3)	32
blk1_conv2 (Conv_2D)	(Nil, 224, 224, 64)	36928	-	(3, 3)	32
blk1_pool (MaxPool_2D)	(Nil, 112, 112, 64)	0	(2, 2)	-	32
blk2_conv1 (Conv_2D)	(Nil, 112, 112, 128)	73856	-	(3, 3)	32
blk2_conv2 (Conv_2D)	(Nil, 112, 112, 128)	147584	-	(3, 3)	32
blk2_pool (MaxPool_2D)	(Nil, 56, 56, 128)	0	(2, 2)	-	32
blk3_conv1 (Conv_2D)	(Nil, 56, 56, 256)	295168	-	(3, 3)	32
blk3_conv2 (Conv_2D)	(Nil, 56, 56, 256)	590080	-	(3, 3)	32
blk3_conv3 (Conv_2D)	(Nil, 56, 56, 256)	590080	-	(3, 3)	32
blk3_pool (MaxPool_2D)	(Nil, 28, 28, 256)	0	(2, 2)	-	32
blk4_conv1 (Conv_2D)	(Nil, 28, 28, 512)	1180160	-	(3, 3)	32
blk4_conv2 (Conv_2D)	(Nil, 28, 28, 512)	2359808	-	(3, 3)	32
blk4_conv3 (Conv_2D)	(Nil, 28, 28, 512)	2359808	-	(3, 3)	32
blk4_pool (MaxPool_2D)	(Nil, 14, 14, 512)	0	(2, 2)	-	32
blk5_conv1 (Conv_2D)	(Nil, 14, 14, 512)	2359808	-	(3, 3)	32
blk5_conv2 (Conv_2D)	(Nil, 14, 14, 512)	2359808	-	(3, 3)	32
blk5_conv3 (Conv_2D)	(Nil, 14, 14, 512)	2359808	-	(3, 3)	32
blk5_pool (MaxPool2D)	(Nil, 7, 7, 512)	0	(2, 2)	-	32
avg_pool2d (AvgPool2D)	(Nil, 1, 1, 512)	0	(7, 7)	-	32
flatn (Flatn)	(Nil, 512)	0	-	-	32
dens (Dens)	(Nil, 128)	65664	-	-	32
drpout (Drpout)	(Nil, 128)	0	-	-	32
dens_1 (Dens)	(Nil, 2)	258	-	-	32
Total parameters		**14,780,610**			

2. Convolution Layers: They are a sequence of dense layers that the input images are passed through. Every convolution filter has a tiny filter of 3 × 3 size with a stride of 1. Each window size (also known as kernel size) utilizes

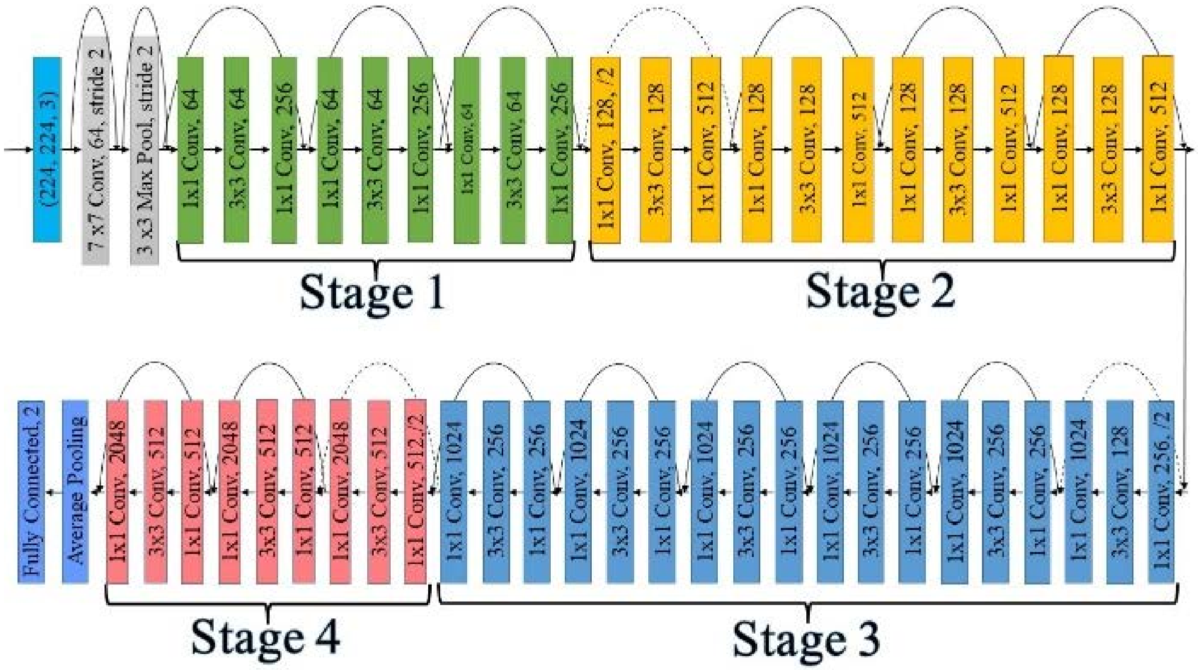


FIGURE 4. The architecture of ResNet-50.

row and column padding to preserve the input and output size as fixed.

3. Max pooling: Is implemented across a 2 × 2 of window size with a stride of 2, indicating that max pool windows are non-overlapping windows.
4. A max pool layer is not always the layer that follows a convolution layer. Instead of the max-pool layer, a convolution layer is followed by another convolution layer.
5. The proposed model modifies the original technique by replacing the last three connected layers of the original method with two FCLs to fit the number of our classes. The first FCL has 1 ×1 × 28 neurons. Increasing the number of neurons means increasing the complexity and processing time of the model with the same accuracy, causing overfitting, while decreasing this number causes underfitting. The second FCL consists of two outputs 1 × 1 × 2 as there are two classes, *Rumex* and non-*Rumex* in our dataset.
6. ReLU is the activation function that is used in the hidden layers.

To justify selecting the window size of 3 × 3 is that it is the minimum potential value to fulfill the required directions of the entire image from top to bottom and from left to right passing through the center. Furthermore, stacking pair-convolutional layers of 3 × 3 excepting max-pooling between them has an effective receptive field of 5 × 5. Similarly, using triple 3 × 3 convolution layers have an effective receptive field of 7 × 7.

B. RESNET-50 EXTRACTOR

There are four stages of the ResNet-50 architecture, as illustrated in Fig. 4. The dimensions of the input image for this network are 224 × 224 × 3. The configuration of kernel sizes in ResNet uses 7 × 7 and 3 × 3 for initial convolution and max-pooling, respectively. After which, the process of the first stage commenced. The first stage consists of three residual blocks. Each block of the residual includes three layers. The kernel sizes of the layers in the block residual are 64, 64, and 256, respectively. There are two types of curved arrows. The first type is connected curved arrows used with an identity connection. The second type of curved arrow is the dashed curved arrow, denoting that the convolution operation is using stride 2 in the residual block. At this stage, the input size of the image was reduced by 50% for the height and width, while the channel increased by dual. Observably, the channel width increases to dual, whereas the input size decreases as it proceeds through the stages. Most deeper networks such as ResNet-101 and ResNet-50 provide a bottleneck in their architectures.

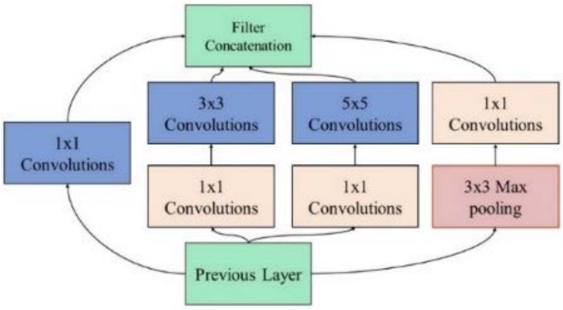


FIGURE 5. GoogLeNet network with the inception layer.

The benefit of using bottlenecks in such an architecture is that it decreases the number of network parameters

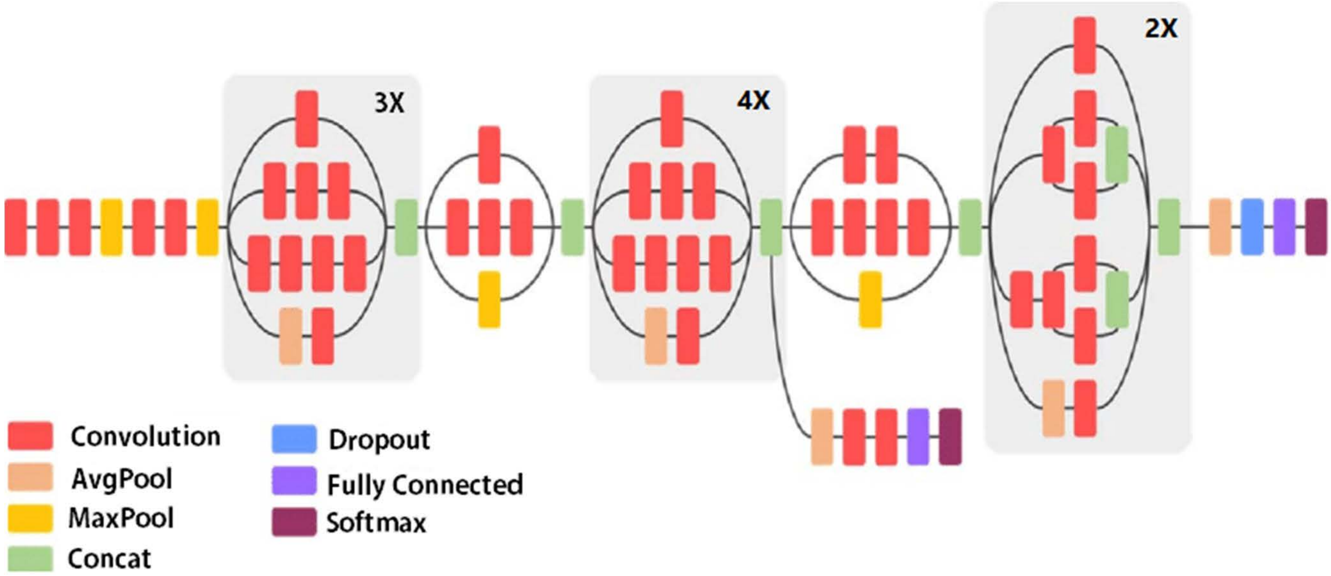


FIGURE 6. The architecture design of Inception-V3.

represented by feature maps in the network while preserving the network's depth. Another advantage is that it permits the network to be generalized with new data. The bottleneck consists of a small-dimensions convolution layer that is 1 × 1, where the number of output channels of this layer is less than the number of input channels. Each residual function comprises three layers assembled on top of one another. The dimensions of these convolutional layers are 1 × 1, 3 × 3, and 1 × 1. The first and third convolutional layers of 11 are employed to reduce and then retrieve the input resolution. At the same time, the second convolution layer, which is 3 × 3 is used as the bottleneck to resize the dimensions for input and output [37]. In addition, our research fine-tunes the FCL to fit with the class numbers of our data, which includes two classes, *Rumex* and non-*Rumex*.

C. INCEPTION-V3 EXTRACTOR

Unlike ResNet-50, the Inception family is a wider-style network rather than a deeper one. In Inception-V3, various multiple transformations of the same input map are calculated simultaneously. Then, the results are concatenated into a solo output. The previous version of Inception used three layers of 5 × 5, 3 × 3 convolution, and one max-pool. In the later versions, the filter size of the 5 × 5 convolution layer was replaced with two 3 × 3 convolution layers, instead of using only one large filter size. This reduction is called factorization. The benefit of factorizing is to reduce the number of parameters by 28%, which helps to reduce the computational cost. Generally, the purpose of increasing the depth of any network is to enhance accuracy. However, it causes vanishing gradient issues, such as consuming additional resources for computation. To overcome this issue, Inception-V3 introduced an auxiliary unit of a 1 × 1 convolution layer. Using these units is helpful because they address the problem of vanishing gradients and make a more comprehensive network [38]. Fig. 5 shows the effects of adding 1 × 1 convolution on the computational cost of Inception-V3. Szegedy *et al.* [36] claimed that using a bottleneck in the initial layers causes the loss of useful information from the input layer. In addition, they adopted one of the principles in all the Inception families to enhance the accuracy of classification at a reasonable computational cost by parallel increasing the width and depth. Inception-V3 differs from the other Inception families in using additional techniques such as factorized 7 × 7 convolutions, label smoothing, and auxiliary units or auxiliary classifiers [36]. Fig. 6 illustrates the Inception-V3 architecture. Our research modified the last two layers to fit our data. The experimental configurations and the parameter details of our Inception-V3 are depicted in Table 2.

IV. MATERIALS AND METHODS

A. DATASET DESCRIPTION

The description of the data used in this study is elucidated thoroughly. In this study, two standard benchmark datasets have been used. The first dataset [dataset 1] is obtained from Kounalakis *et al.* [23]. The total number of images in this dataset is 900 images of *Rumex* weed plants in grassland. The second dataset [dataset 2] is acquired from Van Evert *et al.* [17]. The total number of images in this dataset is 677 images of *Rumex* weed plants in grassland. Both these datasets are two-dimensional RGB-colored images. The format of these datasets is Joint Photographic Group (JPG). The images in the dataset have various resolution sizes. The first data was captured using a robotic system on an organic dairy farm

TABLE 2. Summary of Inception-V3 configuration.

Layer (type)	Output Shape	Stride	Kernel-Size	Float	Layer (type)
inpt_2 (InptLyr)	[(Nil, 224, 224, 3)]	-	-	32	inpt_2 (InptLyr)
blk1_conv1 (Conv_2D)	(Nil, 111, 111, 32)	(2, 2)	(3, 3)	32	blk1_conv1 (Conv_2D)
btch_normal	(Nil, 111, 111, 32)	-	-	32	btch_normal
Activ_Fun	(Nil, 111, 111, 32)	-	-	32	Activ Fun.
blk2_conv1 (Conv_2D)	(Nil, 109, 109, 32)	(1, 1)	(3, 3)	32	blk2_conv1 (Conv_2D)
btch_normal(1)	(Nil, 109, 109, 32)	-	-	32	btch_normal(1)
activ(1)	(Nil, 109, 109, 32)	-	-	32	activ(1)
conv-2d(2): Conv_2D	(Nil, 109, 109, 64)	(1, 1)	(3, 3)	32	conv2d(2): Conv_2D
btch_normal(2)	(Nil, 109, 109, 64)	-	-	32	btch_normal(2)
activ(2)	(Nil, 109, 109, 64)	-	-	32	activ(2)
max_pool2d	(Nil, 54, 54, 64)	(2, 2)	(3, 3)	32	max_pool2d
conv2d(3): Conv_2D	(Nil, 54, 54, 80)	(1, 1)	(1, 1)	32	conv2d(3): Conv_2D
btch_normal_3	(Nil, 54, 54, 80)	-	-	32	btch_normal(3)
activ(3)	(Nil, 54, 54, 80)	-	(1, 1)	32	activ(3)
|	|	|	|	|	|
|	|	|	|	|	|
|	|	|	|	|	|
|	|	|	|	|	|
|	|	|	|	|	|
mixed10: Concatenate	(Nil, 5, 5, 2048)	-	-	32	mixd10: Concatenate
flatn: Flatn	(Nil, 51200)	-	-	32	flatn: Flatn
dens_2	(Nil, 128)	-	-	32	dens_2
drpout_1	(Nil, 128)	-	-	32	dropt_1
dens_3	(Nil, 2)	-	-	32	dens_3
Total parameters	**28,356,770**				

in France. In contrast, the second dataset was taken using a Cybershot DSC-60 by Sony, Tokyo, Japan, on two different dairy farms in the Netherlands. The size of dataset 2 is 2304 by 1728 pixels hand-held at a 1.7 m distance from the ground. These datasets are considered real-world images. Real-world data is captured under various circumstances such as illumination, overlapping, and occlusion. Due to a lack of information, these cases are classified as sophisticated issues for ML techniques. However, this information could contain significant features that are crucial in identifying the leaf type.

Through leaf type, it becomes feasible to identify the plant type. The images in this dataset either contain the entire *Rumex* weed surrounded by the grassland or only the leaves of this weed (e.g., *Rumex*). The RoI of this dataset is the ability to identify *Rumex* in the grass or among the leaves of a scene. Fig. 7 shows that both *Rumex* and grass are likely to share the green color, which increases the difficulty of ML to recognize them. However, the grass is a more intense color than *Rumex*. In the pasture, the grass frequently covers the weeds. Therefore, shape and size are considered apparent features. The *Rumex* leaf differs from the grass leaf in its shape. *Rumex* leaves are short and broad, whereas grass leaves are long with a narrow edge of several millimeters. According to texture, *Rumex* is coarser than grass, which carries valuable information in the classification. Van Evert *et al.* [1] claimed that the detection performance of *Rumex* improved when the grass was short and the plant was in rosette form.

B. IMAGE RESIZING

The first step in image preparation is image resizing. Several image sizes were introduced to train our proposed model, commencing from 128 × 128, which achieved acceptable performance. Then, we raise the scale to 196 × 196, which leverages the performance level by 2%. By proceeding with the rescaling process using 224 × 224, 299 × 299, 336 × 336 until 350 × 350, it is observed that the model yields optimum results in network performance when the input image is 224 × 224. Finally, we investigated that increasing the scale over 224 × 224 yields the same performance but with high computation.

C. DATA AUGMENTATION

After image resizing, data augmentation is implemented to boost the number of training samples [39] and mitigate overfitting [40]. Since CNN methods are greedy to vast annotated data [41], several transformations are implemented to enlarge

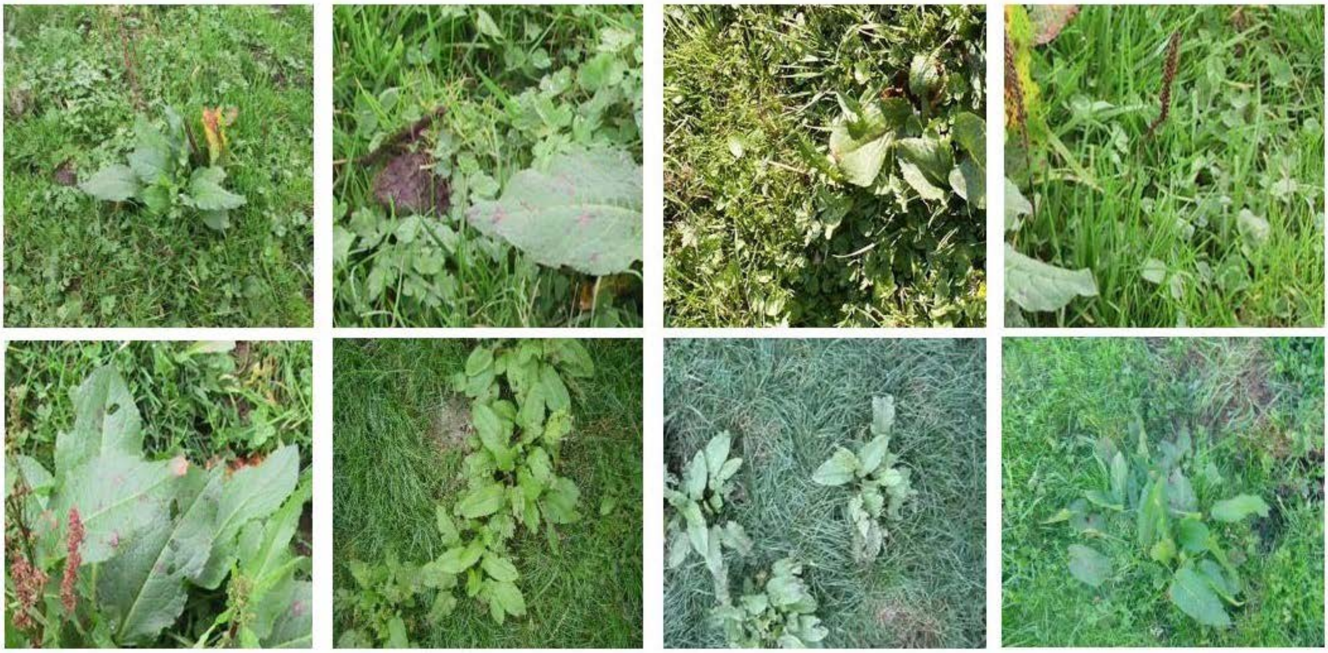


FIGURE 7. Original samples of *Rumex obtusifolius* (broad-leafed) weed plants in real-world conditions.

the training data size and introduce various shapes of *Rumex*, such as flip, mirror, and rotate. These transformations are randomly augmented for each epoch of training and validation. To implement all these transformations in Python, an exciting class, namely ImageDataGenerator, has been used. For rotation, each image is rotated 20 degrees clockwise to extend the dataset by 18 times to cover all potential changes in the input image's position. Then, both the horizontal and vertical scales with a range of 0.5 are utilized to enlarge the image. In addition, a cropping adjustment with a range of 15% is applied. Both vertical and horizontal transformations are performed using the flipping operation. The generated images from these transformations are merely used during batch training [42], [43]. These transformations are executed temporally in memory during runtime, but they are not saved to disk. These three extractor models are incorporated to establish a hybrid backbone for the weed classification model. The hybrid model can handle the overlapping occlusion and illumination conditions in real-world images. The details of the hybrid method are discussed in the subsequent sections.

D. PROPOSED METHOD IMPLEMENTATION

The experiments were implemented on a machine using Windows 10 64-bit as an operating system. The hardware components of this machine comprise an Intel Core i7-10 Gen. The primary memory size was 32 GB. The GPU was an RTX 2070 with 16 GB of memory. Python 3.7 with CUDA 10.1 was the programming language for developing the DL model. PyCharm was employed as the framework for coding. Python provides the entire package of both the Pip and Conda libraries. The proportion of training data to testing data is 80:20 samples. The initial values for the batch size, epoch size, and learning rate are 32, 10, and 10^{-4}, respectively. The framework of this approach is shown in Fig. 8.

Based on this figure, the processes involve three main stages, including data preparation, image pre-processing, and image classification, distributed into eight steps. The data preparation stage includes two steps, and the image pre-processing stage involves two steps. Finally, the feature extraction and image classification stage consists of two steps. An additional step is introduced to evaluate the classification results of the proposed model using quantitative measures. These are the summary descriptions of the functions of each step:

Step 1: Collecting samples from the source to be prepared for the training process.

Step 2: Dividing the dataset into 80% for the training set and 20% for the testing set.

Step 3: In the pre-processing stage, image resizing reduces the memory space and time-consuming execution. The proposed model converts all the images from various sizes to a specific size, which is 224 × 224 pixels as a standard size.

Step 4: Increasing the scale of the dataset using the data augmentation technique. This technique is used to increase dataset size and reduce overfitting. Several augmentation operations are utilized in this step, such as mirroring, flipping, zooming, and rotations.

Step 5: Training the three extractor models with all the annotated training. Individually, these extractor models are used to extract the features, so that the output of each model yields its classification results.

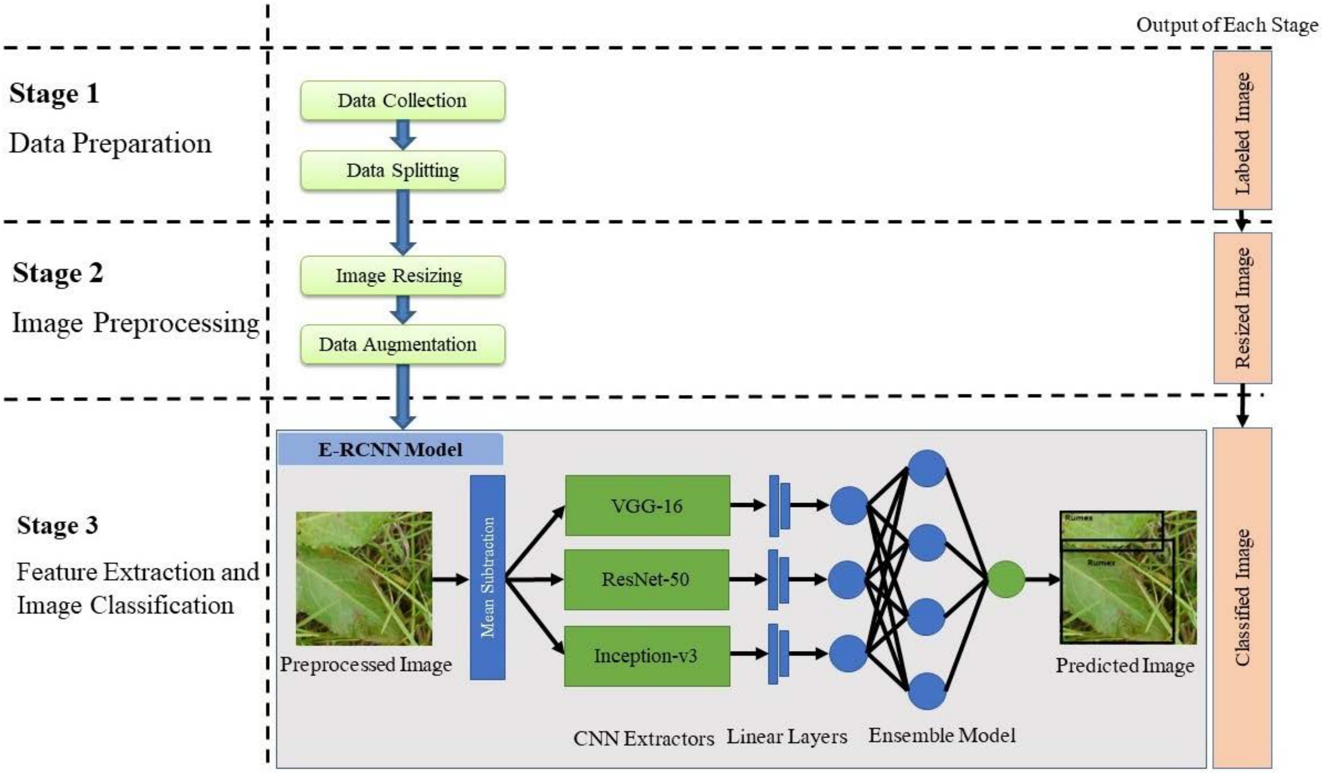


FIGURE 8. The entire design framework of the proposed model includes three stages: data collection, image preprocessing, and feature extraction and classification.

Step 6: Finally, all the outputs of the three extractor models are grouped into the ensemble model for voting by the majority. The result of the ensemble model yields one classified label as the predicted result.

Step 7: Evaluating the results of the proposed method using the performance metrics including precision, accuracy, recall, and F1-score for classification accuracy.

The concentration of this research would be on the classification stage to be the basis for weed detection. In the following sections, these steps are illustrated in more detail. Fig. 9 demonstrates the algorithmic step code of our proposed model.

E. RESULTS AND DISCUSSIONS

To test the efficacy of our proposed approach, we compared it with the previous competing studies. The effectiveness of the ML and DL methods is tested to measure the method's validity on a designated test problem. Similarly, the proposed method aims to improve classification accuracy. Fig. 10 and Fig. 11 illustrate the analysis of the training accuracy and loss error rate of the three backbone networks of this approach applied to two various *Rumex* datasets, including dataset 1 [23] and dataset 2 [17]. Fig. 10 and Fig. 11 compare the performances of three backbone networks, VGG-16, ResNet-50, and Inception-V3, during the training process for dataset 1 and dataset 2. It is observed that VGG-16 is more stable than Inception-V3 and ResNet-50 networks during the training process of dataset 1 and dataset 2. However, the performance of Inception-V3 decreased in both datasets at the final level of the training process. According to ResNet-50, it is monitored that the performance of this network increases sluggishly compared to other backbone networks. It requires ample time to be trained due to its dense layers. To analyze the error rate or loss of the three backbone networks, the pay attention is recorded for ResNet-50, which has a lower error rate than Inception-V3 and VGG-16. For VGG-16, however, the greater error rate is considered.

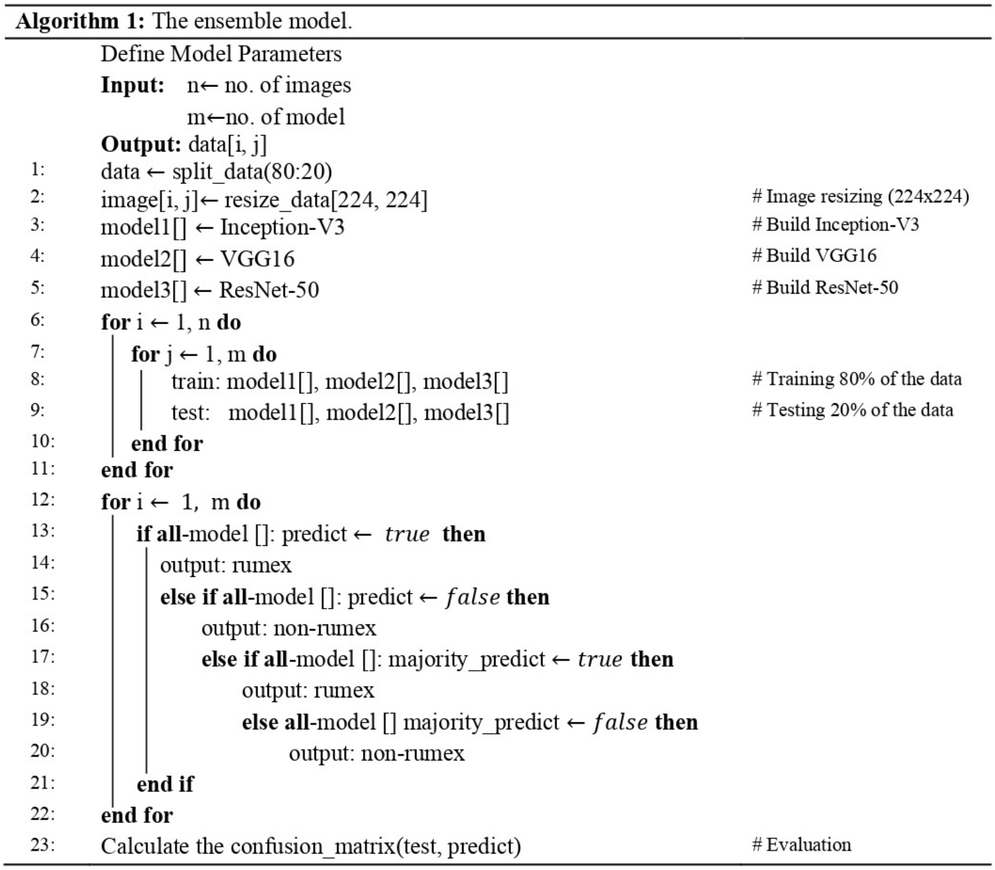
For evaluation, some well-known metrics such as accuracy, F1-score [44], precision, and recall [45] are employed to observe the effectiveness of the proposed method. In this research, we focus on using quantitative measurement to quantify the robustness of our proposed model. This set of metrics compares the predicted label with the ground-truth label in terms of accuracy [46], precision or PPV [21], recall [32], and F1-score [47]. The standard formula of accuracy, precession, recall, and F1-score are shown in the following Equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall\ \textbf{\textit{or}}\ Sensitivity\ \textbf{\textit{or}}\ TPR = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$



```
Algorithm 1: The ensemble model.
Define Model Parameters
Input:   n← no. of images
         m←no. of model
Output: data[i, j]
data ← split_data(80:20)
image[i, j]← resize_data[224, 224]                    # Image resizing (224x224)
model1[] ← Inception-V3                               # Build Inception-V3
model2[] ← VGG16                                      # Build VGG16
model3[] ← ResNet-50                                  # Build ResNet-50
for i ← 1, n do
    for j ← 1, m do
        train: model1[], model2[], model3[]           # Training 80% of the data
        test:  model1[], model2[], model3[]           # Testing 20% of the data
    end for
end for
for i ← 1, m do
    if all-model []: predict ← $true$ then
        output: rumex
        else if all-model []: predict ← $false$ then
            output: non-rumex
            else if all-model []: majority_predict ← $true$ then
                output: rumex
                else all-model [] majority_predict ← $false$ then
                    output: non-rumex
    end if
end for
Calculate the confusion_matrix(test, predict)         # Evaluation
```

FIGURE 9. The algorithm of the proposed model.

$$FP = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{8}$$

where:

TP: represents the total number of *Rumex* weeds classified by both images of ground truth and the proposed model.

FP: represents the total number of non-*Rumex* weeds (e.g., grass) that are not classified as ground truth images while they are recognized as *Rumex* through the proposed method.

FN: represents the total number of *Rumex* weeds that are recognized via the ground truth image and not recognized through the proposed model.

TN: represents the total number of non-*Rumex* that are not found in both the ground truth and the proposed model. After which, we compute the True-Positive Rate (TPR), and True-Negative Rate (TNR) to make a fair comparison using the confusion matrix.

F. COMPARISON TO PREVIOUS RUMEX CLASSIFICATION APPROACHES

Several hand-crafted and self-learning classification methods are compared to the proposed. These methods were applied to classify the *Rumex* weed plants from grass using real-world data. Table 3 illustrates the classification results of these methods using the standard evaluation metrics. These metrics are applied to verify that each true positive pixel in the Domain of Interest (DoI) has been precisely classified. For fair comparison, all representative methods were applied to the same tested data. The results in this table show that the Scale Invariant Feature Transform (SIFT) feature-based system [15] has the lowest accuracy, precision, and F1-score rates of all the tested techniques due to the high FPR. That means it is inefficient to determine the non-*Rumex* weeds correctly. Later, the SURF feature-based system [48] was proposed to overcome the previous method's shortage by lowering the FPR and False-Negative Rate (FNR), but it remains suffering from FPR sensitivity. The problem with those methods is that they used vectors to extract features, which are inefficient in identifying the negative objects due to the occlusion issue with *Rumex* weeds. Sünderhauf *et al.* [49] proposed Overfeat CNN for feature extraction with Extreme RF for classification. Such a method improves the system's recognition capability by reducing the FPR at a low rate. However, that method registered the highest FNR of all competing methods, classifying the true positive (TP) plant. Reyes *et al.* [50] used fine-tuned AlexNet [51] for weed recognition.

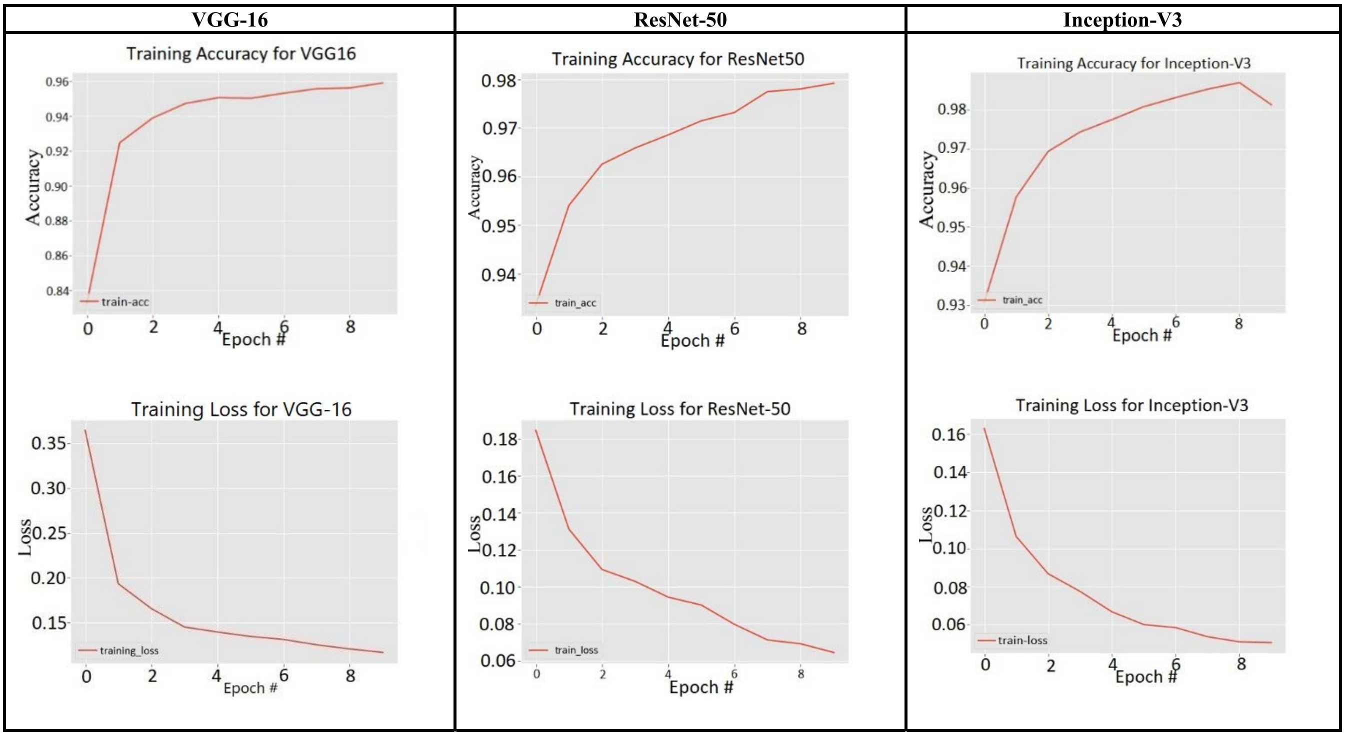


FIGURE 10. The comparison performance for each network of the proposed method over dataset 1.

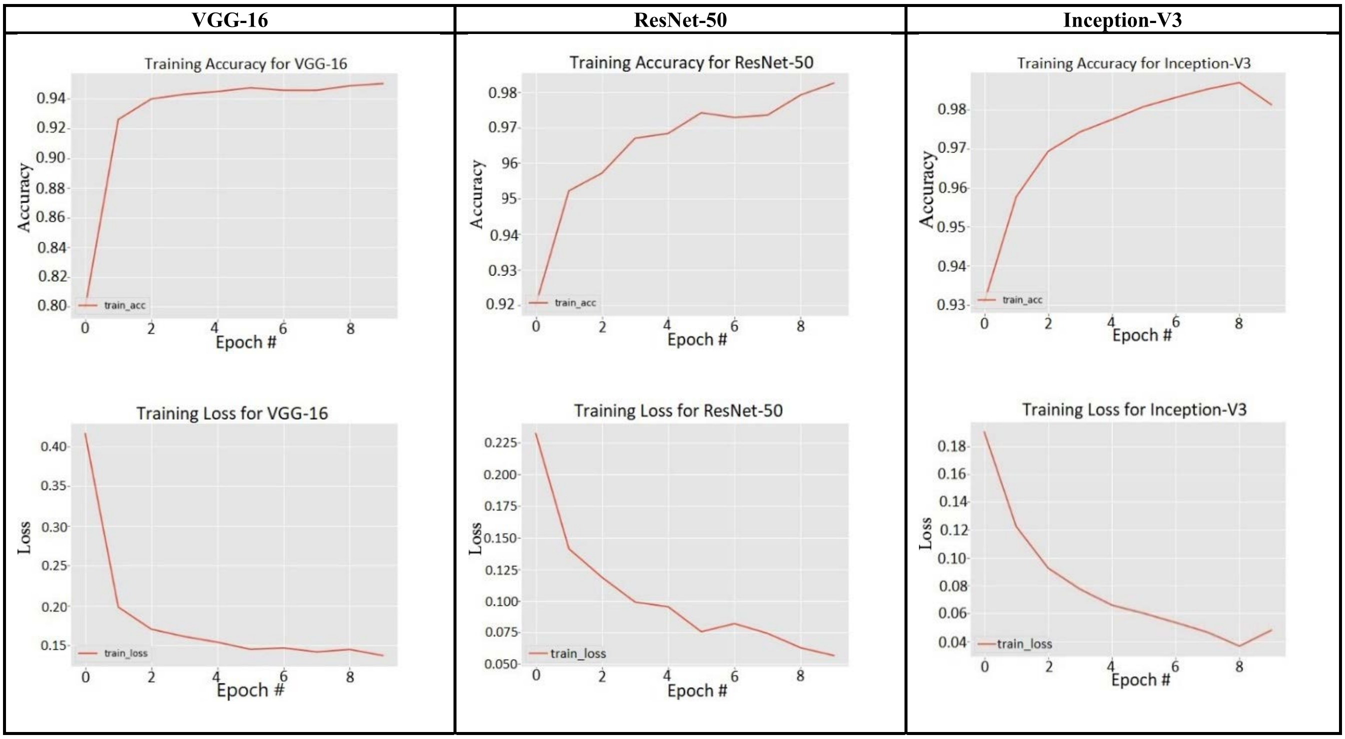


FIGURE 11. The comparison performance for each network of the proposed method over dataset 2.

To analyse the results in Table 3, it is observed that our proposed model achieved accurate classification results compared to competing methods. As shown in Table 3 and Table 4, our method achieved 97.02% accuracy using dataset 1 and 98% accuracy using dataset 2. Although AlexNet finetuning [50] and Overfeat with ExtremeTrees [49] achieved somewhat greater accuracy than our proposed model (1.36% and 1.90%, respectively), their results are relatively poor in

TABLE 3. Classification results of the various classification methods with our proposed model applied to dataset 1 in terms of accuracy, precision, recall, F1-score, FPR, and FNR (all as %).

Method Name	Accuracy	Precision	Recall	F1-score	FPR	FNR
SIFT feature-based system [15]	86.5± 0.8	4.5± 0.3	64.9± 1.4	16.832	13.3± 0.8	35.1± 1.4
SURF feature-based system [48]	88.2± 0.9	5.2± 0.4	66.4± 2.1	19.29	11.6± 0.9	33.6± 2.1
Overfeat+ExtremeTrees [49]	98.93± 0.03	36.02± 2.14	17.25± 0.71	23.328	0.29± 0.04	82.75± 0.71
VGG-VD 16+L1regLogReg	93.98± 0.21	10.71± 0.31	73.11± 0.94	37.366	5.82± 0.22	26.89± 0.94
Resnet 101+L2regLogReg	95.6± 0.14	11.7± 0.22	55.1± 0.84	38.6	4.00± 0.10	44.9± 0.83
Alexnet+L2regLogReg	96.02± 0.16	12.32± 0.45	52.38± 0.71	39.896	3.56± 0.16	47.62± 0.71
Inception_v1+L2regLogReg [23]	96.13± 0.11	15.48± 0.38	69.48± 0.71	25.3	3.62± 0.12	30.52± 0.71
Resnet 50+L2regLogReg	96.1± 0.13	14.8± 0.41	65.3± 1.20	48.2	3.6± 0.14	34.7± 1.23
VGG-F+L2regL2lossSVCp	96.80± 0.12	14.36± 0.39	47.98± 1.17	44.208	2.73± 0.13	52.02± 1.17
Alexnet fine-tuning [50]	98.36± 0.45	29.88± 5.53	47.90± 8.99	36.8	1.16± 0.53	52.10± 8.99
Proposed Model	**97.02**	**96.83**	**96.89**	**96.86**	**0.02± 0.01**	**0.02± 0.01**

terms of precision, recall, and F1-score metrics. However, the proposed model delivered on its promises by recording 81.35%, 27.41%, and 71.56% high difference rates on precision, recall, and F1-score, respectively. The shortcomings of the compared methods are due to the lack of addressing the challenging scenario and focusing on well-separated leaves or plants in the scene. Specifically, the limitation of these methods frequently occurs due to insufficient learning to classify occlusion and overlapped cases [52]. Furthermore, some methods depend on specific features such as shape or texture, which are not adequate to recognize the type of object [53], [54]. Concretely, the empirical results demonstrate, in overall evaluation metrics, that our proposed model provides a higher baseline accuracy than existing methods. Due to the diverse architectural designs of each network in our model, different features are yielded. These features play a crucial role in identifying our complex scenario. The finding of this work is that using a hybrid model produces a higher baseline accuracy against occlusion than using a single method. In some occlusion cases, however, our model showed low performance, especially when there are multi-occluded cases of *Rumex* in the same scene and due to the low-resolution imagery. Another challenging issue is observed when the scene contains a part of *Rumex* leaves distributed on the boundary where most features are absent.

Both Kounalakis *et al.* [23] in their Inception-V1 with L2-regularized with L2-loss logistic regression (L2regLogReg) and Reyes *et al.* [50] in their AlexNet method used the same training parameters by setting 10 to the learning rate for their classifiers. Reyes *et al.* [50] reduced the FPR to raise the recall ratio. At the same time, the FNR of such method does not produce sufficient results to recognize the true positive pixels due to overfitting. It is observed that there are unbalanced results in the evaluation metrics of the same method. The accuracy of most compared methods is high, while the F1-score metric reported low-rate values. In this regard, this proposed method achieves stable performance using the same standard metrics. The first evaluation was applied to the [23] data. In their work, Kounalakis *et al.* [23] demonstrated that the Inception-V1 with the L2regLogReg approach achieved the highest accuracy compared to other representative methods. We compute the F1-measure rate of the Inception-V1+L2regLogReg approach and other compared methods to be evaluated with our proposed model. Based on the analysis, we investigated that some metrics such as recall have high sensitivity to true negative pixels due to the high disparity between positive and negative pixels. Therefore, the second experiment is applied to the second set of *Rumex* data [17] as presented in Table 4.

Van Evert *et al.* [17] used 2-D Fourier analysis in their generated data from the above table. This method achieved 82%-89% acceptable scores using the accuracy metric, while it has not been tested on other evaluation metrics. In addition, their method is not being compared with other methods. In this work, we implemented several methods to measure the performance of our work and the 2-D Fourier analysis method. The comparative results of the competing approaches in terms of FPR and FNR of dataset 1 are shown in Fig. 12. According to this figure, the SIFT feature-based system has the greatest FPR rate, while the Overheat-Extreme Trees technique has the highest FNR. The lowest FPR and FNR, on the other hand, attained the key target by recording

TABLE 4. Classification results of the various classification methods with our proposed model applied to dataset 2 in terms of accuracy, precision, recall, F1-score, FPR, and FNR (all as %).

Method Name	Accuracy	Precision	Recall	F1-score	FPR	FNR
SVM_Poly1* [55]	88	n/a	n/a	n/a	1±1	1±1
2-D Fourier analysis [17]	82-89	n/a	n/a	n/a	n/a	n/a
SVM_Sigmoid [56]	91	86	36	50	64.3±0.63	63±0.63
RF	93	93	43	58	57.3± 0.56	57.3± 0.56
SVM_Linear [57]	92	86	45	59	54.5± 0.53	54.5± 0.53
SVM_RBF [58]	94	91	52	67	47± 0.46	47.5± 0.46
Proposed Model	**98**	**92**	**98**	**95**	**0± 0.0009**	**1± 0.009**

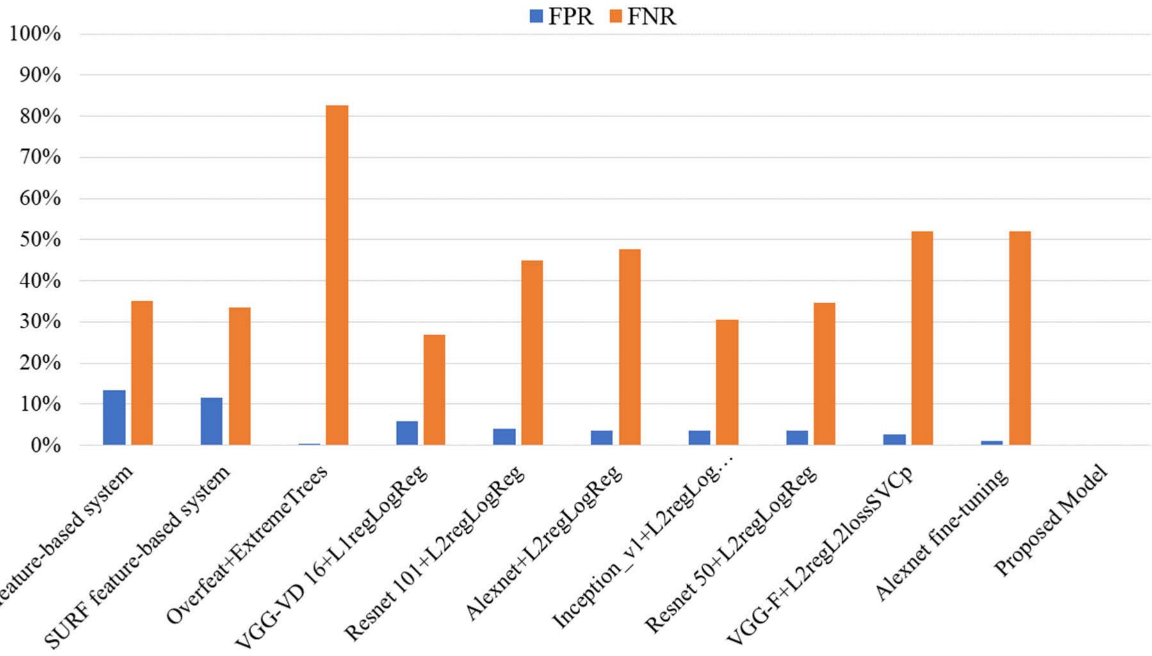


FIGURE 12. The classification results of the compared methods with our proposed model applied to dataset 1 in terms of FPR and FNR metrics.

0.02% on both measures when using the proposed model. That implies the application does not waste time processing the non-existent *Rumex* or misclassifying the actual *Rumex* in reality. The comparative results of the competing approaches in terms of FPR and FNR of dataset 2 are shown in Fig. 13. This figure illustrates that the SVM using a polynomial function scored the highest FPR and FNR of all the competing methods, while its classification accuracy is similar to that of 2-D Fourier analysis. On the other hand, we applied SVM with Gaussian Radial Basis Function (RBF) to the same tested data to achieve the highest accuracy, precision, recall, and F1-score rates of all the competing methods. However, the FPR and FNR of the SVM_RBF are high due to poor images and occlusion, making it inappropriate to be utilized with a robust detection model. Nevertheless, the results show that the proposed model outperforms by 4%, 1%, 46%, and 28% higher results than the best-compared methods on the accuracy, precision, recall, and F1-measure, respectively. Furthermore, its FPR and FNR are tiny to identify the RoI and effectively misclassify unwanted regions. This outperforming leads to the fact that this approach is promising for a new detection model. Table 5 details the comparison results of our proposed methods using two different datasets. On the other hand, the accuracy rate has the lowest rate of the two datasets, with dataset 2 having a 0.98% higher rate. Fig. 14 and Fig. 15 depict the confusion matrix of dataset 1 and

*Polynomial Kernel Function.

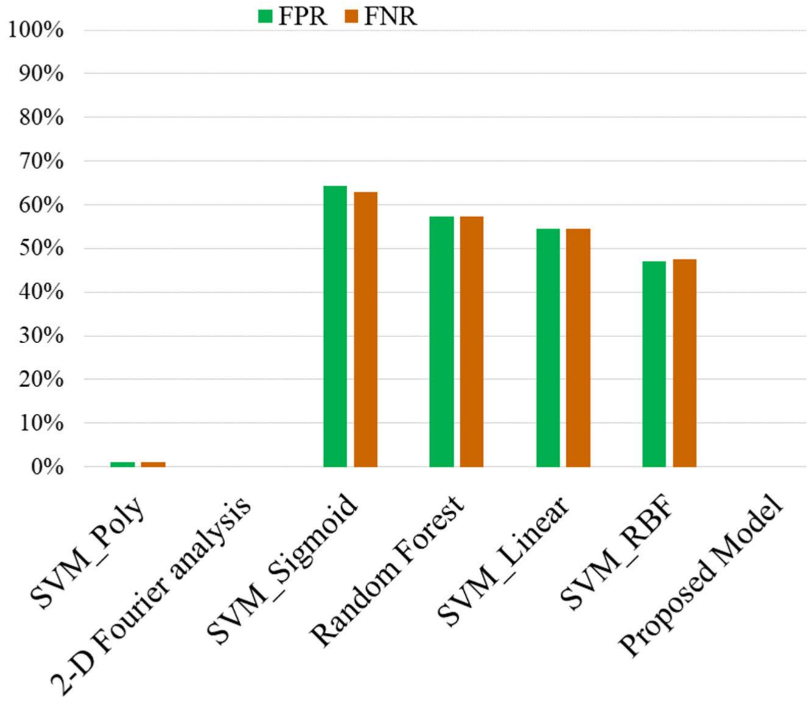
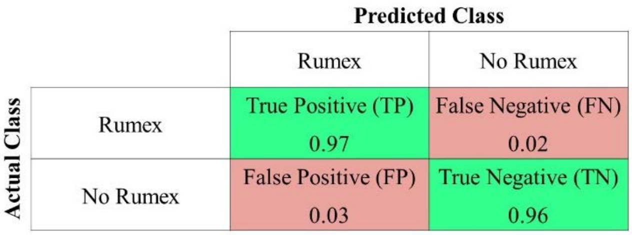
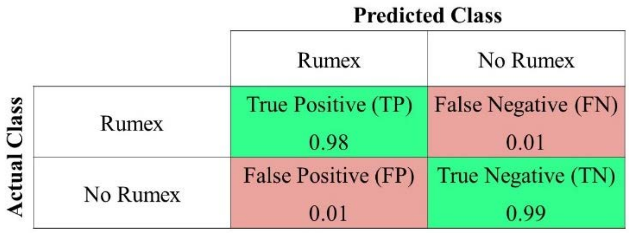


FIGURE 13. The classification results of the compared methods with our proposed model applied to dataset 2 in terms of FPR and FNR metrics.



		Predicted Class	
		Rumex	No Rumex
Actual Class	Rumex	True Positive (TP) 0.97	False Negative (FN) 0.02
	No Rumex	False Positive (FP) 0.03	True Negative (TN) 0.96

FIGURE 14. The confusion matrix of our proposed model applied to dataset 1.



		Predicted Class	
		Rumex	No Rumex
Actual Class	Rumex	True Positive (TP) 0.98	False Negative (FN) 0.01
	No Rumex	False Positive (FP) 0.01	True Negative (TN) 0.99

FIGURE 15. The confusion matrix of our proposed model applied to dataset 2.

dataset 2, respectively. The preliminary results of the proposed model on precision and F1-score in dataset 1 are higher than those in dataset 2. In contrast, the proposed model is higher in accuracy and recall when using dataset 2. Thus, our finding is a balance between these two datasets that could be merged into one dataset. In addition, our method achieved high accuracy (97%-98%), which is higher than the individual methods to classify *Rumex* in different conditions. However, it does not concentrate on a single issue like illumination, as used in Zhang *et al.* [20] work.

TABLE 5. Comparison results of our proposed model applied to both datasets in terms of accuracy, precision, recall, F1-score, FPR, and FNR (all in percent).

Dataset Name	Acc.	Prec.	Recall	F1-score
Dataset 1		✓		✓
Dataset 2	✓		✓	

V. CONCLUSION

Rumex. is a vital weed plant that has a substantial effect on dairy yield and production. Real-world images such as illumination, overlapping, and occlusion reduce the accuracy of the classification model. These issues are considered challenging task to computer vision. Most previous works focused on weed classification under controlled conditions, whereas weeds are certainly allocated in grassland under the real-world conditions of farms. In this study, a new hybrid CNN model with three various extractors at the backbone is proposed to improve classification accuracy in real-world data. Unlike the single approach, combining three different networks into one ensemble model increases the ability to extract deepening (e.g., additional beneficial) features due to the variety of architectural designs for each network. In addition, each extractor provides the ability to address one or more challenging real-world issues so that the shortcomings of each network are addressed by the two remaining networks. Experimental results show that utilizing different extractor networks was able to reduce the FPR and FNR to a low-level rate. Compared to other recent models, this reduction helps generalize the model with unseen fields to classify *Rumex* in real-world conditions.

This work uses a standard benchmark dataset of images captured under real-world conditions. Images in these two datasets are captured in challenging conditions of a real farm, such as various illumination, occlusion, and overlapping conditions. In addition, each image includes single or multi-leaves or entire *Rumex* weed plants. The proposed approach has been compared and evaluated using the same dataset with different methods. The results have shown that the proposed approach produces better results than other competing methods. The total averages of this approach on both datasets are 97.51%, 94.41%, 97.44%, and 95.93% using accuracy, precision, recall, and F1-score, respectively.

This work introduces pivotal knowledge to the computer-vision community. Firstly, it improves the classification methods for *Rumex* in real-world conditions by using a combination of three different classifiers. Regarding the agricultural community, this research can be implemented in a weed management system or an automated weed spraying system. It assists the farmer in alleviating labor-intensive costs, reducing time-consuming tasks, preventing herbicide pollution in the environment, and controlling weed separation.

VI. LIMITATION AND FUTURE DIRECTIONS

Real-world data is a challenging issue in computer vision approaches. The limitation of this data is the deficiency of beneficial information in the occluded and overlapped regions. Increasing these regions adversely impacts the classification accuracy of the results. However, using sufficient samples in the training of DL raises the model's potential for extracting and classifying. For future work, these two datasets can be combined to increase the number of samples, especially those for the entire *Rumex* plant in the grass, due to the limited amount. In addition, this work can be expanded to produce a new detection model focusing on *Rumex* weed plants using You Only Look Once (YOLO) detection and Mask Region-Convolutional Neural Network (R-CNN). Moreover, we will investigate the restrictions on why other networks achieve high performance in their related tasks as compared with our data. Besides, we plan to apply our proposed model to classify the diseases and lesions of *Rumex* or other weed species.

ACKNOWLEDGMENT

The authors thank the Universiti Teknologi of Malaysia (UTM) library for providing their materials and database. They also would like to thank PSU for providing technical support leading to the publication of this research.

REFERENCES

[1] F. K. van Evert, J. Samsom, G. Polder, M. Vijn, H.-J.-V. Dooren, A. Lamaker, G. W. A. M. van der Heijden, C. Kempenaar, T. van der Zalm, and L. A. P. Lotz, "A robot to detect and control broad-leaved dock (Rumex obtusifolius L.) in grassland," *J. Field Robot.*, vol. 28, no. 2, pp. 264–277, Mar. 2011, doi: 10.1002/rob.20377.

[2] A. H. Al-Badri, N. A. Ismail, K. Al-Dulaimi, G. A. Salman, A. R. Khan, A. Al-Sabaawi, and M. S. H. Salam, "Classification of weed using machine learning techniques: A review-challenges, current and future potential techniques," *J. Plant Diseases Protection*, vol. 129, no. 4, pp. 745–768, Aug. 2022, doi: 10.1007/s41348-022-00612-9.

[3] E. Hamuda, M. Glavin, and E. Jones, "A survey of image processing techniques for plant extraction and segmentation in the field," *Comput. Electron. Agricult.*, vol. 125, pp. 184–199, Jul. 2016, doi: 10.1016/j.compag.2016.04.024.

[4] K. Osorio, A. Puerto, C. Pedraza, D. Jamaica, and L. Rodríguez, "A deep learning approach for weed detection in lettuce crops using multispectral images," *AgriEngineering*, vol. 2, no. 3, pp. 471–488, Aug. 2020, doi: 10.3390/agriengineering2030032.

[5] S. Shorewala, A. Ashfaque, R. Sidharth, and U. Verma, "Weed density and distribution estimation for precision agriculture using semi-supervised learning," *IEEE Access*, vol. 9, pp. 27971–27986, 2021, doi: 10.1109/access.2021.3057912.

[6] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, "Machine learning applications for precision agriculture: A comprehensive review," *IEEE Access*, vol. 9, pp. 4843–4873, 2021, doi: 10.1109/ACCESS.2020.3048415.

[7] A. Binch and C. W. Fox, "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland," *Comput. Electron. Agricult.*, vol. 140, pp. 123–138, Aug. 2017, doi: 10.1016/j.compag.2017.05.018.

[8] J. Jia, G. W. Krutz, and H. W. Gibson, "Corn plant locating by image processing," *Proc. SPIE*, vol. 1379, pp. 246–253, Feb. 1991, doi: 10.1117/12.25095.

[9] E. Franz, M. Gebhardt, and K. Unklesbay, "Shape description of completely visible and partially occluded leaves for identifying plants in digital images," *Trans. ASAE*, vol. 34, no. 2, pp. 673–0681, Apr. 1991, doi: 10.13031/2013.31716.

[10] L. Tian, "Knowledge based machine vision system for outdoor plant identification," Ph.D. dissertation, Dept. Biol. Agricult. Eng., Univ. California, Davis, Davis, CA, USA, ProQuest Dissertations, 1995. [Online]. Available: https://dl.acm.org/doi/10.5555/922195

[11] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen, "Shape features for identifying young weeds using image analysis," *Trans. Amer. Soc. Agricult. Eng.*, vol. 38, no. 1, pp. 271–281, 1995, doi: 10.13031/2013.27839.

[12] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Comput. Electron. Agricult.*, vol. 158, pp. 226–240, Mar. 2019, doi: 10.1016/j.compag.2019.02.005.

[13] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, pp. 1–21, May 2021, doi: 10.1007/s42979-021-00592-x.

[14] M. Toğaçar, B. Ergen, and Z. Cömert, "Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models," *Measurement*, vol. 158, Jul. 2020, Art. no. 107703, doi: 10.1016/j.measurement.2020.107703.

[15] T. Kounalakis, G. A. Triantafyllidis, and L. Nalpantidis, "Weed recognition framework for robotic precision farming," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2016, pp. 466–471, doi: 10.1109/IST.2016.7738271.

[16] L. Dürr, T. Anken, H. Bollhalder, J. Sauter, K. Burri, and D. Kuhn, "Machine vision detection and microwave based elimination of Rumex obtusifolius L. on grassland," in *Proc. 5th Eur. Conf. Precis. Agricult. (ECPA)*. Uppsala, Sweden, 2005, p. 5.

[17] F. K. Van Evert, G. Polder, G. W. A. M. Van Der Heijden, C. Kempenaar, and L. A. P. Lotz, "Real-time vision-based detection of rumex obtusifolius in grassland," *Weed Res.*, vol. 49, no. 2, pp. 164–174, Apr. 2009, doi: 10.1111/j.1365-3180.2008.00682.x.

[18] J. Gao, D. Nuyttens, P. Lootens, Y. He, and J. G. Pieters, "Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery," *Biosyst. Eng.*, vol. 170, pp. 39–50, Jun. 2018, doi: 10.1016/j.biosystemseng.2018.03.006.

[19] T. Kounalakis, G. A. Triantafyllidis, and L. Nalpantidis, "Image-based recognition framework for robotic weed control systems," *Multimedia Tools Appl.*, vol. 77, no. 8, pp. 9567–9594, Apr. 2018, doi: 10.1007/s11042-017-5337-y.

[20] W. Zhang, M. F. Hansen, T. N. Volonakis, M. Smith, L. Smith, J. Wilson, G. Ralston, L. Broadbent, and G. Wright, "Broad-leaf weed detection in pasture," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 101–105, doi: 10.1109/ICIVC.2018.8492831.

[21] J. Valente, M. Doldersum, C. Roers, and L. Kooistra, "Detecting rumex obtusifolius weed plants in grasslands from UAV RGB imagery using deep learning," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 4, pp. 179–185, May 2019, doi: 10.5194/isprs-annals-IV-2-W5-179-2019.

[22] O. H. Y. Lam, M. Dogotari, M. Prüm, H. N. Vithlani, C. Roers, B. Melville, F. Zimmer, and R. Becker, ''An open source workflow for weed mapping in native grassland using unmanned aerial vehicle: Using rumex obtusifolius as a case study,'' *Eur. J. Remote Sens.*, vol. 54, no. 1, pp. 71–88, Feb. 2021, doi: 10.1080/22797254.2020.1793687.

[23] T. Kounalakis, G. A. Triantafyllidis, and L. Nalpantidis, ''Deep learning-based visual recognition of rumex for robotic precision farming,'' *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104973, doi: 10.1016/j.compag.2019.104973.

[24] H. R. Bonab and F. Can, ''A theoretical framework on the ideal number of classifiers for online ensembles in data streams,'' in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 2053–2056, doi: 10.1145/2983323.2983907.

[25] L. I. Kuncheva and C. J. Whitaker, ''Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,'' *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, May 2003, doi: 10.1023/A:1022859003006.

[26] M. Dyrmann, H. Karstoft, and H. S. Midtiby, ''Plant species classification using deep convolutional neural network,'' *Biosyst. Eng.*, vol. 151, pp. 72–80, Nov. 2016, doi: 10.1016/j.biosystemseng.2016.08.024.

[27] B. Espejo-Garcia, N. Mylonas, L. Athanasakos, S. Fountas, and I. Vasilakoglou, ''Towards weeds identification assistance through transfer learning,'' *Comput. Electron. Agricult.*, vol. 171, Apr. 2020, Art. no. 105306, doi: 10.1016/j.compag.2020.105306.

[28] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, ''Unsupervised deep learning and semi-automatic data labeling in weed discrimination,'' *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104963, doi: 10.1016/j.compag.2019.104963.

[29] A. Kamilaris and F. X. Prenafeta-Boldú, ''Deep learning in agriculture: A survey,'' *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Aug. 2018, doi: 10.1016/j.compag.2018.02.016.

[30] A.-A. Binguitcha-Fare and P. Sharma, ''Crops and weeds classification using convolutional neural networks via optimization of transfer learning parameters,'' *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5, pp. 2249–8958, Jun. 2019.

[31] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. R. Azghadi, and R. D. White, ''DeepWeeds: A multiclass weed species image dataset for deep learning,'' *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Feb. 2019, doi: 10.1038/s41598-018-38343-3.

[32] H. Jiang, C. Zhang, Y. Qiao, Z. Zhang, W. Zhang, and C. Song, ''CNN feature based graph convolutional network for weed and crop recognition in smart farming,'' *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105450, doi: 10.1016/j.compag.2020.105450.

[33] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*. San Diego, CA, USA, May 2015, pp. 1–14.

[34] S. I. Moazzam, U. S. Khan, W. S. Qureshi, M. I. Tiwana, N. Rashid, W. S. Alasmary, J. Iqbal, and A. Hamza, ''A patch-image based classification approach for detection of weeds in sugar beet crop,'' *IEEE Access*, vol. 9, pp. 121698–121715, 2021, doi: 10.1109/access.2021.3109015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/cvpr.2016.308.

[37] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. de Geus, ''Malicious software classification using transfer learning of ResNet-50 deep neural network,'' in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1011–1014, doi: 10.1109/ICMLA.2017.00-19.

[38] M. Lin, Q. Chen, and S. Yan, ''Network in network,'' 2013, *arXiv:1312.4400*.

[39] X. Jin, J. Che, and Y. Chen, ''Weed identification using deep learning and image processing in vegetable plantation,'' *IEEE Access*, vol. 9, pp. 10940–10950, 2021, doi: 10.1109/access.2021.3050296.

[40] A. K. Rangarajan and R. Purushothaman, ''Disease classification in eggplant using pre-trained VGG16 and MSVM,'' *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020, doi: 10.1038/s41598-020-59108-x.

[41] A. Jahanbakhshi, M. Momeny, M. Mahmoudi, and P. Radeva, ''Waste management using an automatic sorting system for carrot fruit based on image processing technique and improved deep neural networks,'' *Energy Rep.*, vol. 7, pp. 5248–5256, Nov. 2021, doi: 10.1016/j.egyr.2021.08.028.

[42] A. Lin, J. Wu, and X. Yang, ''A data augmentation approach to train fully convolutional networks for left ventricle segmentation,'' *Magn. Reson. Imag.*, vol. 66, pp. 152–164, Feb. 2020, doi: 10.1016/j.mri.2019.08.004.

[43] F. Mohammadimanesh, B. Salehi, M. Mahdianpari, E. Gill, and M. Molinier, ''A new fully convolutional neural network for semantic segmentation of polarimetric SAR imagery in complex land cover ecosystem,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 223–236, May 2019, doi: 10.1016/j.isprsjprs.2019.03.015.

[44] C. Wang, X. Peng, M. Liu, Z. Xing, X. Bai, B. Xie, and T. Wang, ''A learning-based approach for automatic construction of domain glossary from source code and documentation,'' in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Aug. 2019, pp. 97–108, doi: 10.1145/3338906.3338963.

[45] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, ''Automatic crop detection under field conditions using the HSV colour space and morphological operations,'' *Comput. Electron. Agricult.*, vol. 133, pp. 97–107, Feb. 2017, doi: 10.1016/j.compag.2016.11.021.

[46] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, ''Review on convolutional neural networks (CNN) in vegetation remote sensing,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021, doi: 10.1016/j.isprsjprs.2020.12.010.

[47] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, ''Fully convolutional networks with sequential information for robust crop and weed detection in precision farming,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2870–2877, Oct. 2018, doi: 10.1109/LRA.2018.2846289.

[48] T. Kounalakis, G. A. Triantafyllidis, and L. Nalpantidis, ''Vision system for robotized weed recognition in crops and Grasslands,'' in *Proc. Int. Conf. Comput. Vis. Syst. (ICCV)*, vol. 10528. Cham, Switzerland: Springer, Oct. 2017, pp. 485–498, doi: 10.1007/978-3-319-68345-4_43.

[49] N. Sünderhauf, C. McCool, B. Upcroft, and T. Perez, ''Fine-grained plant classification using convolutional neural networks for feature extraction,'' in *Proc. CLEF Working Notes*, 2014, pp. 756–762.

[50] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, ''Fine-tuning deep convolutional networks for plant recognition,'' in *Proc. CLEF Working Notes*, vol. 1391, 2015, pp. 467–475.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[52] A. Bakhshipour, A. Jafari, S. M. Nassiri, and D. Zare, ''Weed segmentation using texture features extracted from wavelet sub-images,'' *Biosyst. Eng.*, vol. 157, pp. 1–12, May 2017, doi: 10.1016/j.biosystemseng.2017.02.002.

[53] S. Abouzahir, M. Sadik, and E. Sabir, ''Enhanced approach for weeds species detection using machine vision,'' in *Proc. Int. Conf. Electron., Control, Optim. Comput. Sci. (ICECOCS)*, Dec. 2018, pp. 1–6, doi: 10.1109/ICECOCS.2018.8610505.

[54] N. Li, X. Zhang, C. Zhang, L. Ge, Y. He, and X. Wu, ''Review of machine-vision-based plant detection technologies for robotic weeding,'' in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 2370–2377, doi: 10.1109/robio49542.2019.8961381.

[55] J. You, W. Liu, and J. Lee, ''A DNN-based semantic segmentation for detecting weed and crop,'' *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105750, doi: 10.1016/j.compag.2020.105750.

[56] A. Bakhshipour and A. Jafari, ''Evaluation of support vector machine and artificial neural networks in weed detection using shape features,'' *Comput. Electron. Agricult.*, vol. 145, pp. 153–160, Feb. 2018, doi: 10.1016/j.compag.2017.12.032.

[57] T. Kounalakis, M. J. Malinowski, L. Chelini, G. A. Triantafyllidis, and L. Nalpantidis, ''A robotic system employing deep learning for visual recognition and detection of weeds in grasslands,'' in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2018, pp. 1–6, doi: 10.1109/IST.2018.8577153.

[58] F. Ahmed, M. H. Kabir, S. Bhuyan, H. Bari, and E. Hossain, ''Automated weed classification with local pattern-based texture descriptors,'' *Int. Arab J. Inf. Technol.*, vol. 11, no. 1, pp. 87–94, Jan. 2014.