Article

Metaheuristic Optimization for Improving Weed Detection in Wheat Images Captured by Drones

El-Sayed M. El-Kenawy [1,*], Nima Khodadadi [2], Seyedali Mirjalili [3,4,*], Tatiana Makarovskikh [5], Mostafa Abotaleb [5], Faten Khalid Karim [6], Hend K. Alkahtani [7,*], Abdelaziz A. Abdelhamid [8], Marwa M. Eid [9], Takahiko Horiuchi [10], Abdelhameed Ibrahim [11] and Doaa Sami Khafaga [6]

[1]   Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
[2]   Department of Civil and Environmental Engineering, Florida International University, Miami, FL, USA
[3]   Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia
[4]   Yonsei Frontier Lab, Yonsei University, Seoul 03722, Republic of Korea
[5]   Department of System Programming, South Ural State University, Chelyabinsk, Russia
[6]   Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[7]   Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[8]   Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
[9]   Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, Egypt
[10] Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
[11] Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
*   Correspondence: skenawy@ieee.org (E.-S.M.E.-K.); ali.mirjalili@torrens.edu.au (S.M.); hkalqahtani@pnu.edu.sa (H.K.A.)

Abstract: Background and aim: Machine learning methods are examined by many researchers to identify weeds in crop images captured by drones. However, metaheuristic optimization is rarely used in optimizing the machine learning models used in weed classification. Therefore, this research targets developing a new optimization algorithm that can be used to optimize machine learning models and ensemble models to boost the classification accuracy of weed images. Methodology: This work proposes a new approach for classifying weed and wheat images captured by a sprayer drone. The proposed approach is based on a voting classifier that consists of three base models, namely, neural networks (NNs), support vector machines (SVMs), and K-nearest neighbors (KNN). This voting classifier is optimized using a new optimization algorithm composed of a hybrid of sine cosine and grey wolf optimizers. The features used in training the voting classifier are extracted based on AlexNet through transfer learning. The significant features are selected from the extracted features using a new feature selection algorithm. Results: The accuracy, precision, recall, false positive rate, and kappa coefficient were employed to assess the performance of the proposed voting classifier. In addition, a statistical analysis is performed using the one-way analysis of variance (ANOVA), and Wilcoxon signed-rank tests to measure the stability and significance of the proposed approach. On the other hand, a sensitivity analysis is performed to study the behavior of the parameters of the proposed approach in achieving the recorded results. Experimental results confirmed the effectiveness and superiority of the proposed approach when compared to the other competing optimization methods. The achieved detection accuracy using the proposed optimized voting classifier is 97.70%, F-score is 98.60%, specificity is 95.20%, and sensitivity is 98.40%. Conclusion: The proposed approach is confirmed to achieve better classification accuracy and outperforms other competing approaches.

Keywords: smart farming; metaheuristic optimization; weed detection; machine learning; sine cosine algorithm; grey wolf optimization algorithms

1. Introduction

Climate change and worldwide population expansion are exerting significant pressure on agriculture to expand food production in terms of quality and quantity. Because the global population is expected to grow to nine billion people by 2050, agricultural production will need to quadruple to keep up [1]. Plant diseases, pests, and weed infestation pose enormous problems to agriculture [2–5]. Weeds are unwelcome plants that take nutrients from the soil, compete with profitable crops for light, water, and other resources, and spread by seeds or rhizomes. Weeds, pests, and diseases diminish crop yields and quality, reducing the amount of food, fiber, and biofuel that can be produced. Losses might be sudden or long-term, but on average, 42% of a few key food crops' productions is lost.

To achieve reasonable weed control and increased crop output, farmers invest billions of dollars every year in weed management. It is, therefore, critical to managing weeds in horticulture crops, as failure to do such results in lower yields and product quality [6]. If not handled properly, the employment of chemical and cultural control methods might negatively affect the ecosystem. Weed control will be more successful and long-lasting with low-cost technology for identifying and mapping weeds early in their life cycle. Crop diseases and pests can be reduced, and crop yields can be increased by as much as 34% when early weed management is used [7]. Weeds may be managed in various ways, all of which take environmental considerations into account. Image processing is one of the most promising of these methods. Unmanned aerial vehicles (UAVs) are used in image processing to monitor crops and capture images of potential weeds in the fields. Due to their capacity to cover enormous areas quickly, UAVs have been proven to be helpful in agriculture because they do not create soil compaction or damage in the fields [8]. It is still a challenge to turn data gathered by UAVs into relevant information. Due to the manual labor required for segment size tweaking, feature selection, and rule-based classifier building, conventional data gathering and classification cannot be automated.

With the goal of increasing crop productivity while decreasing the prevalence of unwanted weeds, agricultural mechanization has emerged as the leading research field [9]. The intelligent spraying system relies heavily on accurately identifying weed plants to maximize agricultural yield [10]. Many machine learning-based algorithms have been developed for weed identification, making it a promising area of study for data scientists [11]. Many scientists used computer vision algorithms to categorize agricultural, and weed plants [12]. Various deep learning and hand-crafted models have also been published and have made substantial contributions [13]. Color classification strategies for perennial weed identification [12], CNN-based method/approach for distinguishing sugar beet plants from weeds [14], deep convolutional neural network (DNN) [15], Gabor wavelets and neural network [16], hyperspectral imaging with wavelet analysis [17], decision trees, and artificial neural network [18] have all been proposed for the classification of weeds. The agriculture industry has benefited greatly from these strategies, which have produced extraordinary results. Better weed plant classification, however, requires more advanced and efficient methodologies to boost the accuracy of weed detection.

In this work, a publicly available wheat images dataset is employed as the overarching inspiration for this research. This dataset is utilized for training a deep neural network through transfer learning and feature extraction. In addition, to boost the classification accuracy of weed images, a new optimization algorithm is proposed to optimize the parameters of a new voting ensemble classifier composed of neural network (NN), k-nearest neighbors (KNN), and support vector machine (SVM) machine learning models. Moreover, a binary optimizer is proposed to optimize the feature selection process to select the best set of features. To evaluate the performance of the proposed methodology, a set of evaluation criteria is adopted to assess the effectiveness of the feature selection algorithm

and an optimized voting ensemble model. On the other hand, statistical tests, such as the one-way analysis of variance and the Wilcoxon signed-rank test, are conducted to evaluate the significance and statistical difference of the proposed methodology. The recorded results are compared to those of other algorithms to show the superiority of the proposed approach.

This paper is structured in terms of five sections. Section 1 presents the introduction of the problem addressed in this paper and a summary of the proposed solution. Section 2 discusses the main milestones in the literature related to the task of weed detection. The materials and methods employed in the proposed solution are presented in Section 3. The proposed algorithms and solutions are explained in Section 4, and the experimental results are discussed in Section 5. Finally, the conclusions are presented in Section 6.

2. Literature Review

Weed identification using machine learning and image analysis has been increasingly popular in recent years, and the research presented here examines some of the most notable examples. Weed maps may be generated using several classification methods using UAV images [19–22]. However, recent state-of-the-art publications [23] reveal that machine learning algorithms are superior to traditional parametric methods in terms of accuracy and efficiency when dealing with complex data. The random forest (RF) classifier is one of the most well-liked machine learning algorithms for use in remote sensing [24]. This is because of its high generalization performance and fast processing time. The classification of high-resolution UAV images and agricultural mapping with RF has been proven beneficial. SVM is another well-known machine learning classifier [25–28], and it has been widely used to categorize weeds and crops. Meanwhile, the authors in [29] employed the KNN method to identify spreading thistles in sugar beet fields. New efforts on a machine learning-based approach for weed detection are summarized in Table 1.

Table 1. Weed detection approaches published in the literature.

Reference	Task	Target Crop	Model	Precision
[3]	Detection and classification of weeds	Unspecified	RF	95%
[4]	Canopy structure measurement	Avocado tree	RF	96%
[5]	Recognition of weed types	Maize	KNN, RF	81%, 76.95%
[30]	Early weed mapping	Sunflower, cotton	RF	87.90%
[31]	Weed detection by UAV	Maize	YOLOv3	98%
[24]	Water aquatic vegetation monitoring	Stratiotes aloides	RF	92.19%
[25]	Mapping of land cover	9 perennial crops	SVM	84.80%
[26]	Recognition of weed types	8 weed plants	SVM	92.35%
[27]	Detection of weeds using shape feature	Sugar beet	SVM	95%
[28]	Detection of weeds	Soybean	SVM	95.07%
[32]	Weed detection using image processing	Chilli	RF	96%
[33]	Mapping of weeds using images from UAV	Maize, Sunflower	SVM	95.50%

In [25], the authors create a land cover map of the Riverina region in New South Wales, Australia, covering a total area of 6200 km^2, to identify and categorize perennial crops in this vast region. They used object-based image analysis with supervised support vector machine classification to improve precision. After analyzing the data, they determined that the accuracy for a total item count using all twelve classes was 84.80%, but it increased to

90.20% when weighted by object area. The outcomes proved the feasibility of employing a succession of medium-resolution remote sensing images to generate comprehensive land cover maps over extensive perennial cropping regions. With an RF classifier, the authors of [3] created a real-time computer vision-based system to identify weeds in agricultural fields. The classification model was trained using the authors' dataset, and then field data were used to verify its accuracy. They also created a fluid flow control system based on pulse width modulation, which uses the information provided by the vision system to regulate the spraying of an agrochemical. As a result, the authors proved the utility of their pesticide spraying method based on real-time vision.

In [34], a support vector machine (SVM) technique is used to detect weeds in chili field images. Examining how well the SVM classifier functions within a comprehensive weed-control strategy was the focus of their study. Five distinct types of weeds were depicted in the images they took of Bangladeshi chili crops. The authors used a global thresholding-based binarization algorithm to segment the images, separating the plants from the ground to extract features. Fourteen features were extracted from each image and sorted into color, shape, and moment invariants. Eventually, a support vector machine classifier was utilized to search for weeds. Their experiments determined that the SVM was 97% accurate over a set of 224 images. The authors of [27] presented a method for weed identification in sugar beet cultivation by utilizing a combination of numerous form elements to establish patterns that would be used to distinguish between sugar beets and weeds, which are visually quite similar. Images of sugar beet farms at Shiraz University served as the basis for this study. These images were altered using the MATLAB toolbox. Shape factors, moment invariants, and Fourier descriptors were among the properties of geometric space that the authors investigated to establish a distinction between weeds and sugar beets. Next, the authors utilized KNN and SVM classifiers, whose combined accuracy was 92.92% and 95%, respectively.

A color-index-based histogram is utilized to distinguish between weed, soybean, and soil classes, and a monochrome image is produced, as described in [28]. After scaling the image to a range of 0–255, greyscale images were obtained by creating and normalizing image histograms, which were then utilized for training BPNN and SVM classifiers. This study set out to find an alternate feature vector that will guarantee a high weed identification rate while also being computationally straightforward. In total, this method yielded accuracies of 96.60% for BPNN and 95.08% for SVM. The authors presented an automated weed identification system in [31] that could identify weeds at different developmental stages. In this case, sensors mounted on an unmanned aerial vehicle (UAV) were used to acquire color, multispectral, and thermal imagery. Using color images as the ground truth, researchers manually drew bounding boxes around plant bulbs and labeled them by hand. Next, they turned the gathered images into normalized difference vegetation index (NDVI) images using image processing techniques. At last, they used machine learning techniques to sort the weeds from the useful plants.

Images were obtained from a plant laboratory in Belgium, and the authors of [5] studied how well a hyperspectral snapshot mosaic camera worked at identifying weeds and maize. The calibration formula reflectance was obtained after these raw images were processed for the band features. One hundred eighty-five features were discovered across reflectance, NDVI, and RVI in the VB and NIR spectrums. To further streamline the process, the authors turned to a principal component analysis-based feature reduction technique. This data was then fed into feature selection algorithms, which were used to isolate relevant features. In the end, an RF classifier was employed to distinguish between weeds and crops. Accuracy for identifying various weeds was up to 81% overall. At an early stage in the development of herbaceous crops, the authors of [30] proposed an automated, RF-based image processing method for weed detection. This method combines digital surface models (DSMs) with orthomosaic methods using images captured by unmanned aerial vehicles (UAVs). After that, an RF classifier was utilized to differentiate between weeds and crops/soil, with results of 87.90% for sunflower fields and 84% for cotton fields, respec-

tively. Obtaining radiometrically calibrated multispectral imaging, segmenting images, and employing a machine learning model are the essential components of a straightforward methodology presented in [24] for monitoring emerging and submerged invasive water soldiers. The eBee mapping drone was used to get the imagery. Pix4Dmapper Pro 3.0 was used to create the orthomosaic from the multispectral images.

The authors of [4] presented a method that uses UAVs to precisely predict when avocado plants are at particular stages of development. They shot the multispectral images with a Parrot Sequoia camera. After separating the digital terrain model from the digital surface model, a canopy height model was utilized to determine the height of the trees. Then, they used orthomosaic at-surface reflectance images and a variety of vegetation indices depending on the brightness of the plants in the red edge and NIR bands. The final step was implementing an RF method, which ended up being 96% accurate. UAV images from sunflower and maize fields were utilized in a weed mapping strategy proposed for precision agriculture in [33]. Object-based image analysis (OBIA) with a support vector machine (SVM) method linked with feature selection approaches was utilized to solve the spectral similarity problem for crop and weed pixels in the early growth stage. Spain's private farms, La Monclova and El-Mazorcal, had images of sunflower and corn fields captured by the UAV. After that, the images were mosaicked with the help of the Agisoft Photoscan program, and then the items in the subsample were labeled using unsupervised feature selection approaches. At the same time, the automatic labeling was done under human oversight. These items were categorized as color histograms and data features based on remote-sensing measurements (first-order statistics, textures, etc.). The results showed that this SVM-based method had an overall accuracy of about 95.50%.

3. Materials and Methods

In this section, the dataset employed in this study is presented along with the key machine learning techniques, such as baseline classification models and ensemble approaches. In addition, the basics of grey wolf and sine cosine optimization methods forming the basis of the proposed optimization algorithm are presented in this section.

3.1. Data Collection

Field crops can be captured using sensors and UAVs equipped with cameras. In this work, the wheat crop images are captured using an autonomous sprayer drone, and the dataset is freely available on Kaggle [35]. Sample images in this dataset are shown in Figure 1. The collection dataset consists of 1176 wheat images and 4851 weed images in the training set. The testing set is composed of 130 wheat images and 540 weed images.
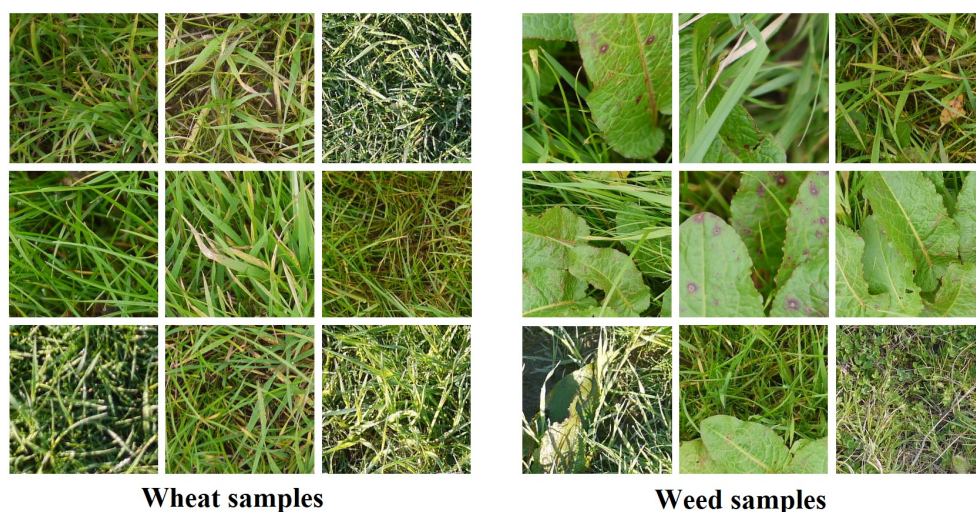

Wheat samples **Weed samples**

Figure 1. Sample wheat and weed images in the employed dataset.

3.2. Pre-Trained AlexNet

Convolutional neural networks (CNNs) are a subset of multi-layer neural networks that extract information from images by analyzing their pixels [36]. Convolution, pooling, and fully linked layers are conventional CNN's three fundamental building blocks. Convolution layers do the bulk of a CNN's computations and are the most important building blocks. In other words, it filters the input using a convolutional filter and sends the result to the following layer. The input is filtered by the applied filter, which also serves as a feature identifier, yielding a feature map. The pooling layer's job is to lower the space needed for the spatial representation and the computations that follow each successive convolution. Each sliced input is pooled in the pooling layer, lowering the computational burden of the subsequent convolution layer. Extraction and reduction of features from input images is achieved by applying convolution and pooling layers. When the fully linked layer is used, an amount of output proportional to the number of classes is produced. The layers that make up a CNN architecture are layered versions of each other. Despite some subtle differences, all CNNs are built on the same basic structure. In this work, the AlexNet pre-trained architecture is employed to extract useful features for classifying wheat and weed images.

3.3. Grey Wolf Optimizer

Grey wolf optimizer (GWO) motions are based on those of genuine wolves while they are on the prowl or hunting. Wolves tend to live in packs of varied sizes. A pack of wolves has a minimum of five members and a maximum of twelve members. There are four distinct varieties of wolves, each with a specific function within the pack. These are known as alpha, beta, omega, and delta [37]. Alpha-type wolves often make decisions on when and where to go for a stroll, hunt, and sleep with the assistance of beta-type wolves in the pack. It is generally accepted that alpha wolves are dominant wolves, with beta wolves serving as their subordinates. There are few better candidates to take over for the alpha wolf when they die. When alphas make judgments, the betas are there to support them and provide feedback to the alphas so they may make better decisions in the future; when it comes to the wolves of types alpha and beta, the delta wolves are typically subservient to the omegas. Delta wolves are divided into five groups: caretakers, hunters, elders, and scouts. In the group, each category has a distinct purpose. As the group's "scapegoats", the omega-type wolves had to submit to all the other wolves in the pack.

The grey wolf optimizer uses alpha, beta, and delta agents to lead the search for the optimum solution. In contrast, omega agents follow these three agents in the quest for the best solution. The alpha solution is considered the best-fitting solution in the grey wolf optimizer. On the other hand, the solutions of type beta and delta signify the second and third most suitable solutions.

Mathematically, the first, second, and third fittest solutions are denoted by $(\mathbf{P}_\alpha)$, $(\mathbf{P}_\beta)$, and $(\mathbf{P}_\delta)$, respectively, whereas $(\mathbf{P}_\omega)$ refers to all other solutions. The update process of the GWO algorithm is depicted in Figure 2. In this figure, the gamma wolves and other hunters are guided by the alpha, beta, and delta wolves, to efficiently manage the hunting process. The position updating is performed as follows.

$$\mathbf{P}(t+1) = \mathbf{P}_s(t) - \mathbf{A}.|\mathbf{C}.\mathbf{P}_s(t) - \mathbf{P}(t)| \tag{1}$$

where $\mathbf{P}$ is the wolf's current location and t is the number of iterations the search algorithm has gone through. The prey's location is denoted by $\mathbf{P}_s(t)$, and the coefficient vectors $\mathbf{A}$ and $\mathbf{C}$ are defined as follows.

$$\mathbf{A} = 2\mathbf{a}.\mathbf{r}_1 - \mathbf{a} \tag{2}$$

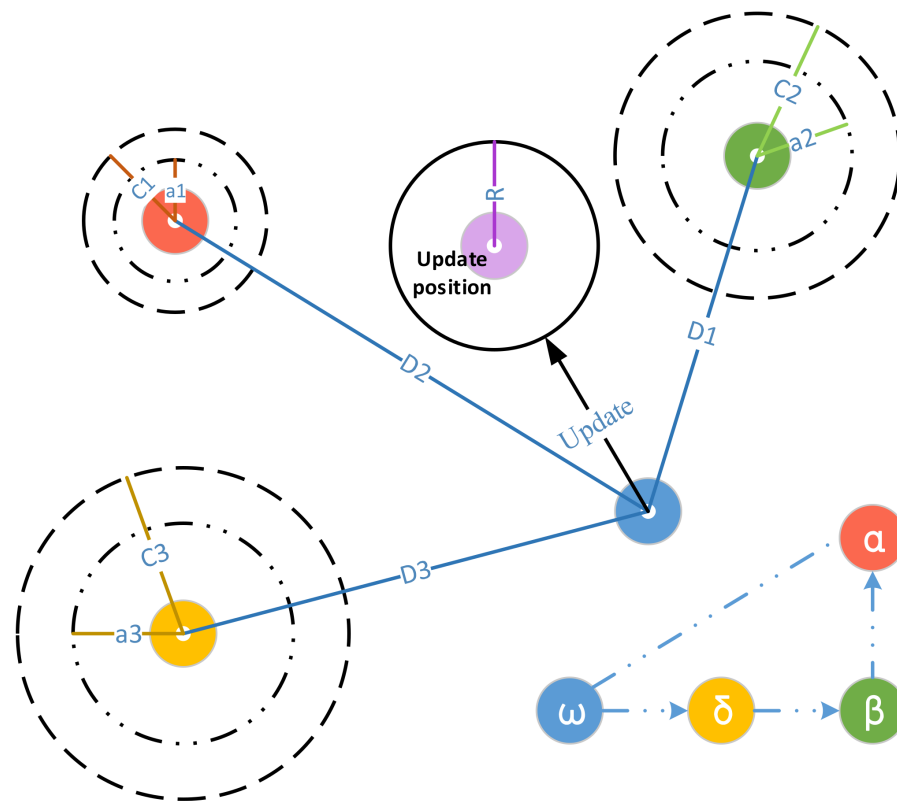$$\mathbf{C} = 2\mathbf{r}_2 \tag{3}$$

Figure 2. The updating process of the grey wolf optimization algorithm.

There are two sets of random values for the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, and the values of $\mathbf{a}$ are chosen from the range $[0,2]$ in descending order. The updated values of the vector $\mathbf{a}$ govern the balance between the exploitation and exploration operations [37]. The following formula calculates the most recent change to this vector.

$$\mathbf{a} = 2 - t.\frac{2}{M_t} \tag{4}$$

where M_t is the number of possible iterations. These positions are utilized to lead the other solutions, given by the symbol $\mathbf{P}_\omega$, to move in the direction of the prey, as seen in the search process in Figure 2. The three best-fitting solutions are $\mathbf{P}_\alpha$, $\mathbf{P}_\beta$, and $\mathbf{P}_\delta$. The process of updating the positions of the wolves is described using the following equations by substitution of $\mathbf{P}_s(t)$ in Equation (1) by $\mathbf{P}_\alpha$, $\mathbf{P}_\beta$, and $\mathbf{P}_\delta$.

$$\begin{aligned}
\mathbf{D}_\alpha &= |\mathbf{C}_1.\mathbf{P}_\alpha - \mathbf{P}(t)|, & \mathbf{P}_1 &= \mathbf{P}_\alpha - \mathbf{A}_1.\mathbf{D}_\alpha \\
\mathbf{D}_\beta &= |\mathbf{C}_2.\mathbf{P}_\beta - \mathbf{P}(t)|, & \mathbf{P}_2 &= \mathbf{P}_\beta - \mathbf{A}_2.\mathbf{D}_\beta \\
\mathbf{D}_\delta &= |\mathbf{C}_3.\mathbf{P}_\delta - \mathbf{P}(t)|, & \mathbf{P}_3 &= \mathbf{P}_\delta - \mathbf{A}_3.\mathbf{D}_\delta
\end{aligned} \tag{5}$$

The calculations of $\mathbf{A}_1$–$\mathbf{A}_3$ and $\mathbf{C}_1$–$\mathbf{C}_3$ are performed by Equations (2) and (3), respectively. The population's new position is calculated as follows.

$$\mathbf{P}(t+1) = \frac{\mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3}{3} \tag{6}$$

3.4. Sine Cosine Algorithm

In [38], the sine cosine algorithm (SCA) was presented for the first time. The sine cosine oscillation function plays a crucial role in identifying the best possible solution locations, as shown in Figure 3. A set of random variables are used to represent the steps of the SCA's operation [39,40].

- The movement location.
- The motion direction.
- Swapping between the sine and cosine components.
- Emphasizing/de-emphasizing the destination effect.

The update process of the candidate solutions is performed using the following equation.

$$\mathbf{S}(t+1) = \begin{cases} \mathbf{S}(t) + \mathbf{r}_1.sin(\mathbf{r}_2).|\mathbf{r}_3\mathbf{P}(t) - \mathbf{S}(t)| & \mathbf{r}_4 < 0.5 \\ \mathbf{S}(t) + \mathbf{r}_1.cos(\mathbf{r}_2).|\mathbf{r}_3\mathbf{P}(t) - \mathbf{S}(t)| & \mathbf{r}_4 \geq 0.5 \end{cases} \qquad (7)$$

where t is the iteration number, the positions of the current and previous solutions at iterations $t+1$ and t are denoted by $\mathbf{S}(t+1)$, and $\mathbf{S}(t)$, respectively. The position of the best solution is referred to as $\mathbf{P}$. The values of $[0-1]$ are allocated to the random variables $\mathbf{r}_2, \mathbf{r}_3$, and $\mathbf{r}_4$. The equation shows, for instance, that the positions of the optimal solutions affect the location of the current solution, making it simpler to reach the optimal solution. The following equation expresses the dynamic change in the value of $\mathbf{r}_1$.

$$\mathbf{r}_1 = a - \frac{a \times t}{t_{max}} \qquad (8)$$

where a is a constant, t and t_{max} represent the current and maximum iterations, respectively. Due to its reliance on a single optimal solution to guide the other solutions, the SCA algorithm is more robust than many other meta-heuristic algorithms presented in the literature [39,40]. The convergence speed and memory usage of this approach are relatively low when compared to other algorithms. However, as the number of locally optimal solutions increases, the algorithm's performance degrades. To avoid being stuck in a local optimum, the proposed new algorithm incorporates the SCA optimizer and the GWO algorithm, taking advantage of their rapid convergence rates and memory efficiency and ensuring a balanced set of exploration and exploitation activities.
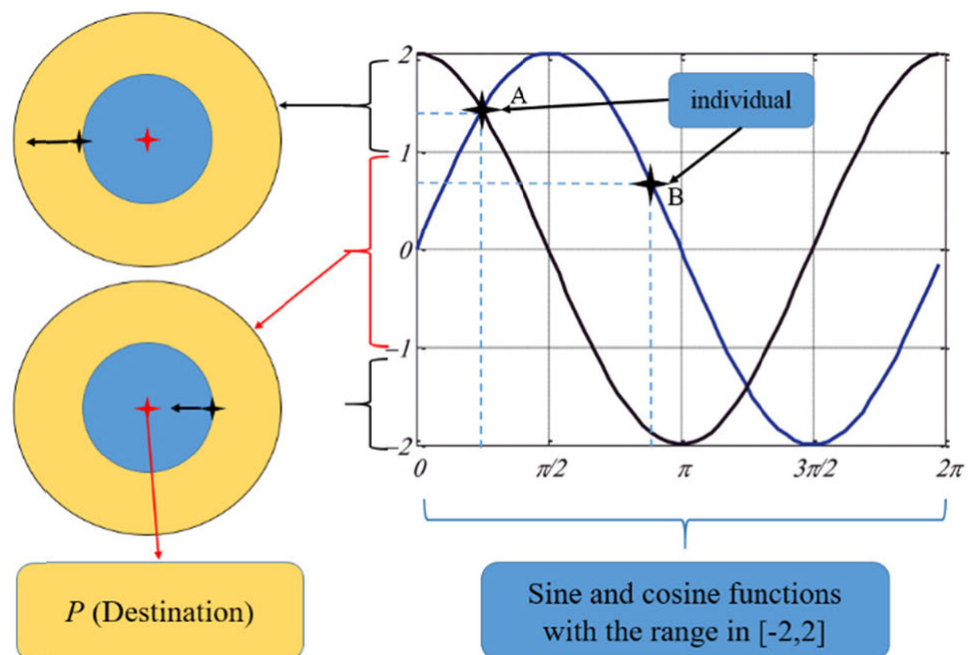

Figure 3. Tolerance given to the solutions that proceed in either direction toward or away from the destination for sine and cosine functions [41].

3.5. Baseline Machine Learning Models

This paper employs three baseline machine learning models to form the proposed ensemble voting approach. These base models are neural networks, k-nearest neighbors, and support vector machines. In this section, the basics of these models are presented briefly. On the other hand, ensemble methods cover two main types of ensemble methods: bagging classifier with a random forest as a type of averaging technique and Adaboost with a voting ensemble as a type of boosting technique. An introduction to these types of ensemble models is also presented in this section.

3.5.1. Neural Networks (NN)

Two or more layers of neurons and connections in the neural network structure allow it to learn a non-linear decision boundary. The term "processing elements" refers to what these neurons are often known as (PEs). The PEs use special training algorithms (such as ADAM and SGD) to try to mimic the operation of the human nervous system [42]. Input and output data can be separated by a "hidden layer", a layer between the two that is not visible to the user. The calculation of the node output value weighted total is as follows:

$$S_j = \sum_{i=1}^{n} w_{ij} I_i + \beta_j \tag{9}$$

in which I_i is the input variable i, and w_{ij} represents the hidden layer connection weight between I_i and neuron j its β_j, the bias. The sigmoid activation function may be used to define the node j output as follows:

$$f_j(S_j) = \frac{1}{1 + exp^{-S_j}} \tag{10}$$

The value of $f_j(S_j)$ is used to define the network output using the previously hidden layer neurons as:

$$y_k = \sum_{j=1}^{m} w_{jk} f_j(S_j) + \beta_k \tag{11}$$

The weights between neurons in the hidden layer and the output node are represented by w_{jk} and β_k, respectively.

3.5.2. K-Nearest Neighbors (KNN)

The K-nearest-neighbors (KNN) method is a non-parametric supervised classification technique that is both straightforward and useful in many contexts. Among classifiers used for pattern recognition, the KNN classifier is recommended due to its simple implementation, high accuracy, and speed of results [25]. It has various applications, including pattern recognition, machine learning, text classification, data mining, and object recognition. The KNN method employs a technique known as "classification by analogy", wherein an unknown data item is compared to its neighbors in the training set. The Euclidean distance is the standard for comparing two samples' degrees of similarity. Attributes with broader ranges are not given more weight than attributes with lower ranges by normalizing their attribute values. Using KNN, the most common category is used to categorize an unknown pattern. The following Euclidean distance equation is used to measure the distance between known/unknown data to determine the best category of the unknown label.

To classify dataset samples using the KNN approach, the nearest samples are considered to determine the final decision [43]. This approach depends mainly on the value of *K*, which represents the number of neighbors considered in classifying the dataset samples in terms of the following Euclidean distance.

$$D(x_{train}, x_{test}) = \sqrt{\sum_{i=1}^{k}(x_{train} - x_{test})^2} \tag{12}$$

This technique is used in conjunction with the NN in the proposed voting ensemble classifier for boosting the classification accuracy of wheat and weed images.

3.5.3. Support Vector Machine (SVM)

Support vector machine (SVM) is one of the effective machine learning models that can achieve promising performance when combined with deep networks and other machine learning models [44]. The basic formula of SVM is presented in the following.

$$f(a) = w.a + d \tag{13}$$

where a is the input variable, w is the weight vector, and d is the model's error value. The discrepancy between anticipated and actual values can be reduced using SVM. According to the error indicator, SVM predicts the output label using an error reduction strategy based on the following optimization model.

$$Minimize : \frac{1}{2}||w||^2 + c\sum_{i=1}^{k}(c_i^- - c_i^+) \tag{14}$$

$$Subject(to) : (wx_i + d) - b_i < \epsilon + c_i^+, \tag{15}$$

$$b_i - (w_i a_i + d) \leq (\epsilon + \epsilon^-) \tag{16}$$

There are data violations whose varied values are larger than ϵ, the acceptable range with observable values, which are denoted by the coefficient of punishment C, the weight of the variables, the input variable, and the target observation (w_i), a_i and b_i. The values of the variables in Equations (15) and (16) are estimated to be used in Equation (14). It is possible to use a kernel function in SVM to describe the high-dimensional feature space of the input data points. The kernels are equipped to deal with a wide range of problems. Sigmoid, linear, polynomial, and radial basis functions (RBF) are among the four well-known SVM kernels employed. Because the RBF kernel has been shown to be capable of generalizing well to varied datasets, it was utilized in this investigation. As a result, Equation (13) may be interpreted as follows:

$$f(a) = w.H(a, a_i) + d \tag{17}$$

$$H(a, a_i) = exp\left(-\frac{a - a_i}{2\gamma^2}\right) \tag{18}$$

where $H(a, a_i)$ is the kernel function and γ is the parameter of this function. Unknown values of SVM parameters such as C and ϵ are used as decision variables. The optimization procedure must thus include them.

3.5.4. Ensemble Models

Because of the reduced variance, the ensemble approaches aim to combine the outputs of machine learning (ML) classifiers. Several classifiers are built independently, and their results are then averaged (e.g., bagging classifier, random forest, and voting techniques) (soft and hard). Overfitting is less of an issue with these methods. One of the most often used and effective ensemble techniques for classification and regression is random choice forests (RF). Compared to single classifiers, ensemble models have attracted much attention because of their accuracy, and noise tolerance [45]. The average ensemble classifier results from the output predictions using the following formula.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \tag{19}$$

Using a set of weak classifiers, the boosting approach of ensemble models creates a robust classifier. If you want to forecast unexpected observations accurately, this technique uses a set of classifier weights and a training data case for each iteration. Any machine learning approach that takes weights on the training set can be used as a base estimator. Various classifiers are trained on randomly generated training sets to generate the final result. The test sample is classified by combining the outputs of all models using uniform averaging or voting procedures over class labels by all classifiers in the ensemble. Because of the randomization in its structural approach, this methodology may be utilized to reduce variance and subsequently be used to form an ensemble. Machine learning estimators are combined, and a majority vote (i.e., the output of each estimator) is termed hard voting in this approach. The class label is returned as the argmax of the sum of the predicted probabilities via soft voting or average predicted probabilities on the other hand [46]. For each classifier, the anticipated class probabilities are gathered. An average of the weights assigned to each classifier is then calculated. The class label determines the final class label (i.e., the highest average probability). ML classifiers of equivalent performance can use this strategy to counteract one other's flaws.

4. The Proposed Methodology

In this section, the proposed methodology is explained. The methodology starts with extracting the features of the input images using the deep network through transfer learning. The extracted features are then processed to select the most relevant features that boost the classification accuracy. The selected features are then used to learn three base models: NN, KNN, and SVM. These models are employed in a voting classifier optimized using the proposed optimization algorithm. The steps of the proposed methodology are depicted in Figure 4. The next section discusses the main steps of the proposed methodology.
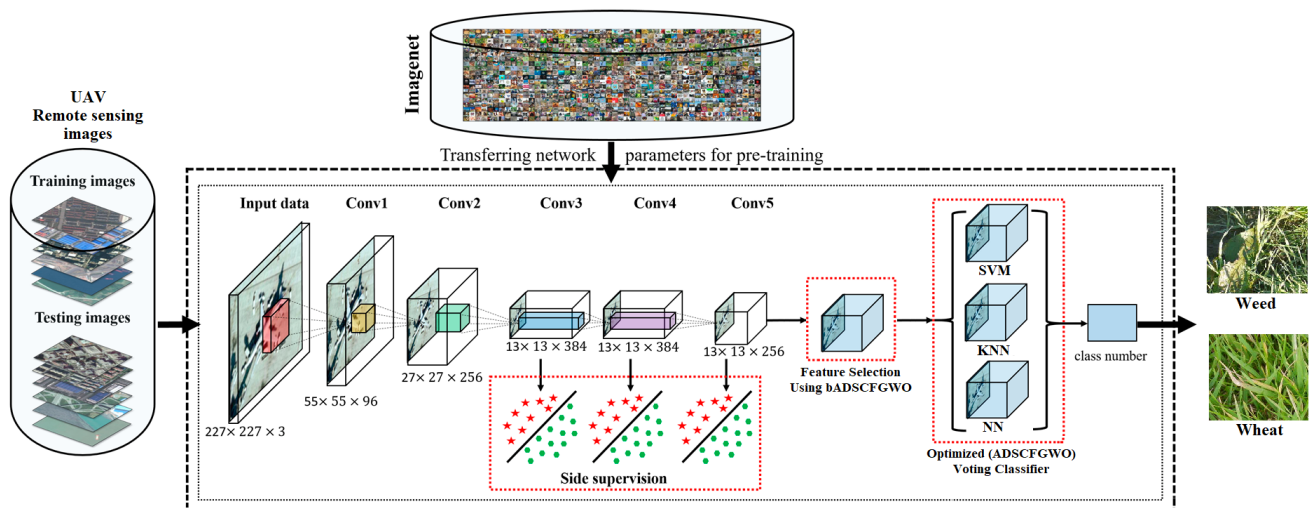
Figure 4. The proposed methodology for weed/wheat classification.

4.1. Transfer Learning

In deep learning applications, the process of transfer learning is widely used [47], which is beneficial in the case of a limited dataset. To learn a new classification task, a pre-trained network is considered, such as AlexNet. In this work, we adopted AlexNet, which is trained using a large dataset, ImageNet. In the transfer learning process, the three connected layers of the AlexNet are replaced with the proposed voting classifier. The transfer learning process employed in this work is depicted in Figure 5.
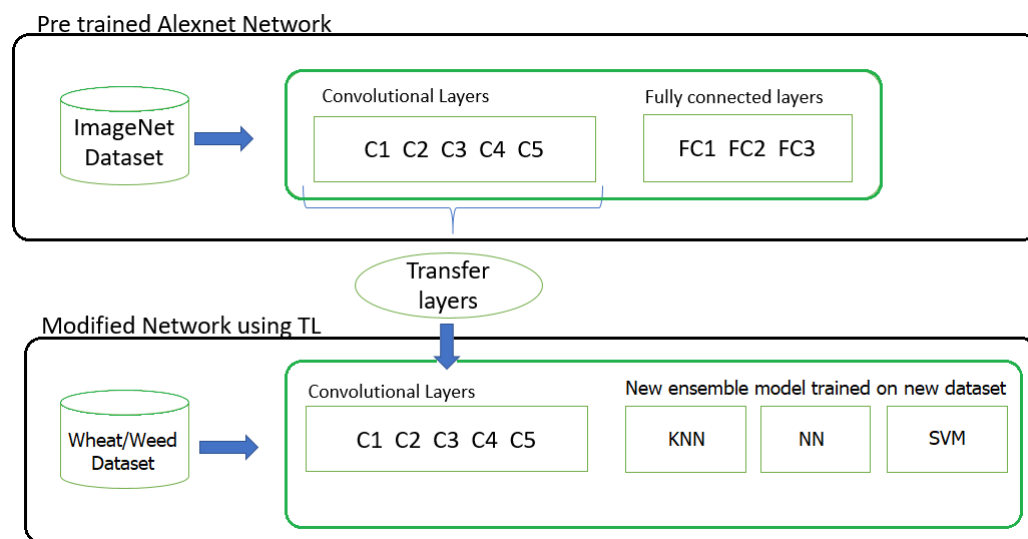
Figure 5. The transfer learning process.

4.2. Feature Extraction

Processing raw data to extract additional variables that aid machine learning algorithms is the focus of the feature extraction process. This work adopts AlexNet [48] for feature extraction. Figure 4 shows how AlexNet enlarges the input image to a fixed size of $227 \times 227 \times 3$, using a 256-layer convolution filter with a window shape of 11×11, a 256-layer filter with a window shape of 5×5, and then 384, 384, and 256-layer convolution filters with a window size of 3×3 for the remaining three layers of the process. After the first, second, and final convolutional layers, the network has a maximum number of 3×3 pooling layers with a stride of 2. There are two fully connected layers with 4096 neuron outputs following the fifth convolutional layer in addition to these five layers. Afterward, there is a single completely linked output layer at the end of the network, which originally had 1000 output classes. In the end, Dropout, ReLU, and preprocessing are critical if you want top results in computer vision applications. This work replaces the last three layers with the proposed optimized voting ensemble model, which classifies only two classes (wheat and weed).

4.3. The Proposed Optimization Algorithm

The optimization of the proposed voting ensemble classifier is performed in terms of a new optimization algorithm based on the SCA and GWO optimization algorithms. The proposed optimization algorithm is referred to as the adaptive dynamic sine cosine fitness grey wolf optimization (ADSCFGWO) algorithm, with the steps listed in Algorithm 1.

Algorithm 1: The proposed ADSCFGWO algorithm.

1: **Initialize** population $\mathbf{P}_i (i = 1, 2, ..., n)$ with size n, maximum iterations $Iter_{Max}$, fitness function $\mathbf{H}_n$, $\mathbf{a}$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{r}_4$
2: **procedure** DYNAMICSEARCH($\mathbf{H}_n$)
3:    **if** (Fitness $\mathbf{H}_n$ did not change for three iterations) **then**
4:       **Increase** the exploration group solutions
5:       **Decrease** the exploitation group solutions
6:    **end if**
7: **end procedure**
8: **Calculate** $\mathbf{H}_n$ fitness for each $\mathbf{P}_i$
9: **Find** the first three best solutions denoted by $\mathbf{P}_\alpha, \mathbf{P}_\beta, \mathbf{P}_\delta$
10: **Set** t = 1
11: **while** $t \leq Iter_{Max}$ **do**
12:    **Update** $\mathbf{r}_1$ by $\mathbf{r}_1 = \mathbf{a}\left(1 - \frac{t}{Iter_{Max}}\right)$
13:    **for** $(i = 1 : i < n_1 + 1)$ **do**
14:       **DynamicSearch($\mathbf{H}_n$)**
15:       **Update** $\mathbf{H}_\alpha = \frac{\mathbf{H}_\alpha}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
16:       **Update** $\mathbf{H}_\beta = \frac{\mathbf{H}_\beta}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
17:       **Update** $\mathbf{H}_\delta = \frac{\mathbf{H}_\delta}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
18:       **Calculate** $\mathbf{M} = |\mathbf{C}_1 . (\mathbf{H}_\alpha * \mathbf{P}_\alpha + \mathbf{H}_\beta * \mathbf{P}_\beta + \mathbf{H}_\delta * \mathbf{P}_\delta) - \mathbf{P}(t)|$
19:       **Calculate** $\mathbf{V}_1 = \mathbf{P}_\alpha - \mathbf{A}_1 . \mathbf{M}$
20:       **Calculate** $\mathbf{V}_2 = \mathbf{P}_\beta - \mathbf{A}_2 . \mathbf{M}$
21:       **Calculate** $\mathbf{V}_3 = \mathbf{P}_\delta - \mathbf{A}_3 . \mathbf{M}$
22:       **Update** positions as $\mathbf{P}(t+1) = \frac{\mathbf{V}_1 + \mathbf{V}_2 + \mathbf{V}_3}{3}$
23:       **if** $(\mathbf{r}_4 < 0.5)$ **then**
24:          $\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{r}_1 \times \sin(\mathbf{r}_2) \times |\mathbf{r}_3 \mathbf{P}_\alpha - \mathbf{P}(t)|$
25:       **end if**
26:    **end for**
27:    **for** $(i = 1 : i < n_2 + 1)$ **do**
28:       **DynamicSearch($\mathbf{H}_n$)**
29:       **Update** $\mathbf{H}_\alpha = \frac{\mathbf{H}_\alpha}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
30:       **Update** $\mathbf{H}_\beta = \frac{\mathbf{H}_\beta}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
31:       **Update** $\mathbf{H}_\delta = \frac{\mathbf{H}_\delta}{\mathbf{H}_\alpha + \mathbf{H}_\beta + \mathbf{H}_\delta}$
32:       **Calculate** $\mathbf{M} = |\mathbf{C}_2 . (\mathbf{H}_\alpha * \mathbf{P}_\alpha + \mathbf{H}_\beta * \mathbf{P}_\beta + \mathbf{H}_\delta * \mathbf{P}_\delta) - \mathbf{P}(t)|$
33:       **Calculate** $\mathbf{V}_1 = \mathbf{P}_\alpha - \mathbf{A}_1 . \mathbf{M}$
34:       **Calculate** $\mathbf{V}_2 = \mathbf{P}_\beta - \mathbf{A}_2 . \mathbf{M}$
35:       **Calculate** $\mathbf{V}_3 = \mathbf{P}_\delta - \mathbf{A}_3 . \mathbf{M}$
36:       **Update** positions as $\mathbf{P}(t+1) = \frac{\mathbf{V}_1 + \mathbf{V}_2 + \mathbf{V}_3}{3}$
37:       **if** $(\mathbf{r}_4 \geq 0.5)$ **then**
38:          $\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{r}_1 \times \cos(\mathbf{r}_2) \times |\mathbf{r}_3 \mathbf{P}_\alpha - \mathbf{P}(t)|$
39:       **end if**
40:    **end for**
41:    **Update** $\mathbf{H}_n$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{r}_4$, $\mathbf{P}_\alpha$, $\mathbf{P}_\beta$, $\mathbf{P}_\delta$, t
42:    **Find** best individual $\mathbf{P}^*$
43: **end while**
44: **Return** $\mathbf{P}^*$

4.3.1. Exploration/Exploitation Balance

The proposed ADSCFGWO algorithm automatically strikes the right balance between exploration and exploitation by dividing the population into smaller groups. Exploration and exploitation groups make up 70% of the population in this algorithm, which divides

the population into two categories. An early exploration group with a high number of participants helps to identify new and intriguing search regions. Overall fitness grows, but the number of exploration group members drops from 70% to 30% due to more exploitative people gaining fitness values. If a better solution cannot be identified, an elitist technique maintains convergence by keeping the process leader in place in subsequent populations. ADSCFGWO can expand the size of the exploration group at any point if the fitness of the group's leader has not increased sufficiently throughout three iterations.

4.3.2. Fitness Function

The following equation is used to assess the quality of the solutions discovered by the optimization algorithms.

$$H_n = \alpha Error(P) + \beta \frac{|S|}{|A|} \tag{20}$$

where P stands for the model's variables. It can be noted that a certain trait is important to the population by looking for the values of $\alpha \in [0 \text{ to } 1]$ and $\beta = 1 - \alpha$. The total number of features is denoted by $|S|$.

4.4. Feature Selection

Features that meet particular criteria, such as originality, consistency, and meaningfulness, are selected and identified throughout the feature selection process. Two binary values (0 and 1) are utilized in the feature selection procedure to limit the search space. Therefore, continuous values-based optimizers require an update to deal with this problem effectively. This is the essential phase in feature engineering since it allows optimizers to choose the most optimal features for maximum performance. There are several ways to think about selecting features, such as a binary vector, in which each feature has an equal chance of being included in the solution or not [49]. Random populations of vectors with random features can be utilized as a starting point for meta-heuristic algorithms. This is followed by an iterative process of exploring and exploiting to identify the best collection of features [50]. To determine whether a feature is relevant, the search space is confined to binary values (0 and 1) alone. The proposed binary ADSCFGWO (bADSCFGWO) method is proposed to transform the continuous values from the continuous ADSCFGWO algorithm into binary {0, 1} values to fit the feature selection procedure. The *Sigmoid* function used to convert the continuous solution to binary values is represented by the following equation, and the steps of the proposed binary optimization algorithm are listed in Algorithm 2.

Algorithm 2: The proposed bADSCFGWO feature selection algorithm.

1: **Initialize** the population, and configuration parameters
2: **Convert** the solution to binary {0, 1}.
3: **Select** the best solutions using the objective function
4: **while** $t \leq Iter_{Max}$ **do**
5: 　**Run** the proposed ADSCFGWO algorithm
6: 　**Convert** solutions to binary using Equation (21)
7: 　**Measure** fitness function
8: 　**Update** the algorithm parameters
9: **end while**
10: **Return** P^*

$$\mathbf{B}^{(t+1)} = \begin{cases} 1 & \text{if } Sigmoid(m) \geq 0.5 \\ 0 & otherwise \end{cases},$$

$$Sigmoid(m) = \frac{1}{1 + e^{-10(m-0.5)}} \tag{21}$$

where at iteration t, m refers to the best answer.

5. Experimental Results

To evaluate the proposed approach of weed detection, a set of experiments were conducted to assess the performance of the process of the proposed approach. The assessment included the steps of feature selection, classification methods, and the proposed optimized voting classifiers approach. The coming sections present the details of the achievements.

5.1. Configuration Parameters

The first set of experiments was conducted to determine the best collection of values assigned to the configuration parameters. Table 2 presents the set of values of the parameters of the optimization of the feature selection process. In addition, the configuration parameters of the grey wolf and other optimization algorithms are presented in Table 3. The values of these parameters are employed in the proposed optimization algorithm and the algorithms used in the comparison experiments.

Table 2. Configuration parameters of the proposed feature selection method.

Parameter	Value
Iterations	100
Agents	10
α of Equation (20)	0.99
β of Equation (20)	0.01
θ	$[0, 12\pi]$
a	$[-10, 10]$
b	$[-10, 10]$

Table 3. Configuration parameters of the optimization algorithms.

Algorithm	Parameter	Value
GA	Cross over	0.9
	Agents	10
	Iterations	80
	Selection mechanism	Roulette wheel
	Mutation ratio	0.1
PSO	Acceleration constants	[2,2]
	Iterations	80
	Inertia W_{min}, W_{max}	[0.6, 0.9]
	Particles	10
WOA	r	[0, 1]
	Whales	10
	Iterations	80
	a	2 to 0
GWO	a	2 to 0
	Iterations	80
	Wolves	10
FA	Iterations	80
	Fireflies	10

5.2. Evaluation Metrics

The metrics used to assess the feature selection approach are presented in Table 4. These criteria include the average fitness size, average error, standard deviation, best fitness, worst fitness, and mean error. In their criteria, M refers to the number of runs of the optimizer, j refers to the run number, g_j^* is the best solution at run number j, and the size of the best solution vector is referred to as $size(g_j^*)$, the number of points in the test set is denoted by N. The output class label is C_i for the data point i corresponding to the label L_i. D denotes the total number of features.

On the other hand, the metrics used to assess the proposed voting classifier are presented in Table 5. These metrics include F1-score, specificity, accuracy, sensitivity, Nvalue, and Pvalue. The true positive, true negative, false positive, and false negative measures used in these metrics are denoted by TP, TN, FP, and FN, respectively.

Table 4. Evaluation metrics used in assessing the feature selection approach.

Metric	Formula
Average fitness size	$\frac{1}{M} \sum_{i=1}^{M} size(g_*^i)$
Average error	$\frac{1}{M} \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} mse(C_i, L_i)$
Standard deviation	$\sqrt{\frac{1}{M-1} \sum_{i=1}^{M} \left(g_*^i - Mean\right)^2}$
Best fitness	$min_{i=1}^{M} g_*^i$
Worst fitness	$max_{i=1}^{M} g_*^i$
Mean	$\frac{1}{M} \sum_{i=1}^{M} g_*^i$

Table 5. Evaluation metrics used in assessing the proposed optimized voting classifier.

Metric	Formula
F1-score	$\frac{TP}{TP+0.5(FP+FN)}$
Specificity (TNR)	$\frac{TN}{TN+FP}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity (TPR)	$\frac{TP}{TP+FN}$
Nvalue (NPV)	$\frac{TN}{TN+FN}$
Pvalue (PPV)	$\frac{TP}{TP+FP}$

5.3. Feature Extraction Results

Deep learning is one of the main approaches to extracting effective features from images. In this work, to select the deep network that selects the best features from the given images, an experiment is conducted to study the effectiveness of the features extracted from three deep networks, namely, VGGNet, ResNet-50, and AlexNet. The results are recorded and presented in Table 6. As presented in this table, the best results are achieved using AlexNet, and thus this network is adopted for the rest of the conducted experiments.

Table 6. Evaluation of three deep networks for feature extraction.

	VGGNet	ResNet-50	AlexNet
Accuracy	0.769	0.833	0.847
Specificity (TNR)	0.800	0.800	0.783
Sensitivity (TPR)	0.714	0.862	0.889
Nvalue (NPV)	0.833	0.833	0.818
Pvalue (PPV)	0.667	0.833	0.865
F-score	0.690	0.847	0.877

5.4. Evaluating the Proposed Feature Selection Method

The feature selection applied to the weed features is performed using the proposed binary ADSCFGWO. The achieved results are compared with other state-of-the-art binary optimization techniques; namely, grey wolf optimizer (GWO) [51], hybrid GWO and PSO

(bGWO_PSO) [52], particle swarm optimization (PSO) [53], whale optimization algorithm (WOA) [54], firefly algorithm (FA) [55], and genetic algorithm (GA) [56]. The evaluation results are presented in Table 7. As presented in the table, the proposed binary ADSCFGWO algorithm achieved the best average error (0.69504) compared to the other optimization algorithms. In addition, average select size, mean fitness, worst fitness, and standard deviation are superior for the proposed algorithm compared to the other algorithms.

Table 7. Evaluation of the proposed feature selection method and three other competing methods.

	bADSCFGWO	**bGWO**	**bGWO_PSO**	**bPSO**	**bWAO**	**bFA**	**bGA**
Average select size	0.647	0.847	0.981	0.847	1.011	0.882	0.790
Std fitness	0.580	0.585	0.603	0.584	0.586	0.621	0.586
Average fitness	0.758	0.774	0.782	0.772	0.780	0.824	0.785
Average error	0.695	0.712	0.751	0.746	0.745	0.744	0.725
Worst fitness	0.758	0.761	0.846	0.820	0.820	0.841	0.804
Best fitness	0.660	0.694	0.736	0.753	0.744	0.743	0.689

5.5. Evaluating The Proposed Optimized Voting Classifier

Three baseline models have been experimented with and evaluated separately. These models are KNN, SVM, and NN. The recorded results are presented in Table 8. In this table, the accuracy achieved by KNN, SVM, and NN is 89.1%, 92.1%, and 93.5%, respectively. From these results, it can be noted that the NN baseline model achieves the best performance.

Table 8. Evaluating the results achieved by three baseline machine learning models.

	KNN	**SVM**	**NN**
Accuracy	0.891	0.921	0.935
Specificity (TNR)	0.743	0.857	0.870
Sensitivity (TPR)	0.952	0.952	0.971
Nvalue (NPV)	0.867	0.900	0.943
Pvalue (PPV)	0.899	0.930	0.930
F-score	0.925	0.941	0.950

The proposed ADSCFGWO algorithm is used to optimize the parameters of a voting ensemble model composed of the three baseline models. To prove the proposed approach's superiority, the results are compared to those of four other optimization algorithms, namely, WOA, GWO, GA, and PSO. Table 9 presents the recorded results. In this table, it can be noted that the results achieved by the proposed optimized voting ensemble are better than those achieved by optimizing the voting ensemble using other optimization methods.

On the other hand, a set of experiments is conducted to analyze the proposed approach's performance statistically. Table 10 presents the statistical analysis results. This table shows that based on 20 random samples, the mean accuracy is 97.74% and the standard deviation is relatively tiny (0.0004894), indicating the proposed approach's robustness. When these results are compared to those of the other optimization algorithms, the superiority of the proposed approach is obvious.

Table 9. Evaluation of the results achieved by optimizing the proposed voting classifier using the proposed optimization method and three other optimizers.

	Voting (ADSCFGWO)	Voting (WOA)	Voting (GWO)	Voting (GA)	Voting (PSO)
Accuracy	0.977	0.943	0.945	0.951	0.954
Specificity (TNR)	0.952	0.870	0.870	0.870	0.870
Sensitivity (TPR)	0.984	0.977	0.977	0.981	0.983
Nvalue (NPV)	0.943	0.943	0.943	0.943	0.943
Pvalue (PPV)	0.987	0.943	0.945	0.954	0.958
F-score	0.986	0.960	0.961	0.967	0.970

Table 10. Statistical analysis of the achieved classification results using the proposed optimized voting ensemble model.

	ADSCFGWO	WOA	GWO	GA	PSO
Number of values	20	20	20	20	20
25% Percentile	0.9774	0.943	0.945	0.951	0.954
75% Percentile	0.9774	0.943	0.945	0.951	0.954
Std. error of mean	0.0001	0.001	0.001	0.001	0.001
Std. deviation	0.0004	0.003	0.003	0.004	0.003
Mean	0.9775	0.9439	0.945	0.950	0.954
Minimum	0.9774	0.934	0.935	0.938	0.944
Maximum	0.9794	0.951	0.956	0.955	0.964
Median	0.9774	0.943	0.945	0.951	0.954
Range	0.002	0.017	0.021	0.0172	0.020
Upper 95% CI of mean	0.9777	0.945	0.947	0.952	0.956
Lower 95% CI of mean	0.9773	0.942	0.944	0.949	0.953
Coefficient of variation	0.0501%	0.353%	0.366%	0.397%	0.340%
Geometric SD factor	1.001	1.004	1.004	1.004	1.003
Geometric mean	0.9775	0.944	0.945	0.950	0.954
Upper 95% CI of harm. mean	0.9777	0.945	0.947	0.952	0.956
Lower 95% CI of harm. mean	0.9773	0.942	0.944	0.949	0.953
Sum	19.55	18.88	18.90	19.01	19.09

The significance and stability of the proposed approach are studied in terms of the analysis of variance (ANOVA) and Wilcoxon signed-rank tests. The results are shown in Tables 11 and 12. The measured p-value of the ANOVA and Wilcoxon tests is ($p < 0.0001$), which indicates the significance of the proposed approach.

Table 11. ANOVA test of the achieved classification results using the optimized voting ensemble model.

	SS	DF	MS	F (DFn, DFd)	p Value
Treatment	0.01496	4	0.003739	F (4, 95) = 389.0	$p < 0.0001$
Residual	0.0009132	95	0.000009612		
Total	0.01587	99			

Table 12. Wilcoxon signed-rank test of the results achieved by the proposed optimized voting classifier.

	ADSCFGWO	WOA	GWO	GA	PSO
Theoretical median	0	0	0	0	0
Actual median	0.9774	0.9434	0.9449	0.9513	0.9544
Number of values	20	20	20	20	20
Wilcoxon signed-rank test					
p value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Sum of positive ranks	210	210	210	210	210
Sum of signed ranks (W)	210	210	210	210	210
Sum of negative ranks	0	0	0	0	0
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
p value summary	****	****	****	****	****
Discrepancy	0.9774	0.9434	0.9449	0.9513	0.9544
Significant (alpha = 0.05)?	Yes	Yes	Yes	Yes	Yes

A visual representation of the results achieved by the proposed approach is shown in Figure 6. This figure shows the residual plot with residual error in the range of (−0.015 to 0.010). The homoscedasticity and QQ plots show a robust prediction of the class labels. On the other hand, the heatmap indicates a promising performance using the proposed ADSCFGWO algorithm, which is better than the other optimization algorithms.
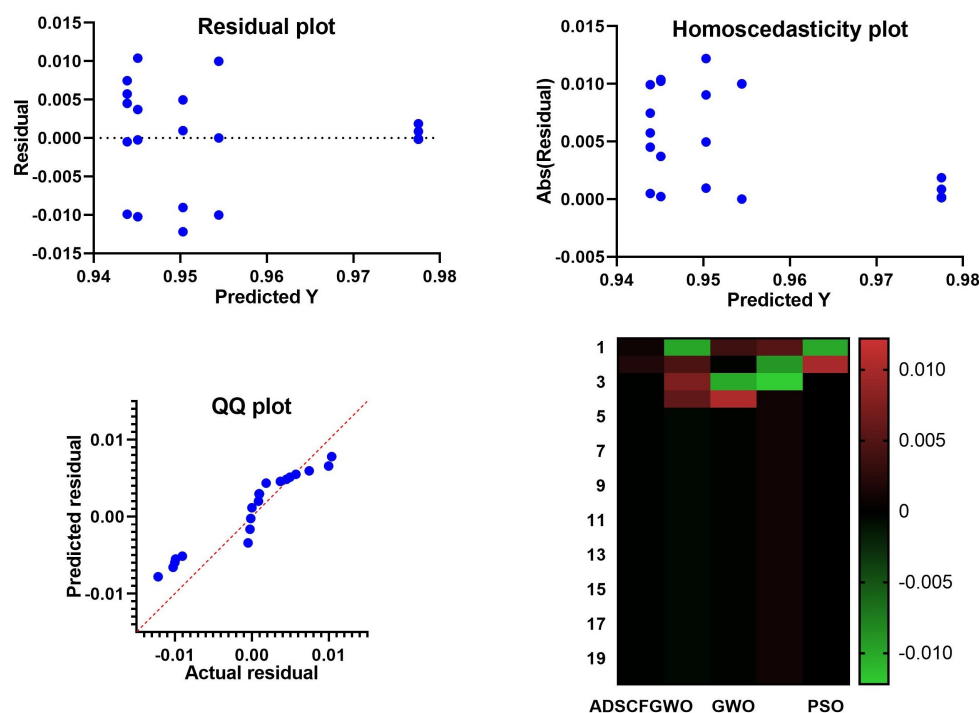
Figure 6. Residual, homoscedasticity, and QQ plots and heatmap of the ADSCFGWO and compared algorithms.

The receiver operating characteristic (ROC) plot depicted in Figure 7 shows robust detection results. In addition, this figure's accuracy and histogram plots show a promising performance that outperforms the results of the other optimization algorithms.
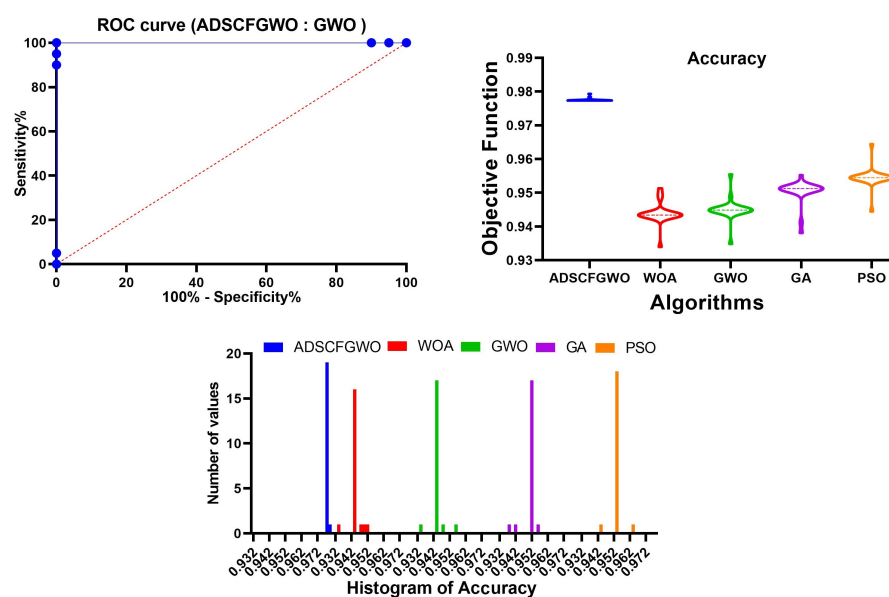
Figure 7. ROC, accuracy, and histogram plots of the ADSCFGWO and compared algorithms.

5.6. Sensitivity Analysis of the Proposed Approach

One-at-a-time (OAT) sensitivity analysis was used to study the sensitivity analysis. Regarding sensitivity analysis, OAT is one of the most straightforward methods. To test the algorithm's performance, one parameter at a time is changed while the other parameters remain the same. As the values of various factors were varied, the convergence time and ADSCFGWO's fitness values changed accordingly (as presented in Tables 13–16). When evaluating each parameter, 20 values are selected in that parameter's interval by multiplying the interval's length by 5%. As a result, the algorithm ran ten times for each number; the results are shown in the table below. It took 100 runs of ADSCFGWO for each parameter.

Table 13. Convergence time results (in seconds) for different values of ADSCFGWO's parameters (1).

r_1		r_2		r_3		r_4	
Values	**Time**	**Values**	**Time**	**Values**	**Time**	**Values**	**Time**
0.05	12.21	0.05	11.99	0.05	11.82	0.05	11.89
0.10	12.21	0.10	11.64	0.10	12.21	0.10	11.63
0.15	12.36	0.15	12.15	0.15	12.01	0.15	11.61
0.20	11.82	0.20	12.20	0.20	11.98	0.20	12.32
0.25	11.77	0.25	12.04	0.25	12.12	0.25	12.02
0.30	11.98	0.30	11.73	0.30	11.59	0.30	12.15
0.35	12.31	0.35	11.59	0.35	12.40	0.35	12.00
0.40	12.00	0.40	12.35	0.40	11.63	0.40	11.78
0.45	12.44	0.45	11.72	0.45	12.06	0.45	12.44
0.50	12.07	0.50	12.31	0.50	11.61	0.50	11.73
0.55	11.64	0.55	11.88	0.55	12.35	0.55	12.45
0.60	11.65	0.60	11.81	0.60	12.06	0.60	12.05
0.65	12.20	0.65	11.80	0.65	11.62	0.65	12.04
0.70	12.14	0.70	11.77	0.70	11.76	0.70	12.44
0.75	11.61	0.75	11.76	0.75	11.58	0.75	12.24
0.80	12.20	0.80	11.68	0.80	12.15	0.80	12.26
0.85	11.76	0.85	11.67	0.85	11.62	0.85	11.97
0.90	11.83	0.90	11.91	0.90	11.68	0.90	12.16
0.95	11.78	0.95	12.36	0.95	11.94	0.95	11.80
1.00	11.95	1.00	12.25	1.00	11.79	1.00	12.15

Table 14. Convergence time results (in seconds) for different values of ADSCFGWO's parameters (2).

A_1		A_2		A_3		C_1		C_2	
Values	**Time**	**Values**	**Time**	**Values**	**Time**	**Values**	**Time**	**Values**	**Time**
0.05	11.63	0.10	11.78	0.10	11.60	0.10	11.68	0.10	11.70
0.10	12.40	0.20	11.73	0.20	11.88	0.20	12.06	0.20	12.21
0.15	12.39	0.30	11.58	0.30	12.41	0.30	11.62	0.30	11.75
0.20	11.83	0.40	11.63	0.40	12.30	0.40	12.26	0.40	12.37
0.25	11.90	0.50	11.77	0.50	11.62	0.50	11.87	0.50	12.38
0.30	12.02	0.60	12.24	0.60	12.11	0.60	12.27	0.60	11.98
0.35	11.88	0.70	12.36	0.70	11.67	0.70	12.17	0.70	11.65
0.40	12.12	0.80	11.61	0.80	12.21	0.80	12.11	0.80	12.32
0.45	12.38	0.90	12.32	0.90	11.88	0.90	12.24	0.90	11.85
0.50	12.08	1.00	11.67	1.00	11.93	1.00	11.72	1.00	12.32
0.55	11.69	1.10	11.66	1.10	12.17	1.10	12.26	1.10	12.09
0.60	11.90	1.20	11.64	1.20	12.10	1.20	11.63	1.20	12.27
0.65	11.97	1.30	11.89	1.30	11.71	1.30	11.70	1.30	12.23
0.70	12.39	1.40	12.10	1.40	11.84	1.40	11.96	1.40	12.18
0.75	11.75	1.50	12.24	1.50	11.84	1.50	12.35	1.50	12.29
0.80	12.33	1.60	12.44	1.60	12.14	1.60	11.68	1.60	12.41
0.85	11.63	1.70	12.34	1.70	11.77	1.70	12.03	1.70	12.04
0.90	12.40	1.80	11.95	1.80	12.12	1.80	11.61	1.80	12.13
0.95	12.19	1.90	12.21	1.90	12.29	1.90	12.43	1.90	11.65
1.00	12.29	2.00	11.68	2.00	11.92	2.00	12.04	2.00	12.41

Table 15. Minimization results for different values of ADSCFGWO's parameters (1).

r_1		r_2		r_3		r_4	
Values	**Fitness**	**Values**	**Fitness**	**Values**	**Fitness**	**Values**	**Fitness**
0.05	117.51	0.05	116.11	0.05	117.04	0.05	116.53
0.10	117.37	0.10	117.58	0.10	115.64	0.10	116.46
0.15	115.86	0.15	115.45	0.15	116.76	0.15	117.46
0.20	116.21	0.20	117.57	0.20	115.54	0.20	116.39
0.25	115.61	0.25	117.61	0.25	116.95	0.25	116.76
0.30	115.89	0.30	117.02	0.30	117.42	0.30	116.16
0.35	116.52	0.35	117.57	0.35	116.80	0.35	116.15
0.40	115.60	0.40	116.54	0.40	116.81	0.40	117.11
0.45	116.00	0.45	117.51	0.45	116.21	0.45	116.09
0.50	116.13	0.50	115.55	0.50	115.46	0.50	116.99
0.55	117.47	0.55	117.54	0.55	117.38	0.55	116.64
0.60	115.76	0.60	116.78	0.60	116.21	0.60	117.15
0.65	116.36	0.65	115.88	0.65	117.05	0.65	116.83
0.70	115.80	0.70	117.00	0.70	116.63	0.70	116.90
0.75	116.67	0.75	116.39	0.75	116.49	0.75	116.97
0.80	117.29	0.80	116.46	0.80	116.62	0.80	116.72
0.85	117.32	0.85	115.57	0.85	116.92	0.85	116.10
0.90	115.82	0.90	117.43	0.90	117.28	0.90	116.42
0.95	115.40	0.95	117.28	0.95	117.22	0.95	116.52
1.00	115.50	1.00	116.75	1.00	116.08	1.00	115.64

Table 16. Minimization results for different values of ADSCFGWO's parameters (2).

A_1		A_2		A_3		C_1		C_2	
Values	**Fitness**	**Values**	**Fitness**	**Values**	**Fitness**	**Values**	**Fitness**	**Values**	**Fitness**
0.05	116.77	0.10	115.88	0.10	116.87	0.10	116.45	0.10	117.25
0.10	116.62	0.20	117.05	0.20	116.45	0.20	117.06	0.20	115.41
0.15	117.38	0.30	117.58	0.30	116.97	0.30	116.05	0.30	115.71
0.20	117.53	0.40	115.96	0.40	117.58	0.40	115.97	0.40	117.01
0.25	116.93	0.50	115.64	0.50	116.53	0.50	117.66	0.50	115.44
0.30	117.54	0.60	115.48	0.60	116.67	0.60	116.82	0.60	116.53
0.35	116.24	0.70	116.61	0.70	116.47	0.70	116.93	0.70	115.99
0.40	116.83	0.80	117.02	0.80	116.67	0.80	116.16	0.80	117.20
0.45	115.75	0.90	116.78	0.90	116.39	0.90	117.17	0.90	115.40
0.50	117.03	1.00	115.74	1.00	116.78	1.00	117.24	1.00	116.36
0.55	115.36	1.10	115.43	1.10	116.82	1.10	117.02	1.10	115.67
0.60	116.75	1.20	115.82	1.20	117.17	1.20	117.28	1.20	115.84
0.65	116.77	1.30	116.93	1.30	116.63	1.30	115.83	1.30	117.19
0.70	116.93	1.40	115.50	1.40	115.76	1.40	116.47	1.40	116.32
0.75	115.98	1.50	115.92	1.50	116.40	1.50	116.04	1.50	116.02
0.80	116.82	1.60	117.51	1.60	116.88	1.60	115.79	1.60	117.02
0.85	115.55	1.70	117.36	1.70	115.55	1.70	115.96	1.70	116.86
0.90	115.79	1.80	117.35	1.80	116.52	1.80	117.10	1.80	116.63
0.95	116.55	1.90	115.80	1.90	117.28	1.90	116.05	1.90	116.78
1.00	117.50	2.00	116.91	2.00	115.82	2.00	116.44	2.00	116.54

5.6.1. Statistical Significance of the Results

The one-way analysis of variance (ANOVA) is performed to assess the significant difference between the proposed approach and other approaches. While modifying the settings of ADSCFGWO, two ANOVA tests are applied to both the convergence time and the fitness values. Table 17 shows the ANOVA test results for ADSCFGWO's convergence time and lowest fitness. Table 18 shows that *p*-values are less than (0.05) and F is larger than the F-critical level. Because of this, there is a statistically significant difference between the average of each parameter's five groups of convergence time. When each parameter's value is changed, a statistically significant difference can be seen between the means of all five minimal fitness groups. ANOVA does not tell which groups have statistical significance. As a result, a post hoc analysis is carried out between each pair of groups. A significance threshold of (0.05) was used for this purpose in a one-tailed *t*-test. Table 19 tested the algorithm's parameters using the *t*-test based on the convergence time and minimum fitness of ADSCFGWO. There is a statistically significant difference between groups, with *p*-values less than (0.05) according to the table. As for convergence time, there is a *t*-test between the exploration percentage and mutation rate that is statistically significant (0.05). This proposes that no statistically significant difference exists between their impacts on the time of convergence. The number of iterations or the mutation rate does not affect the minimal fitness. A visual representation of the study of the sensitivity of the algorithm parameters is represented by the plots shown in Figure 8. In this figure, the residual plot and homoscedasticity show the stability of the parameters. In addition, the QQ and heatmap plots show the robustness of the optimized parameters.

Table 17. ANOVA test analyzing the convergence time.

	SS	**DF**	**MS**	**F (DFn, DFd)**	*p* **Value**
Treatment	0.7603	8	0.09504	F (8, 171) = 1.336	p = 0.002288
Residual	12.17	171	0.07115		
Total	12.93	179			

Table 18. Wilcoxon signed-rank test of the convergence time.

	r_1	r_2	r_3	r_4	A_1	A_2	A_3	C_1	C_2
Number of values	20	20	20	20	20	20	20	20	20
Actual mean	12	11.94	11.9	12.06	12.06	11.95	11.98	11.99	12.12
Theoretical mean	0	0	0	0	0	0	0	0	0
df	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19
t	t = 209.6	t = 208.2	t = 200.1	t = 207.8	t = 196.0	t = 175.3	t = 222.7	t = 195.7	t = 205.8
p value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Discrepancy	12	11.94	11.9	12.06	12.06	11.95	11.98	11.99	12.12
Significant (alpha = 0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SEM of discrepancy	0.0572	0.0573	0.0594	0.0580	0.0615	0.0681	0.0538	0.0612	0.0588
SD of discrepancy	0.2561	0.2564	0.266	0.2596	0.2752	0.3048	0.2406	0.2739	0.2633
R squared	0.9996	0.9996	0.9995	0.9996	0.9995	0.9994	0.9996	0.9995	0.9996
95% confidence (From)	11.88	11.82	11.78	11.94	11.93	11.81	11.87	11.86	11.99
95% confidence (To)	12.12	12.06	12.03	12.18	12.19	12.09	12.09	12.12	12.24

Table 19. Statistical analysis of the convergence time.

	r_1	r_2	r_3	r_4	A_1	A_2	A_3	C_1	C_2
Number of values	20	20	20	20	20	20	20	20	20
Minimum	11.61	11.6	11.58	11.62	11.64	11.58	11.61	11.61	11.65
Range	0.830	0.760	0.820	0.830	0.760	0.860	0.800	0.820	0.760
25% Percentile	11.78	11.73	11.62	11.83	11.85	11.67	11.79	11.69	11.88
75% Percentile	12.21	12.19	12.11	12.26	12.37	12.25	12.17	12.26	12.32
Mean	12.0	11.94	11.90	12.06	12.06	11.95	11.98	11.99	12.12
Median	11.99	11.85	11.89	12.05	12.05	11.84	11.93	12.04	12.20
Maximum	12.45	12.37	12.41	12.46	12.41	12.45	12.41	12.43	12.42
Std. error of mean	0.057	0.057	0.059	0.058	0.061	0.068	0.053	0.061	0.058
Std. deviation	0.256	0.256	0.266	0.259	0.275	0.304	0.240	0.273	0.263
Sum	240.0	238.7	238.0	241.2	241.3	239.0	239.6	239.8	242.3
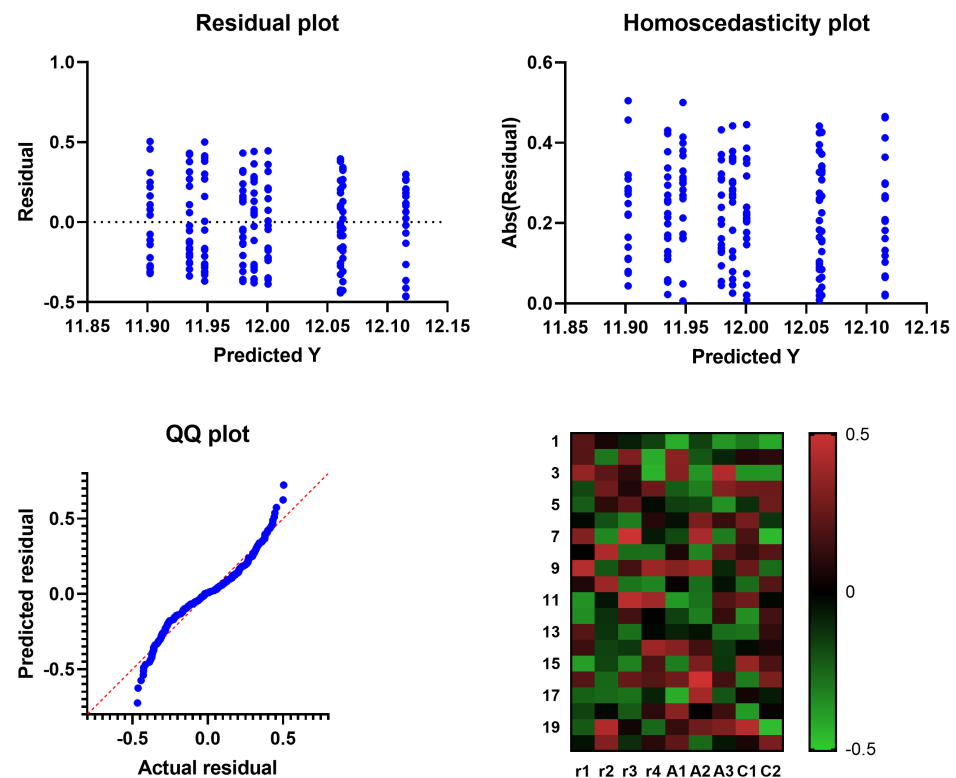
Figure 8. Residual, homoscedasticity, and QQ plots and heatmap of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2) based on convergence time.

The histogram depicted in Figure 9 shows the convergence time of the parameters of the proposed algorithm. In this figure, it can be noted that some parameters converge faster than others. For example, r_3, r_2, A_2, and C_1 converge faster than the other parameters. However, all the parameters converge in 12.4 s. In addition, the histogram of the convergence time of the proposed ADSCFGWO is depicted in Figure 10.
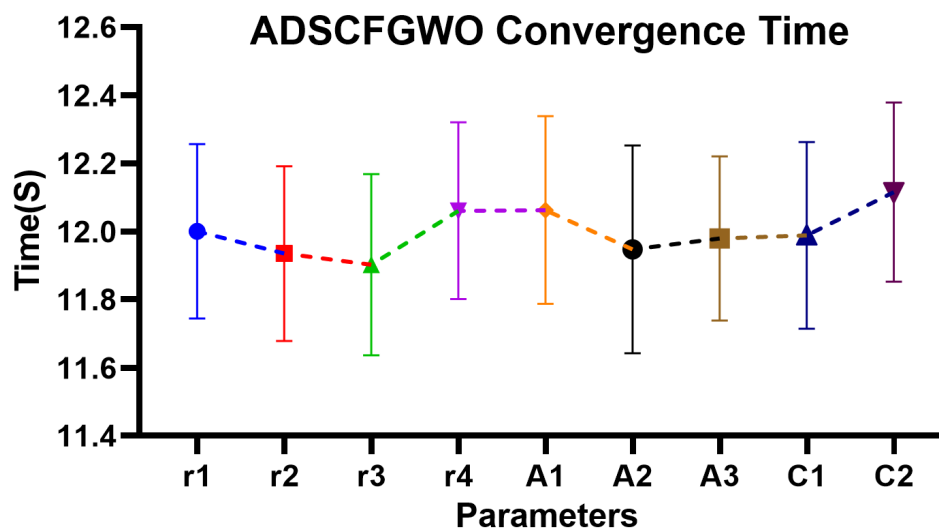

Figure 9. Convergence time of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2).
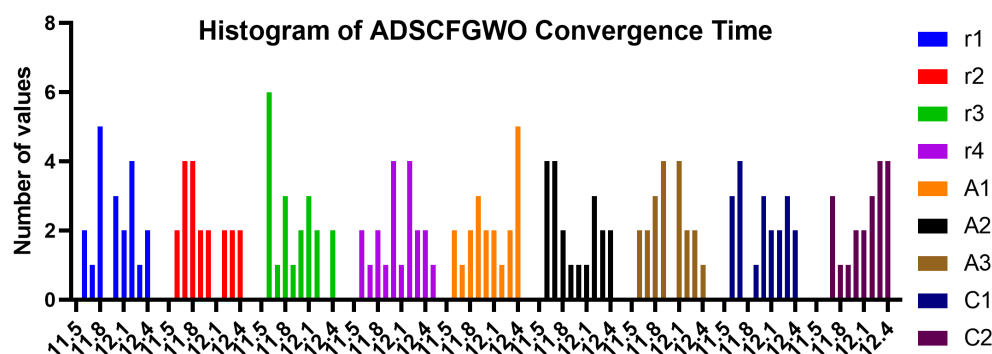
Figure 10. Histogram of convergence time of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2).

A study of the sensitivity of the fitness of the proposed approach is conducted, and the results are recorded in Tables 20–22. These tables represent the ANOVA test, the Wilcoxon test, and the statistical analysis of the achieved results. It can be noted from these tables that the parameters of the proposed algorithm as the *p*-value < 0.0001.

Table 20. Wilcoxon signed-rank test of the fitness of the proposed ADSCFGWO algorithm.

	r_1	r_2	r_3	r_4	A_1	A_2	A_3	C_1	C_2
Number of values	20	20	20	20	20	20	20	20	20
Actual mean	116.3	116.8	116.6	116.6	116.6	116.4	116.6	116.6	116.4
Theoretical mean	0	0	0	0	0	0	0	0	0
df	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19	df = 19
t	t = 721.8	t = 694.1	t = 872.0	t = 1178	t = 791.3	t = 685.0	t = 1051	t = 903.7	t = 818.3
Discrepancy	116.3	116.8	116.6	116.6	116.6	116.4	116.6	116.6	116.4
p value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Significant (alpha = 0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SEM of discrepancy	0.1611	0.1683	0.1337	0.09896	0.1474	0.1699	0.1109	0.129	0.1422
SD of discrepancy	0.7206	0.7525	0.5981	0.4426	0.6592	0.76	0.496	0.5769	0.6359
R squared	1	1	1	1	1	1	1	1	1
95% confidence (from)	116.0	116.4	116.4	116.4	116.3	116.1	116.4	116.3	116.1
95% confidence (to)	116.6	117.1	116.9	116.8	116.9	116.8	116.8	116.8	116.7

The fitness of the parameters of the proposed approach is analyzed, and the results are presented in Tables 20–22 in terms of the statistical analysis using Wilcoxon and ANOVA tests. These results show the effectiveness of the analyzed parameters in solving the optimization problem.

More investigation of the effectiveness of the parameters of the proposed approach is performed using the set of plots depicted in Figures 11–13. These figures show the significance of the parameters in the optimization problem and the convergence of the fitness.

Table 21. ANOVA test of the fitness of the proposed ADSCFGWO algorithm.

	SS	**DF**	**MS**	**F (DFn, DFd)**	*p* **Value**
Treatment	3.749	8	0.4686	F (8, 171) = 1.160	p = 0.003259
Residual	69.05	171	0.4038		
Total	72.8	179			

Table 22. Statistical analysis of the fitness achieved by the proposed optimization algorithm.

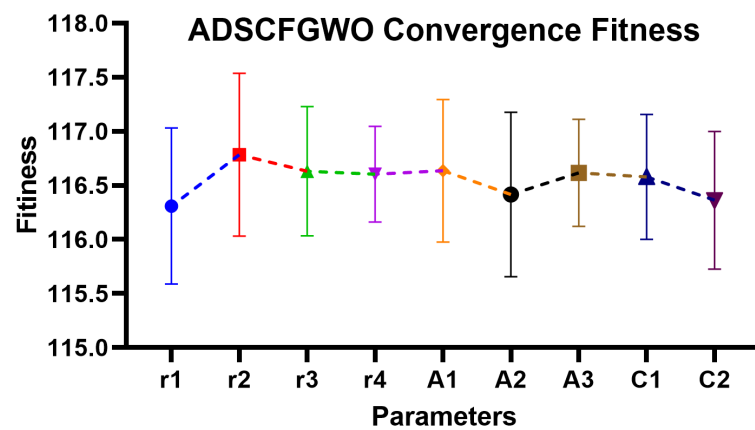|	r_1	r_2	r_3	r_4	A_1	A_2	A_3	C_1	C_2
Number of values	20	20	20	20	20	20	20	20	20
Range	2.108	2.158	1.967	1.824	2.176	2.148	2.029	1.87	1.851
25% Percentile	115.8	116.2	116.2	116.2	116.1	115.8	116.4	116.0	115.7
75% Percentile	117.1	117.5	117.1	117.0	117.0	117.0	116.9	117.1	117.0
Minimum	115.4	115.5	115.5	115.6	115.4	115.4	115.6	115.8	115.4
Mean	116.3	116.8	116.6	116.6	116.6	116.4	116.6	116.6	116.4
Median	116.1	116.9	116.8	116.6	116.8	116.3	116.7	116.5	116.4
Maximum	117.5	117.6	117.4	117.5	117.5	117.6	117.6	117.7	117.3
Std. error of mean	0.161	0.168	0.133	0.098	0.147	0.169	0.111	0.129	0.142
Std. deviation	0.721	0.752	0.598	0.442	0.659	0.760	0.496	0.577	0.636
Sum	2326	2336	2333	2332	2333	2328	2332	2332	2327

Figure 11. Convergence fitness of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2).
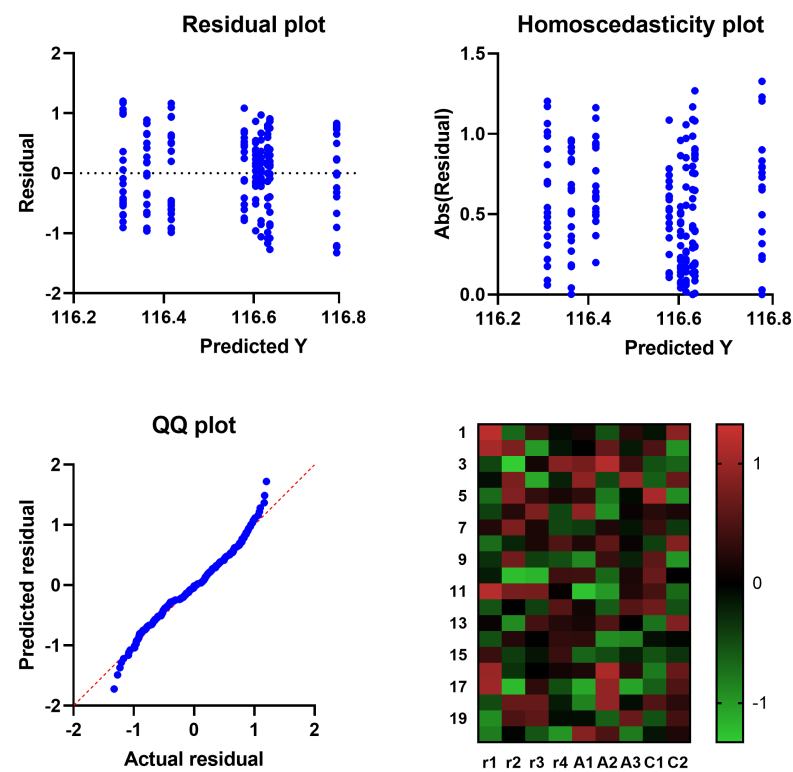

Figure 12. Residual, homoscedasticity, and QQ plots and heatmap of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2) based on convergence fitness.
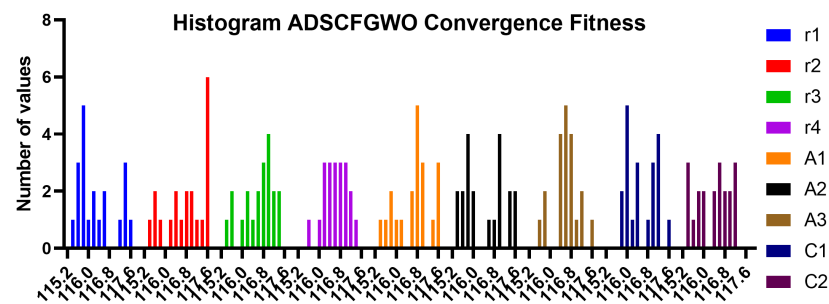

Figure 13. Histogram of convergence fitness of ADSCFGWO's parameters (r_1, r_2, r_3, r_4, A_1, A_2, A_3, C_1, and C_2).

5.6.2. Discussion and Ranking of Parameters

The parameters of the proposed ADSCFGWO algorithm can be ordered according to their effect on the fitness values as follows: C_2, A_1, r_4, r_1, C_1, A_3, A_2, r_2, and r_3. It is also possible to rank the following variables in the order of their impact on the convergence time: r_1, A_2, C_2, r_3, r_4, A_1, C_1, and r_2. The ADSCFGWO algorithm's convergence time is influenced by r_2. The value of r_2 has the least impact on the algorithm's convergence for all of these reasons. The ADSCFGWO algorithm's convergence time is strongly influenced by r_1, r_2, r_3, and A_2 values. The ADSCFGWO algorithm has a convergence time sensitive to exploration percentages larger than 25%. In terms of fitness, C_2 and A_1 significantly impact the algorithm's performance.

5.7. Discussion

A set of experimental setups is used to evaluate the effectiveness of the proposed methodology in identifying wheat/weed images. Firstly, positive results indicate the effectiveness of the features derived from the AlexNet architecture through transfer learning. The features retrieved from AlexNet are then used in a feature selection scenario. In the second scenario, the proposed ADSCFGWO algorithm proves both stable and dependable in its quest to identify the best possible collection of features in a reasonable period. In addition, the Wilcoxon rank-sum test highlights the relevance of the proposed ADSCFGWO algorithm by demonstrating its statistical significance. On the other hand, other experiments are conducted to demonstrate that the proposed optimized voting classifier outperforms the competing methods when classifying the input crop image, with a mean accuracy of (97.75%). A sensitivity analysis is carried out to ensure the proposed method is reliable. Testing and results show that the proposed technique is highly effective in classifying wheat/weed images.

6. Conclusions

This paper proposes a new approach to classify wheat and weed in drone-captured images based on metaheuristic optimization and machine learning. The proposed approach is based on a new optimized voting classifier that could efficiently classify the features extracted using AlexNet. To boost the classification accuracy, the extracted features are optimized to select the significant features based on a new binary optimization algorithm. The optimization of the voting classifier and the binary optimization algorithm developed for feature selection are based on the GWO and SCA optimization algorithms in a new hybrid optimization algorithm referred to as the ADSCFGWO algorithm. The proposed voting classifier comprises three machine-learning models: NN, SVM, and KNN. These classifiers' contribution to the final results is optimized using the proposed optimization algorithm. The proposed approach's efficiency was evaluated using various metrics, including accuracy, precision, recall, false positive rate, and kappa coefficient. In addition, the ANOVA and Wilcoxon signed-rank tests are used to assess the reliability and validity of the proposed methodology. Moreover, a sensitivity analysis is carried out to investigate the impact of varying parameters of the proposed approach on the observed outcomes. The proposed methodology was superior to existing optimization strategies in a series of experiments with a detection accuracy of 97.70%, an F-score of 98.60%, a specificity of 95.20%, and a sensitivity of 98.40%. From the statistical analysis, the ANOVA and Wilcoxon signed-rank tests showed the value of p as ($p < 0.005$), indicating the proposed approach's statistical difference.

Author Contributions: Conceptualization, E.-S.M.E.-K. and N.K.; methodology, S.M.; software, T.M.; validation, M.A., F.K.K. and H.K.A.; formal analysis, A.A.A.; investigation, M.M.E.; resources, T.H.; data curation, A.I.; writing—original draft preparation, A.A.A.; writing—review and editing, A.A.A.; visualization, D.S.K.; supervision, D.S.K.; project administration, E.-S.M.E.-K.; funding acquisition, H.K.A. All authors have read and agreed to the published version of the manuscript.

References

1.	Singh, A.; Ganapathysubramanian, B.; Singh, A.K.; Sarkar, S. Machine Learning for High-Throughput Stress Phenotyping in Plants. *Trends Plant Sci.* **2016**, *21*, 110–124. [CrossRef] [PubMed]
2.	Dadashzadeh, M.; Abbaspour-Gilandeh, Y.; Mesri-Gundoshmian, T.; Sabzi, S.; Hernández-Hernández, J.L.; Hernández-Hernández, M.; Arribas, J.I. Weed Classification for Site-Specific Weed Management Using an Automated Stereo Computer-Vision Machine-Learning System in Rice Fields. *Plants* **2020**, *9*, 559. [CrossRef]
3.	Alam, M.; Alam, M.S.; Roman, M.; Tufail, M.; Khan, M.U.; Khan, M.T. Real-Time Machine-Learning Based Crop/Weed Detection and Classification for Variable-Rate Spraying in Precision Agriculture. In *Proceedings of the 2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*; IEEE: Antalya, Turkey, 2020; pp. 273–280. [CrossRef]
4.	Tu, Y.H.; Johansen, K.; Phinn, S.; Robson, A. Measuring Canopy Structure and Condition Using Multi-Spectral UAS Imagery in a Horticultural Environment. *Remote Sensing* **2019**, *11*, 269. [CrossRef]
5.	Gao, J.; Nuyttens, D.; Lootens, P.; He, Y.; Pieters, J.G. Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery. *Biosyst. Eng.* **2018**, *170*, 39–50. [CrossRef]
6.	Myers, S.S.; Smith, M.R.; Guth, S.; Golden, C.D.; Vaitla, B.; Mueller, N.D.; Dangour, A.D.; Huybers, P. Climate Change and Global Food Systems: Potential Impacts on Food Security and Undernutrition. *Annu. Rev. Public Health* **2017**, *38*, 259–277. [CrossRef] [PubMed]
7.	Aharon, S.; Peleg, Z.; Argaman, E.; Ben-David, R.; Lati, R.N. Image-Based High-Throughput Phenotyping of Cereals Early Vigor and Weed-Competitiveness Traits. *Remote Sensing* **2020**, *12*, 3877. [CrossRef]
8.	Herrmann, I.; Bdolach, E.; Montekyo, Y.; Rachmilevitch, S.; Townsend, P.A.; Karnieli, A. Assessment of maize yield and phenology by drone-mounted superspectral camera. *Precis. Agric.* **2020**, *21*, 51–76. [CrossRef]
9.	Juan-wei, Z. Review of Mechanical Weeding Technique in Field at Home and Abroad. *J. Agric. Mech. Res.* **2006**, *10*, 14–16.
10.	Barbosa, P.F.P. Voltammetric Techniques for Pesticides and Herbicides Detection- an Overview. *Int. J. Electrochem. Sci.* **2019**, *14*, 3418–3433. [CrossRef]
11.	Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D.D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. [CrossRef]
12.	Forero, M.G.; Herrera-Rivera, S.; Ávila-Navarro, J.; Franco, C.A.; Rasmussen, J.; Nielsen, J. Color Classification Methods for Perennial Weed Detection in Cereal Crops. In Proceedings of the CIARP, Madrid, Spain, 19–22 November 2018.
13.	Dyrmann, M.; Karstoft, H.; Midtiby, H.S. Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **2016**, *151*, 72–80. [CrossRef]
14.	Milioto, A.; Lottes, P.; Stachniss, C. Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2229–2235. [CrossRef]
15.	dos Santos Ferreira, A.; Freitas, D.M.; da Silva, G.G.; Pistori, H.; Folhes, M.T. Weed detection in soybean crops using ConvNets. *Comput. Electron. Agric.* **2017**, *143*, 314–324. [CrossRef]
16.	Ishak, A.J.; Hussain, A.; Mustafa, M.M. Weed image classification using Gabor wavelet and gradient field distribution. *Comput. Electron. Agric.* **2009**, *66*, 53–61. [CrossRef]
17.	OKAMOTO, H.; MURATA, T.; KATAOKA, T.; HATA, S.I. Plant classification for weed detection using hyperspectral imaging with wavelet analysis. *Weed Biol. Manag.* **2007**, *7*, 31–37. [CrossRef]
18.	Goel, P.; Prasher, S.; Patel, R.; Landry, J.; Bonnell, R.; Viau, A. Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn. *Comput. Electron. Agric.* **2003**, *39*, 67–93. [CrossRef]
19.	Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process. Agric.* **2020**, *7*, 1–19. [CrossRef]
20.	Wang, A.; Zhang, W.; Wei, X. A review on weed detection using ground-based machine vision and image processing techniques. *Comput. Electron. Agric.* **2019**, *158*, 226–240. [CrossRef]
21.	Herrmann, I.; Shapira, U.; Kinast, S.; Karnieli, A.; Bonfil, D.J. Ground-level hyperspectral imagery for detecting weeds in wheat fields. *Precis. Agric.* **2013**, *14*, 637–659. [CrossRef]

22. Weis, M.; Gutjahr, C.; Rueda Ayala, V.; Gerhards, R.; Ritter, C.; Schölderle, F. Precision farming for weed management: Techniques. *Gesunde Pflanz.* **2008**, *60*, 171–181. [CrossRef]

23. Abdelhamid, A.; Alotaibi, S. Optimized Two-Level Ensemble Model for Predicting the Parameters of Metamaterial Antenna. *Comput. Mater. Contin.* **2022**, *73*, 917–933. [CrossRef]

24. Chabot, D.; Dillon, C.; Shemrock, A.; Weissflog, N.; Sager, E. An Object-Based Image Analysis Workflow for Monitoring Shallow-Water Aquatic Vegetation in Multispectral Drone Imagery. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 294. [CrossRef]

25. Brinkhoff, J.; Vardanega, J.; Robson, A.J. Land Cover Classification of Nine Perennial Crops Using Sentinel-1 and -2 Data. *Remote Sensing* **2019**, *12*, 96. [CrossRef]

26. Zhang, S.; Guo, J.; Wang, Z. Combing K-means Clustering and Local Weighted Maximum Discriminant Projections for Weed Species Recognition. *Front. Comput. Sci.* **2019**, *1*, 4. [CrossRef]

27. Bakhshipour, A.; Jafari, A. Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Comput. Electron. Agric.* **2018**, *145*, 153–160. [CrossRef]

28. Abouzahir, S.; Sadik, M.; Sabir, E. Enhanced Approach for Weeds Species Detection Using Machine Vision. In *Proceedings of the 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*; IEEE: Kenitra, Morocco, 2018; pp. 1–6. [CrossRef]

29. Kazmi, W.; Garcia-Ruiz, F.J.; Nielsen, J.; Rasmussen, J.; Jørgen Andersen, H. Detecting creeping thistle in sugar beet fields using vegetation indices. *Comput. Electron. Agric.* **2015**, *112*, 10–19. [CrossRef]

30. Fernandez-Gallego, J.A.; Lootens, P.; Borra-Serrano, I.; Derycke, V.; Haesaert, G.; Roldán-Ruiz, I.; Araus, J.L.; Kefauver, S.C. Automatic wheat ear counting using machine learning based on RGB UAV imagery. *Plant J.* **2020**, *103*, 1603–1613. [CrossRef]

31. Etienne, A.; Saraswat, D. Machine learning approaches to automate weed detection by UAV based sensors. In *Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV*; Thomasson, J.A., McKee, M., Moorhead, R.J., Eds.; SPIE: Baltimore, ML, USA, 2019; p. 25. [CrossRef]

32. Islam, N.; Rashid, M.M.; Wibowo, S.; Xu, C.Y.; Morshed, A.; Wasimi, S.A.; Moore, S.; Rahman, S.M. Early Weed Detection Using Image Processing and Machine Learning Techniques in an Australian Chilli Farm. *Agriculture* **2021**, *11*, 387. [CrossRef]

33. Pérez-Ortiz, M.; Peña, J.M.; Gutiérrez, P.A.; Torres-Sánchez, J.; Hervás-Martínez, C.; López-Granados, F. Selecting patterns and features for between- and within- crop-row weed mapping using UAV-imagery. *Expert Syst. Appl.* **2016**, *47*, 85–94. [CrossRef]

34. Ahmed, F.; Al-Mamun, H.A.; Bari, A.H.; Hossain, E.; Kwan, P. Classification of crops and weeds from digital images: A support vector machine approach. *Crop. Prot.* **2012**, *40*, 98–104. [CrossRef]

35. Armstrong, G. Open Sprayer Images. 2018. Available online: https://www.kaggle.com/datasets/gavinarmstrong/open-sprayer-images (accessed on 1 November 2022).

36. Ateeq, T.; Majeed, M.N.; Anwar, S.M.; Maqsood, M.; ur Rehman, Z.; Lee, J.W.; Muhammad, K.; Wang, S.; Baik, S.W.; Mehmood, I. Ensemble-classifiers-assisted detection of cerebral microbleeds in brain MRI. *Comput. Electr. Eng.* **2018**, *69*, 768–781. [CrossRef]

37. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]

38. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]

39. Eid, M.M.; El-kenawy, E.S.M.; Ibrahim, A. A binary Sine Cosine-Modified Whale Optimization Algorithm for Feature Selection. In Proceedings of the National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 27–28 March 2021; pp. 1–6. [CrossRef]

40. Abdelhamid, A.A.; El-Kenawy, E.S.M.; Khodadadi, N.; Mirjalili, S.; Khafaga, D.S.; Alharbi, A.H.; Ibrahim, A.; Eid, M.M.; Saber, M. Classification of Monkeypox Images Based on Transfer Learning and the Al-Biruni Earth Radius Optimization Algorithm. *Mathematics* **2022**, *10*, 3614. [CrossRef]

41. Xian, H.; Yang, C.; Wang, H.; Yang, X. A Modified Sine Cosine Algorithm With Teacher Supervision Learning for Global Optimization. *IEEE Access* **2021**, *9*, 17744–17766. [CrossRef]

42. El-Kenawy, E.S.M.; Mirjalili, S.; Ibrahim, A.; Alrahmawy, M.; El-Said, M.; Zaki, R.M.; Eid, M.M. Advanced Meta-Heuristics, Convolutional Neural Networks, and Feature Selectors for Efficient COVID-19 X-Ray Chest Image Classification. *IEEE Access* **2021**, *9*, 36019–36037. [CrossRef] [PubMed]

43. El-Kenawy, E.S.M.; Eid, M.M.; Saber, M.; Ibrahim, A. MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection. *IEEE Access* **2020**, *8*, 107635–107649. [CrossRef]

44. Shi, X.; Huang, Q.; Chang, J.; Wang, Y.; Lei, J.; Zhao, J. Optimal parameters of the SVM for temperature prediction. *Proc. Int. Assoc. Hydrol. Sci.* **2015**, *368*, 162–167. [CrossRef]

45. El-Kenawy, E.S.M.; Ibrahim, A.; Mirjalili, S.; Eid, M.M.; Hussein, S.E. Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images. *IEEE Access* **2020**, *8*, 179317–179335. [CrossRef]

46. Ibrahim, A.; Tharwat, A.; Gaber, T.; Hassanien, A.E. Optimized superpixel and AdaBoost classifier for human thermal face recognition. *Signal, Image Video Process.* **2018**, *12*, 711–719. [CrossRef]

47. Maqsood, M.; Nazir, F.; Khan, U.; Aadil, F.; Jamal, H.; Mehmood, I.; Song, O.y. Transfer Learning Assisted Classification and Detection of Alzheimer's Disease Stages Using 3D MRI Scans. *Sensors* **2019**, *19*, 2645. [CrossRef]

48. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

49. Al-Tashi, Q.; Abdul Kadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection. *IEEE Access* **2019**, *7*, 39496–39508. [CrossRef]

50. Madadi, A.; Motlagh, M.M. Optimal Control of DC motor using Grey Wolf Optimizer Algorithm. *Tech. J. Eng. Appl. Sci.* **2014**, *4*, 373–379.
51. El-kenawy, E.S.; Eid, M. Hybrid Gray Wolf and Particle Swarm Optimization for Feature Selection. *Int. J. Innov. Comput. Inf. Control. IJICIC* **2020**, *16*, 831–844. [CrossRef]
52. Şenel, F.A.; Gökçe, F.; Yüksel, A.S.; Yigit, T. A novel hybrid PSO–GWO algorithm for optimization problems. *Eng. Comput.* **2019**, *35*, 1359–1373. [CrossRef]
53. Bello, R.; Gomez, Y.; Nowe, A.; Garcia, M.M. Two-Step Particle Swarm Optimization to Solve the Feature Selection Problem. In Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, Brazil, 20–24 October 2007; pp. 691–696. [CrossRef]
54. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
55. Fister, I.; Yang, X.S.; Fister, I.; Brest, J. Memetic firefly algorithm for combinatorial optimization. *arXiv* **2012**, arXiv:1204.5165.
56. Kabir, M.M.; Shahjahan, M.; Murase, K. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* **2011**, *74*, 2914–2928. [CrossRef]