Pak. J. Agri. Sci., Vol. 58(3), 1007-1015;2021
ISSN (Print) 0552-9034, ISSN (Online) 2076-0906
DOI: 10.21162/PAKJAS/21.479
http://www.pakjas.com.pk

A NOVEL MACHINE LEARNING BASED ALGORITHM TO DETECT WEEDS IN SOYBEAN CROP

Rameen Sohail[1], Qamar Nawaz[1], Isma Hamid[2,*], Humair Amin[3], Junaid Nawaz Chauhdary[4], Syed Mushhad Mustuhzar Gilani[5] and Imran Mumtaz[1]

[1]Department of Computer Science, University of Agriculture, Faisalabad, Pakistan; [2]Department of Computer Science, National Textile University Faisalabad, Pakistan; [3]Dept of Sociology and Criminology, University of Sargodha; [4]Water Management Research Centre, University of Agriculture, Faisalabad, Pakistan; [5]University Institute of Information Technology PMAS-Arid Agriculture University Rawalpindi, Pakistan
***Corresponding author's e-mail: ismahamid@ntu.edu.pk**

The traditional ways of weed management including spraying of herbicides on the whole field and manually uprooting them, are still in practice in many agricultural farms. This leads to herbicide overuse that causes serious health issues due to food quality degradation and environmental pollution. Computer aided weed detection systems can help in smart utilization of herbicides by detecting weeds through images. The aim of this study is to propose a novel weed detection system that provides accurate results in recognizing crops and weed using Machine Learning and Image Processing techniques. The image dataset chosen for this work is comprised of four different classes including broadleaf weed, grass, soil, and soybean. The proposed algorithm extracts texture and color features from each image in dataset and uses Random Forest algorithm to train a model using extracted feature descriptors. The working of the model is evaluated by computing regression metrics, precision, recall and F1 scores. Results showed that the model achieved a correct classification accuracy of 91% for weed, 100% for soil, 90% for grass and 99% for the soybean crop. The complete program took only 80 sec to execute which is ideal for a real-time environment.

Keywords*:* Image processing techniques, machine learning, automatic weed detection, machine vision, robotic weed control.

INTRODUCTION

Weeds refer to the unwanted plants that tend to grow along with the crops. These undesirable plants compete for sunlight, water and essential nutrients that results in the weakening of the crop plants and hence, the crops become more susceptible to disease and pest infestation. Some of the weed species are poisonous which, in severe cases, can cause death if consumed by human or animals (Azadbakht and Mana, 2003). In addition, pollens released by some of the weeds like Convolvulus Arvensis, Oxalis Corniculata, Sorghum Halepense, Ragweed, Pigweed, Sagebrush, Tumbleweed, and Lambs' Quarter are highly allergic for humans (Khan *et al.*, 2020). Another major problem caused by weeds is drastic decrease in the yield. To increase the quality and production of the crop, a suitable weeding methodology is required to be implemented according to the size of the field. For small fields, weeds can be removed manually whereas for large fields, mechanical or chemical methods are adopted. Specially for the large fields, manual and mechanical weeding is time consuming process that requires large labor force which makes the process expensive and tedious along with the greater chances of crop damage by the mechanical tools (Metwally and Wakeel, 2019). Chemical weeding methods aim at spraying herbicides directly on the weeds either to kill them or to stop their growth. However, spraying herbicides on weed has its drawbacks. In practice, weeding chemicals are applied on the whole field instead of applying them specifically on the weed plants which boosts the cost of application. In addition, spraying excessive herbicides on crops may have adverse effects on crop health and the environment (Marshall *et al.*, 2019).

Artificial Intelligence (AI) has brought a great revolution in almost every industry including security (Abdalla *et al.*, 2016), Big Data management (Abdalla *et al.*, 2020), agriculture, and many more. In combination with other Information Technology tools and AI, Precision Agriculture (PA) has revolutionized the traditional ways of weed management (Chlingaryan *et al.*, 2018). PA ensures that the crops and soil receive exactly what is necessary for their optimum productivity and growth and posed new ways to protect crop yield from various concerning factors including climate change, population growth, and other food security problems (Talaviya *et al.*, 2020). By collecting and analyzing aerial images of fields for site-specific herbicide application, the amount of herbicide can be decreased up to 80% (Andújar *et al.*, 2019). This would also eliminate the risk of crop damage, environmental pollution, the resistance of pests to chemicals, and product contamination. Weed detection systems are trained through efficient ML algorithms based on

image processing to differentiate between target weeds and non-weeds thus, making real-time and precise predictions (Partel *et al.*, 2019; Wang *et al.*, 2019).

In this study, an AI based weed detection system has been proposed aiming to reduce economic loss, herbicide usage, and the cost of weeding while improving quality crops and production. The proposed system utilizes image processing and machine learning techniques to extract, compare, and analyze features from the images of the field.

LITERATURE REVIEW

A novel approach was proposed in Slaughter *et al.* (2008) for the automatic classification of leafy vegetables and weeds with the help of a robot. The technique used by them was named "Crop Signaling" in which all the crop plants were marked with a signaling compound that was easily readable by the machine. This method took 1.2 seconds to process a pair of images with 99% accuracy for crop classification and 98.11% for weed classification.

A weed detection system was proposed in Ishak *et al.* (2008) for the successful identification of broad and narrow wed plants. The model was based on a Support Vector Machine (SVM) classifier that utilizes a combination of Gabor and FFT filter to extract features. The accuracy was initially 90% which was improved later to 100% by tuning the parameters.

The method presented in Masuda *et al.* (2010) utilizes two feature categories to identify rice crops from weed plants. The researchers had extracted area and moment of order features separately from all pixels and then observe the results. Results showed that the extracted third-moment order features generate better results as compared to the area.

The research work presented in Dyrmann *et al.* (2016) utilized a Deep Convolutional Neural Network to segment 22 different plant species at early growth stages. A 5x5 convolutional layer was applied after segmenting foreground from background (soil) pixels. Batch normalization was done to bring all the input layers in the same range. The activation function used in the work was Rectified Linear Unit (ReLU). After that, filter capacity and coverage were determined to decide the features that should be mapped and the area of an image to be covered, respectively. The training process was completed after 18 epochs with a batch size of 200 images. Classification accuracy of 86.2% was achieved using this technique that could be improved with more training samples.

Another technique discussed in Pulido *et al.* (2017) used texture features to differentiate between weeds and vegetable crops. For this purpose, a Gray-Level Co-occurrence Matrix was generated to compute texture features namely correlation, contrast, autocorrelation, dissimilarity, energy, homogeneity, difference variance and variance. After that, only those features were kept which showed maximum variance between weed and crop plants. The model was trained using the SVM classifier. This model achieved greater than 90% specificity and sensitivity values.

The model elaborated in Gao *et al.* (2018) utilized spectral features to detect weed in maize crops. The feature set was comprised of 80 Normalized Difference Vegetation Index (NDVI) and 80 Ratio Vegetation Index (RVI) feature descriptors. The research then selected only those features that provide maximum information by using Principal Component Analysis (PCA). The model was trained using two algorithms including K-Nearest Neighbor and Random Forest. Results showed that Random Forest performed better than KNN with a maize classification accuracy of 1 and 0.70, 0.79, and 0.75 for three weed species.

PROPOSED METHOD

The model in the proposed study is developed for automatic weed identification in soybean crops using Image Processing and ML techniques. This algorithm involves training the model through labelled image data and then evaluating it using a validation method. Once the model is trained and validated, it can predict new unseen images based on the training knowledge.

Data Acquisition: The image dataset is downloaded from an online community of data scientist and ML practitioners known as "Kaggle" (Peccia, 2018). The dataset consists of a total of 15,336 images comprising four different classes. 200 images are randomly selected for each class which accounts for 800 images in total. These classes include broadleaf weed, soybean crop, grass and soil images. The images are stored in their respective directories in ".TIF" format. This model is implemented in Python language using an open-source Integrated Development Environment known as Scientific Python Development Environment (SPYDER). To achieve this goal the Python version used is "Python 3.7.6".

Figure 1 exhibits a sample of raw image data containing the four classes i.e. Broadleaf weed, Grass, Soil, and Soybean. All the images are of different sizes and are captured under varying illumination which can natively affect classification result. To get rid of this problem, the proposed algorithm processes texture features in addition to color features. In this way, one feature category can overcome the limitations of other categories.

Image Resizing: The images are then rescaled to a fixed size of 300 x 300. This standard size was chosen because the width and height of most of the image's range between 200 to 350 pix. Therefore, rescaling them to a fixed size did not make them blur or affect their quality. Image resizing is done through "Interpolation". This would move pixels from one grid to the other. As both image height and width are affected, therefore, interpolation is done in two directions. Image zooming and shrinkage involves adding and replacing old pixels with new pixels having intensities calculated by approximating intensities of surrounding pixels. This task is

achieved through "Non-Adaptive Interpolation" where surrounding pixels are given equal importance in deciding new pixel intensities. It further utilizes bilinear interpolation that considers a linear combination of four surrounding input pixels. Equation (1) represents the bilinear interpolation applied to a unit square.

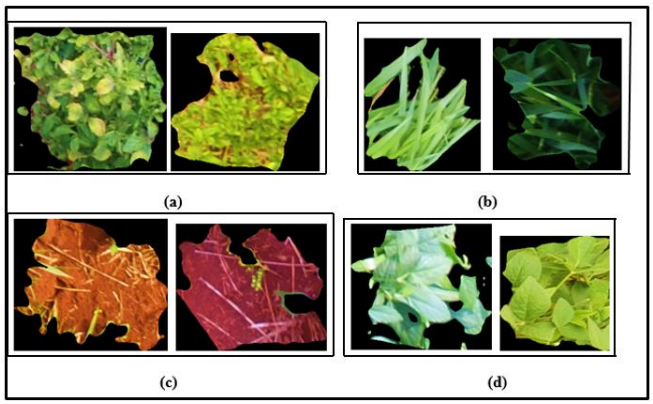

Figure 1. Sample of raw images from the dataset. (a) Broadleaf images, (b) Grass images, (c) Soil images, (d) Soybean images

$$F(x,y) = z00 * (1-x) * (1-y) + z10 * x * (1-y) + z01 * (1-x) * y + z11 * x * y \quad (1)$$

where, F(x,y) represents a point in the unit square matrix. Whole value is calculated by considering the weighted average of the four surrounding pixels z00, z10, z01 and z11. Figure 2 represents the sample of rescaled images from the dataset.

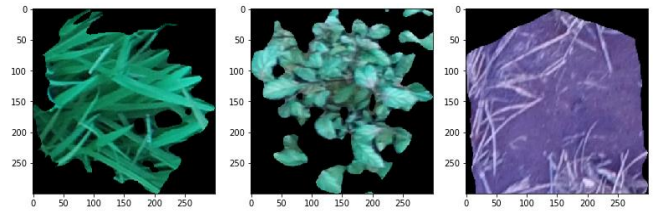

Figure 2. Sample of rescaled images with the x-axis representing the no. of pixels on the x-axis and y-axis showing the no. of pixels on the y-axis. (a) Resized grass image, (b) Resized broadleaf image, (c) Resized soil image

Conversion to Grayscale: After resizing, all the images are converted to greyscale. As a result, it will consume less memory and speed up the training process. To produce grey images from colored RGB images, we have utilized all the three red, green, and blue color channels unlike the typical way of using only one channel. This is because a single channel greyscale image may end up looking dull and thus it may lose a lot of information necessary to recognize textural features. The formula used to calculate the grayscale pixel value is given in (2). Figure 3 represents the results after performing grayscale conversion to the resized RGB image samples.

$$F_g(x,y) = 0.33 * F_B(x,y) + 0.56 * F_G(x,y) + 0.11 * F_R(x,y) \quad (2)$$

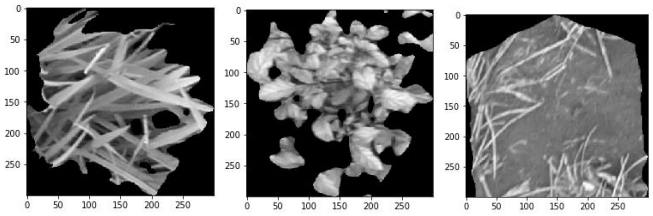

Figure 3. Grayscale conversion on image samples. (a) resized grayscale grass image, (b) resized grayscale broadleaf image, (c) resized grayscale soil image.

Noise Reduction: The next step is to reduce image noise. The technique used in this study is known as Bilateral filtering. This technique is utilized not only to remove image noise but also to preserve useful information about the edges and corners in an image. It applies a flat kernel bilateral filter to the image and then based on the radiometric resolution similarity and spatial closeness of pixels, it calculates the average of their intensity values. The spatial closeness is measured by a structuring element considering the local pixel neighborhood. The pixel values are averaged to have a noise-free image. Its equation is given in (3):

$$BF[I]_p = {}^{1}/_{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (3)$$

where ${}^{1}/_{W_p}$ denotes the normalization factor, $G_{\sigma_s}(\|p-q\|)$ represents the space weight and $G_{\sigma_r}(|I_p - I_q|)$ specifies the Range weight. Figure 4 represents zoomed-in version of dataset images before and after applying the mean-bilateral noise reduction on all four classes of images.

Feature extraction: Mainly there are four prime feature categories i.e. spatial, spectral, morphological, and visual features. Spatial refers to the space or location of the concerned object, spectral refers to the color and light characteristics, visual refers to the object texture features, and morphological features refers to the object area or shape. To generate good classification results, the model needs to extract features that belong to different feature categories. In this way, one feature category may overcome the limitations posed by the other one. For example, it is possible to have a similar texture in different plants therefore, color and shape features would be very helpful in generating a good feature set. For this reason, we have chosen color and texture features after various trials.

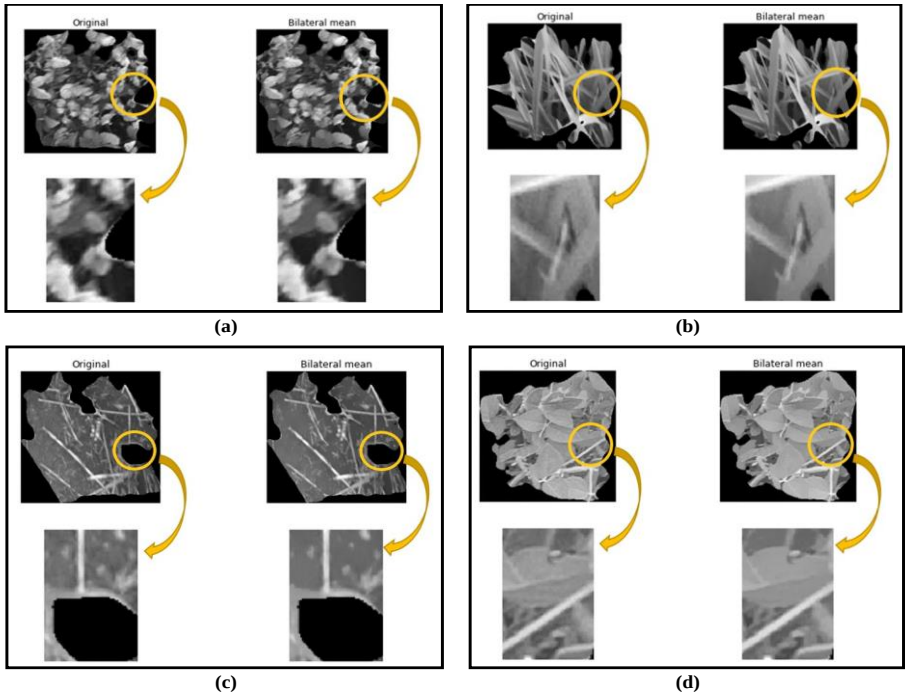

Figure 4. Noise reduction on sample images. (a) normal and zoomed-in version of the original and bilateral broadleaf image, (b) normal and zoomed-in version of original and bilateral grass image, (c) normal and zoomed-in version of original and bilateral soil image, (d) normal and zoomed-in version of original and bilateral soybean image.

Color or spectral features: Color features can be extremely helpful in detecting plants in an image because of the green stem and leaves, they can be easily distinguished from a dark background or soil. In the proposed work, a "color histogram" is used to extract color features. This method works by calculating number of occurrences a color within an image. This technique can be applied to various color models including HSV, RGB, grayscale, or hyperspectral model. We have chosen the HSV color model for this study. Figure 5 represents the sample image of the soybean plant which is then converted to the HSV color space to generate its respective color histogram.

As the images were originally in RGB color-space, so they are converted to an HSV color-space. The method of generating histograms of an HSV image is almost similar to that of an RGB image. The reason why we have chosen this color space is that the Hue, saturation, and intensity provide better spectral information as compared to an RGB image. The image's Hue information is divided into 8 groups, saturation is in 2 groups and intensity in 4 groups where every group creates its feature vector. This grouping is done because maximum feature information is provided by the Hue component followed by the intensity and then saturation of the HSV image. The value of bins is set to 8. This means that the HSV histogram is divided into 8 parts and the number of times a pixel occurs in its corresponding part or distribution is calculated. To enhance the fine and tiny details of the image, the histogram is normalized. This would also prevent biased results and keep the contribution of various bins relative.

Texture Features: After color features, texture features are extracted to obtain information about the ridges and edges of the plant. Textures can be extracted in various ways. The one chosen in this research is by using a Gray Level Co-occurrence Matrix (GLCM). This method utilizes a greyscale image and computes the co-occurrence of an ROI (Region of Interest) and its neighboring pixels. A pixel adjacency with neighboring pixels is calculated in four directions i.e. right and left diagonals, vertical, and horizontal in a square 2-D matrix. These directions are shown in Figure 6. This matrix does not directly give texture information rather, it can be used to calculate various image features that correspond to image texture information.

In this proposed methodology, the image texture feature information is extracted by calculating the contrast, dissimilarity, correlation, and Angular Second Moment (ASM). ASM corresponds to the textural uniformity of an image. The second feature contrast refers to the measure of local variation in an image. The contrast of a pixel with its neighboring pixels is calculated. The next texture feature is correlation which measures the linear dependency of the concerned pixel with the neighbor pixels in GLCM. Its value

ranges from -1 to 1. If the pixels are negatively correlated, then the value of correlation will be -1 and vice versa. Lastly, dissimilarity calculates the distance between the pixel and its surrounding pixels. The ASM, contrast, correlation and dissimilarity values are computed using (4), (5), (6), and (7) where asm represents the ASM value, $cont$ shows contrast, $corr$ represents correlation and $diss$ corresponds to dissimilarity of the matrix p with axis i, j.

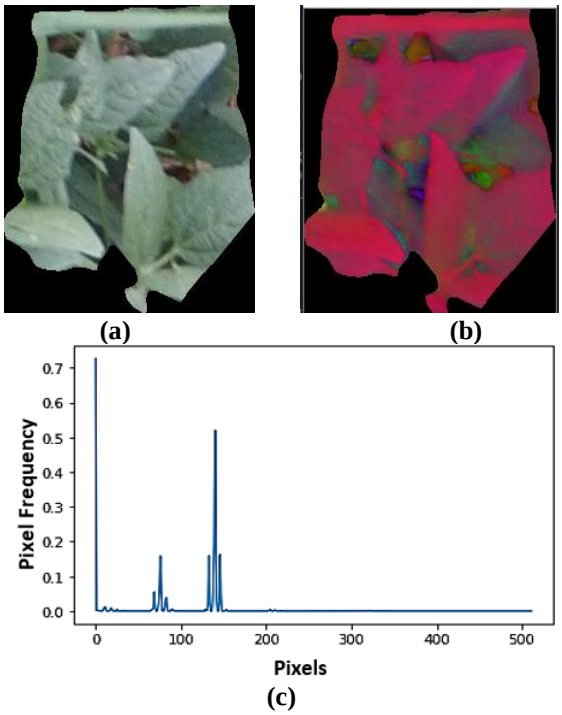

Figure 5. Soybean plant image converted to HSV. (a) Sample soybean plant image, (b) the transformed HSV image, (c) normalized histogram of the image

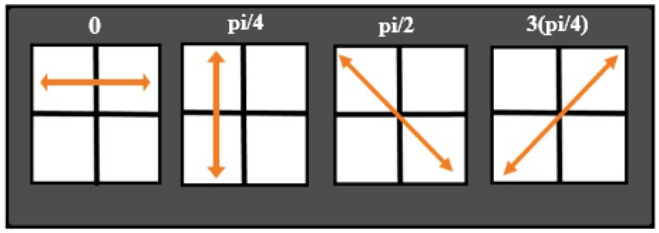

Figure 6. Adjacency directions to calculate co-occurrence of pixel with its neighboring pixels.

$$asm = \sum_i \sum_j p(i,j)^2 \quad (4)$$

$$cont = \sum_i \sum_j (i-j)^2 p(i,j) \quad (5)$$

$$corr = -\left. \sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y \right/ \sigma_x \sigma_y \quad (6)$$

$$diss = \sum_i \sum_j p(i,j)|i-j| \quad (7)$$

Combining and Rescaling Features: Once the features are extracted, they are then combined into a single matrix using the "append" method. It works by updating an existing list by adding an item to its end. This feature set will be used by the classification algorithm to detect weeds. After combining the two feature categories into one matrix, the resultant feature set is rescaled. It would allow the feature descriptors to fall in a specific range. If the features are not rescaled, then certain features would have extremely high values as compared to others and hence, cause biases in generating results. This task is done by using the Min-Max scalar. Its mathematical formula is given in (8):

$$X_{norm} = \left. X - X_{min} \right/ X_{max} - X_{min} \quad (8)$$

This function scales and transforms all the features one by one in a particular range. It utilizes a maximum and a minimum number to rescale data where X_{min} and X_{max} shows the minimum and maximum number, respectively. The resultant feature set is normalized to a particular range i.e. 0-1. In addition to feature normalization, rescaling tends to decrease standard deviation (deviation of values from the mean point). Therefore, it helps in restraining the consequences of outliers in the data.

Splitting the Dataset: The next step is to split the training dataset into two parts test data and train data. The train data will be utilized to train the proposed system or classification algorithm. On the other hand, test data will be used to test the working of the algorithm with the help of a validation technique. In the proposed work, the size of test data is equal to 0.10. This means that only 10% of the data is involved in testing while the rest is used for training purposes.

Training the Classification Algorithm: The next step is to train the classification algorithm based on the extracted features. It refers to algorithm learning by using training data to find relations, make decisions, and then develop understanding. That is why the training data needs to have all the necessary information about various classes involved in the data. In this study, Random Forest (RF) is used as a classification algorithm. The algorithm consists of a forest of multiple decision trees. Each one of these trees predicts a class. The class that gets the maximum number of votes is chosen by the algorithm as the final prediction. The algorithm works by generating random feature subsets which are used as a basis to split nodes of the decision tree. The reason for choosing this algorithm is that it overcomes the limitations posed by decision trees by decreasing the variance of a single tree and considering majority voting. Figure 7 shows a visual representation of the Random Forest algorithm. P(c|f) represents the maximum number of outputs for a particular

class. A total number of 100 'n' decision trees were involved in the RF algorithm.

The relative importance of features is computed with the help of the "Gini" index which is given in the equation below. Random Forest is calculating the importance of each feature by computing the reduction in the "node impurity" or "Gini impurity". If the observations are from a single class, then the Gini impurity will have a smaller value showing that the respective information is important. Equation (9) represents the mathematical formula to compute Gini impurity.

$$G_n = 1 - \sum_c [p(c|n)]^2 \quad (9)$$

where G_n is the Gini impurity, $p(c|n)$ represents the relative importance or frequency of a particular class c at the concerned node n. It is weighted by the probability to reach that node which is given in (10). High probability shows that the feature provides good information for classification. Using these values, the node importance is calculated as in (11).

$$P_n = \frac{no.of\ data\ samples\ at\ node\ n}{overall\ no.of\ samples} \quad (10)$$

$$n_j = w_j G_j - w_{left(j)} G_{left(j)} - w_{right(j)} G_{right(j)} \quad (11)$$

where 'n' represents the node, n_j shows the importance of the node j, n_j is the weighted samples that reach j, and G_j signifies the impurity value. As this equation is for a binary tree so, $left(j)$ indicates the left child node and $right(j)$ is the right child node. From here, we can calculate the feature importance for a specific decision tree as shown in (12).

$$f_i = {\sum j{:}\ node\ j\ splits\ on\ feature\ i^{n_j}}\Big/{\sum n_k} \quad (12)$$

where, f_i shows the importance of the feature 'i', n_j signifies the node j's importance and k features all the nodes. The feature importance is normalized using the equation and the resultant normalized value is used to find the average as represented in equation (13).

$$nf_i = {f_i}\Big/{\sum f_{all}} \quad (13)$$

where nf_i indicates the normalized 'i' feature importance and f_{all} represents all features. The feature importance values generated from each tree are summed and normalized using equation (14).

$$Rf_i = {\sum nf_{it}}\Big/{N} \quad (14)$$

where $\sum nf_{it}$ is the summation of the normalized value of feature importance nf_i in decision tree "t" and N indicates the total no. of decision trees i.e. 100.

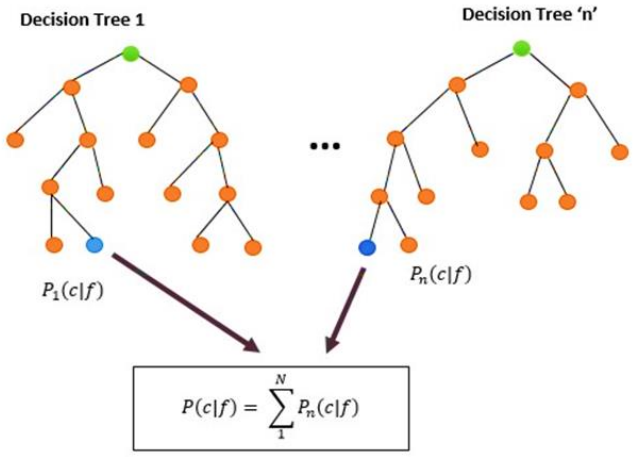

Figure 7. Visual representation of Random Forest with n decision trees

RESULTS AND DISCUSSION

Training Results: The model is trained using Random Forest using a feature set created by vertically combining all the extracted feature descriptors in a single array of size (800, 528). In this array, 800 corresponds to the total number of training images and 528 represents the number of feature descriptors extracted during the training process. Among these 528 feature descriptors, 512 features belong to the HSV color histogram while the rest of the 16 features correspond to the GLCM features including contrast, dissimilarity, correlation, and Angular Second Moment (ASM) of the Gray Level Co-occurrence Matrix. All the values in this feature set are scaled and normalized to a fixed range of 0-1 thus avoiding wrong and unbiased results. The data or feature descriptors are stored in an HDF (Hierarchical Data Format) file. The reason for choosing this file format is that it supports complex and large heterogeneous data to be stored in a compressed file. However, one cannot simply read a file written in HDF format. So, to analyze the training results, the HDF file is retrieved and observed in the form of an np-array (Numpy array). The training data (feature descriptors) and labels are stored in separate HDF files. The array of training labels is of size (800, 1) where each label corresponds to one of the 800 training data images.

Evaluation and Validation Results

K-fold Cross Validation: After training, the next step is to evaluate the ML model proposed in this study by using the k-fold cross-validation technique. To split the dataset, the size of the k-parameter is set to 10. This means that the data sample will be divided into 10 randomly generated equal groups. One of these groups corresponds to the test dataset

Table 1. Accuracies computed k-fold cross validation with k=10

Itr-1	Itr-2	Itr-3	Itr-4	Itr-5	Itr-6	Itr-7	Itr-8	Itr-9	Itr-10
0.9375	0.95	0.875	0.9375	0.925	0.8375	0.9375	0.9125	0.9625	0.9125

and the remaining k-1 groups will be assigned to the sample training dataset to fit the model. The results of the evaluation are stored while the sample datasets generated for k-fold cross-validation is reassigned to the training and test groups in the next iteration.

Since 90% of the images are assigned to the training set so the training set contains a total of 720 image which is 90% of 800 images. Similarly, the remaining 10% of the 800 images i.e. 80 images are allocated to the test dataset. The whole process will be repeated 10 times. The model is evaluated by computing its accuracy through the cross-validation technique. Table 1 represents the accuracies achieved after 10-iterations of the cross-validation technique. The final accuracy computed by the k-fold cross-validation technique is calculated by taking an average of the ones mentioned in Table 1 which is 0.918 or 92%. Mathematically, it is computed as given in the equation (15) where Where n is the total number of samples, y represents true values of the global datatset and $\hat{y}$ is the value predicted by the model.

$$Acc(y,\hat{y}) = \sum_{i=0}^{n-1} 1(\hat{y}_i = y_i) \Big/ n \quad (15)$$

Regression Metrics: To further evaluate the performance of the Random Forest algorithm, we have utilized the metrics including Mean Squared Error, Mean Absolute Error and Root Mean Squared Error. These metrics are used specifically when the algorithm deals with regression problems. As the proposed work is all about designing a model to learn various characteristics and patterns and then make accurate predictions, therefore we have used the following metrics for quantitative assessment of the proposed model.

Mean Absolute Error (MAE) signifies the errors between the actual values of the test dataset and the values predicted after fitting the algorithm on the test dataset. The mathematical formula to compute its value is given in (16).

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} |y_i - \hat{y}_i| \quad (16)$$

Where n represents the total number of samples, y is the true value of the test dataset whereas $\hat{y}$ is the value predicted by the model.

Mean Squared Error (MSE) represents the risk involved in predicting some value. In other words, it gives the expected value of the squared loss or error. Its equation is given in (17).

$$MSE\ (y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \quad (17)$$

Root Mean Squared Error (RMSE) measures how far the predicted values are from the best fit line. In other words, RMSE calculates their standard deviation. The mathematical equation featuring RMSE is given in (18).

$$RMSE = \sqrt{\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \Big/ n} \quad (18)$$

The respective values for all the above-mentioned regression metrics are given in Table 2. If an algorithm performs well then these metrics are expected to have low values. It can be observed from the table that the computed values for MAE, MSE, RMSE are close to zero which indicates a better fitting algorithm.

Table 2. Results of Regression Metrics (MAE, MSE, RMSE) for model evaluation

Mean Absolute Error	0.1125
Mean Squared Error	0.1375
Root Mean Squared Error	0.3708

Confusion Matrix: The performance of the proposed model is also evaluated with the help of a confusion matrix. For this purpose, the test dataset, obtained after splitting the original dataset into train and test set, is utilized. At first, the model is fitted on the training dataset and then it is used to predict the classes of the test dataset. The confusion matrix is generated using the predicted results and the true values of the test dataset. Figure 8 shows the multiclass confusion matrix generated in the proposed work featuring all four classes including broadleaf, soil, grass, and soybean.

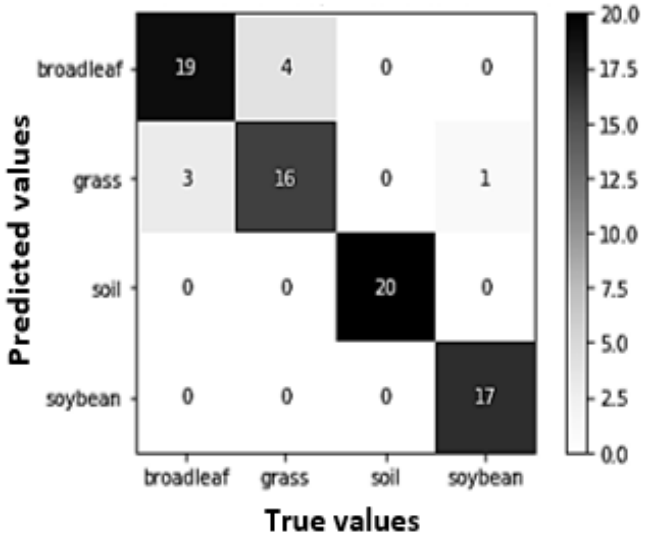

Figure 8. Confusion matrix featuring True and Predicted labels for soil, grass, broadleaf and soybean

Classification Report: The classification report illustrates the recall, precision, support, and F1 scores of the concerned model. These are the prime classification metrics that are measured based on each class separately. Unlike the k-fold cross validation that comprises a global accuracy, these classification metrics do not mask the functional weakness involved in each class and thus allow a deeper insight into the performance of the classifier. The precision, recall, F1 and support scores belong to each class are given in Table 3. The

classification metrics are explained and calculated in terms of the number of true positives, true negatives, false positives, and false negatives. Among them, precision specifies the exactness of the classifier, recall refers to the completeness of the classifier or the percentage of all correctly identified positives, f1 scores represents the weighted (harmonic) mean of precision and recall, and support here measures the occurrences of a particular class in a dataset. The mathematical representations of the respective classification metrics are given in equations (19), (20), and (21) where $prec$ refers to precision, TP, FP, FN indicates true positive, false positive, false negative respectively.

$$prec = {TP}/{TP + FP} \quad (19)$$

$$recall = {TP}/{TP + FN} \quad (20)$$

$$f1 = 2 \times {prec * recall}/{prec + recall} \quad (21)$$

Table 3. Results of Classification report representing precision, recall, F1 and support values of all classes

	Precision	Recall	F1	Support
Broadleaf	0.86	0.83	0.84	23
Grass	0.80	0.80	0.80	20
Soil	1.00	1.00	1.00	20
Soybean	0.94	1.00	0.97	17

It can be seen from Table 3 that the model gives the best results with soil images, followed by soybean, broadleaf and grass images. The highest precision or sensitivity is achieved with soil followed by broadleaf images i.e. 1.00 and 0.94 respectively. The model achieved an f1 score of 1.00 with soil images, 0.97 with soybean images, 0.84 and 0.80 with broadleaf and grass images. The overall classification rate of the model in the test dataset is 0.9% or 90% which is quite good. On calculating the number of TP, TN, FP, FN from the confusion matrix in Figure 8, we observe that the number of true positives for broadleaf weed images is 19, the number of true negatives is 54, and that of a false positive and false negative is 4 and 3 respectively from a total of 80 images. This means that only 3 out of 80 images, belonged to other classes, are classified as a weed. The correct classification and miss-classification rate of each class are given in Table 4. The complete program took only 79.9 seconds to execute which makes it ideal for a real-time environment.

Table 4. Results showing correct classification and miss-classification rate of test dataset broadleaf, grass, soil and soybean

	Correct classification	Miss-classification
Broadleaf weed	0.9125 = 91%	0.0875 = 8.75%
Grass	0.9000 = 90%	0.1000 = 10%
Soil	1.0000 = 100%	0.0000 = 0%
Soybean	0.9875 = 99%	0.0125 = 1%

The trained model is also used to predict unseen images from the dataset. These images are not included in the training and test dataset. Rather these are utilized solely with the purpose to test the validity of the model. A total number of 32 images were stored in a separate location. Each class comprised 8 different images captured under varying light conditions. Some of these images are also affected by noise. These images are passed on to the trained model to predict their respective classes. Figure 9 represents four per class images predicted by the model. It can be seen from the figure that the model is successfully able to classify unseen images. Out of 32 images, 30 images are predicted correctly. The two miss-classified images belong to the grass category. All the weed, soil and soybean plant images are labelled correctly.

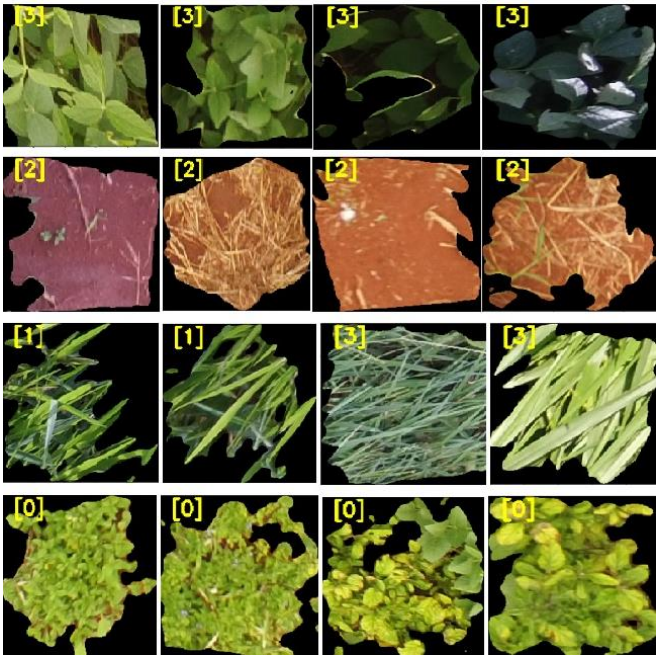

Figure 9. Prediction results on unseen images. Label 1= Soybean, label 2 = Soil, label 3 = Grass, label 4 = Broadleaf weed

Conclusion: The traditional practices of weed management are costly and time-consuming. Automatic weed detection poses a solution to the prevalent problems including wastage of herbicide, high labor costs and poor fruit quality due to weed growth. The problem with the existing weed detection systems is that they lack robustness. Also, most of these systems require large datasets to train the model which would eventually result in an increased computation time. So, these problems need to be resolved. The proposed model aims to solve these problems by reducing average computation time and giving accurate results.

The model works by loading the image dataset and applying various pre-processing steps including image rescaling, grey-scale conversion, and noise-reduction. These pre-processing steps are followed by feature detection and extraction. Two kinds of feature categories are extracted during this process. Texture features are extracted by calculating contrast, correlation, dissimilarity and using a Gray Level Co-occurrence Matrix. Color features are extracted by first converting all images to HSV color space and then computing normalized histograms. The next step is to normalize and combine all feature descriptors into a single matrix. The resultant feature set is used to train the Random Forest Model. The performance of the model is evaluated using K-fold cross-validation, Regression Metric and by computing precision, recall and F1 score of all four classes. At last, the model successfully predicts classes of unseen soil, grass, soybean and broadleaf images.

REFERENCES

Abdalla, H.B., G. Li, J. Lin and M. Alazeez. 2016. NoSQL Injection: Data Security on Web Vulnerability. Int. J. Secur. Its Appl. 10:55–64.

Abdalla, H.B., A.M. Ahmed and M.A.A. Sibahee. 2020. Optimization Driven MapReduce Framework for Indexing and Retrieval of Big Data. KSII Trans. Internet Inf. Syst. 14:1886–1908.

Andújar, D., H. Moreno, J.M.B. Guevara, D. Castro and A. Ribeiro. 2019. Aerial imagery or on-ground detection? An economic analysis for vineyard crops. Comput. Electron. Agric. 157: 351–358.

Azadbakht, M. and T. Mana. 2003. Qualitative and quantitative determination of pyrrolizidine alkaloids of wheat and flour contaminated with senecio in Mazandaran province farms. Iran. J. Pharm. Res. 2:179–183.

Chlingaryan, A., S. Sukkarieh and B. Whelan. 2018. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. Comput. Electron. Agric. 151: 61–69.

Dyrmann, M., H. Karstoft and H.S. Midtiby. 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151:72-80.

Gao, J., D. Nuyttens, P. Lootens, Y. He and J.G. Pieters. 2018. Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery. Biosyst. Eng. 170:39-50.

Ishak, A.J., M.M. Mustafa, M.M. Tahir and A. Hussain. 2008. Weed detection system using support vector machine. International Symposium on Information Theory and Its Applications, Auckland, New Zealand. pp.1–4.

Khan, K., F. Khan, S. Ahmad and K.B. Marwat. 2020. Palynological investigation of allergenic and invasive weeds plants for biodiversity in district Lakki Marwat using scanning electron microscopy. Pak. J. Weed Sci. Res. 26.

Marshall, G., C. Fyfe, M. Coleman, B. Sindel and P. Kristiansen. 2019. Economics of weed management in the Australian vegetable industry. University of New England Press, England.

Martinez, D.A., U.E. Loening and M.C. Graham. 2018. Impacts of glyphosate-based herbicides on disease resistance and health of crops: a review. Environ. Sci. Eur. 30.

Masuda, R., K. Nakayama and K. Nomura. 2010. Rice plant detection in heading term for autonomous robot navigation. XVIIth World Congress of the International Commission of Agricultural and Biosystems Engineering (CIGR), Québec City, Canada. pp.1–8.

Metwally, I.M.E and M.A.E. Wakeel. 2019. Comparison of safe weed control methods with chemical herbicide in potato field. Bull. Natl. Res. Cent. 43:1–7.

Partel, V., S.C. Kakarla and Y. Ampatzidis. 2019. Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. Comput. Electron. Agric. 157:339–350.

Peccia, F. 2018. Weed detection in soybean crops. https://kaggle.com/fpeccia/weed-detection-in-soybean-crops (accessed 8.26.20).

Pulido, C., L. Solaque and N. Velasco. 2017. Weed recognition by SVM texture feature classification in outdoor vegetable crops images. Ing. E Investig. 37:68–74.

Slaughter, D.C., D.K. Giles and D. Downey. 2008. Autonomous robotic weed control systems: A review. Comput. Electron. Agric., Emerging Technologies for Real-time and Integrated Agriculture Decisions. 61:63-78.

Talaviya, T., D. Shah, N. Patel, H. Yagnik and M. Shah. 2020. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. Artif. Intell. Agric. 4:58-73.

Wang, A., W. Zhang and X. Wei. 2019. A review on weed detection using ground-based machine vision and image processing techniques. Comput. Electron. Agric. 158:226–240.

[Received 18 Apr 2020; Accepted 01 Jun. 2021; Published (online) 25 Jun 2021]