International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

Advancing Weed Management: A Novel Image Classification System Using NCA-based Feature Selection and PSO-GSA Optimized Random Forest Classifier

Dr. Rahul Gedam
Assistant Professor and HOD, Department of Electronics and Telecommunication, Chouksey Engineering College, Bilaspur CG
Email ID: engg.rahul2801@gmail.com

Dr. Shubhlakshmi Tiwari,
Associate Professor and HOD, Department of Civil Engineering, Chouksey Engineering College, Bilaspur CG
Email ID: hodcivil@cecbilaspur.ac.in

Abstract –Adopting more intelligent and targeted weed management strategies enables farmers to reduce their dependence on chemical treatments, save valuable time and resources, and improve crop yields, all while lessening negative effects on the environment and human health. To address these urgent issues, it is essential to pursue a more sophisticated agricultural approach that incorporates automated technologies and leverages machine learning algorithms. This study presents an innovative weed detection system aimed at differentiating crops from weeds by integrating various feature extraction techniques with advanced machine learning capabilities. At the core of this system is the application of hybrid features, coupled with an effective feature selection method based on neighborhood component analysis. These distinctive features are utilized by a particle swarm optimization and gravitational search algorithm (PSO-GSA) optimized random forest classifier, which effectively categorizes images into either crops or weeds. The results demonstrate that our method, which combines hybrid feature extraction with the PSO-GSA-RF classification approach, significantly outperforms other techniques. Furthermore, this system can be integrated into agricultural robots, allowing for the precise application of herbicides only where necessary, thereby minimizing the introduction of harmful chemicals into the food supply and reducing the risk of human exposure to dangerous substances.

Keywords – Gravitational Search Algorithm, Neighborhood Component Analysis, Particle Swarm Optimi-zation, Random Forest, Weed

I. INTRODUCTION

Weeds pose a significant challenge in agriculture, as they can adversely affect crop production and yield. Competing with crops for essential resources such as nutrients, water, and sunlight, weeds also act as hosts for pests and diseases that can further damage crops. Effective weed management is crucial for sustaining crop productivity and ensuring food security. Proper weed control not only helps in reducing yield losses but also improves crop quality, while reducing the reliance on costly and environmentally harmful herbicides.

Various weed management techniques are available, including cultural, mechanical, and chemical methods. Cultural methods focus on modifying the environment or adjusting crop management practices to inhibit weed growth and competition. Mechanical methods involve physically removing or destroying weeds through practices like tilling, mowing, or manual weeding. Chemical methods utilize herbicides to suppress or eliminate weeds [1].

Nevertheless, a well-planned and carefully implemented weed management strategy is essential to avoid negative outcomes, such as the development of herbicide-resistant weed species or environmental degradation. An integrated approach that combines multiple control methods is often the most effective and sustainable solution for managing weeds [2].

Automation and digitization present significant opportunities for enhancing weed detection and management in agriculture.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
***Article Received:** 25 March 2021 **Revised:** 12 April 2021 **Accepted:** 30 May 2021*

Technologies such as IoT, image processing, and data-driven techniques enable real-time weed monitoring and can assist farmers in making more informed decisions regarding crop management [3].

Despite these advancements, the task of weed detection and identification remains challenging and requires sophisticated computer vision algorithms. Traditional image processing techniques may lack the accuracy required to distinguish between various weed species in a complex agricultural setting. As a result, advanced methods like deep learning and neural networks are being developed to enhance the precision of weed detection. Leveraging these technologies allows for more efficient weed management, ultimately boosting crop productivity and sustainability.

For weed detection in specific agricultural contexts, computer vision systems that utilize a combination of conventional image processing techniques and deep learning algorithms can be employed [4]. Classical image processing methods rely on a variety of image analysis tools to extract significant features from visual data, such as plant shape, color, texture, and size. Classification algorithms can then use these features to differentiate between different weed species and other plants, contributing to the development of more accurate and reliable weed detection systems tailored to specific domains.

This research introduces a novel method for weed identification in row crops, combining geometric and spectral data through automated image processing. The approach assumes that any plant growing between crop rows is a weed, and weeds within crop rows share spectral properties with those between rows. This technique, tested using multispectral images captured by a sensor mounted on a 3-meter pole, is capable of detecting weeds in both inter-row and intra-row areas.

The first key contribution of this study is the development of a classification method that integrates spatial and spectral information while automatically building the training dataset for a supervised classifier, eliminating the need for manual pixel labeling of crops, weeds, or soil. The second contribution is an in-depth analysis of the impact of using either spatial data, spectral data, or both on the classification accuracy for images taken in maize and sugar beet fields.

By combining spatial and spectral information and utilizing spatial data to construct training datasets, this method offers a promising tool for weed detection in row crops. Additionally, the study provides valuable insights into the significance of spatial and spectral data for achieving high-quality classification in weed detection tasks.

II. LITERATURE REVIEW

The application of machine vision technology in developing sustainable and integrated weed management systems holds great promise. By utilizing imaging sensors and sophisticated computer algorithms, these systems can effectively differentiate between crops and weeds, identify specific weed species, and provide targeted weed control measures. This level of precision is instrumental in minimizing the use of herbicides, which is vital for mitigating the problem of herbicide resistance in weeds.

The rise of machine vision-based robotic systems for in-season weed management in crops like cotton is gaining traction. Once weeds are identified and located, various control methods such as targeted herbicide spraying, mechanical cultivation, or thermal techniques can be applied. The use of such technologies in weed management offers the potential for more sustainable farming practices, with reduced herbicide dependency and enhanced crop productivity [5].

Significant research has been conducted on image processing and analysis techniques for weed detection. Various color indices that highlight plant greenness have been proposed to improve weed segmentation and separation from soil backgrounds. However, these methods are often unreliable under varying natural light conditions. To overcome this, deep learning techniques, particularly convolutional neural networks (CNNs), have been adopted, providing improved weed detection accuracy and more reliable classification results [6]. Key features for accurate weed identification include morphology, texture, and spectral reflectance. A real-time computer vision-based weed detection framework was proposed by authors in [7], employing grayscale segmentation for weed classification from a contextual perspective. In this study, weeds were categorized into broad, narrow, and small-leaf classes using thresholding and sample variance calculations. In [8], improved weed classification performance was achieved by utilizing enhanced visual features such as local shape and texture. Their segmentation algorithm (AdaBoost combined with Naïve Bayes) successfully adapted to different lighting conditions. Numerous weed control strategies aimed at reducing herbicide use and boosting crop yield have been proposed [9-12]. In [13], a technique for detecting and classifying citrus diseases was introduced, using a color difference algorithm to isolate diseased areas and employing color histograms and textural features for classification, achieving high accuracy.

Weed detection can be enhanced by integrating both spatial and spectral data. In a study by [14], combining spatial and spectral information led to improved crop-weed

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
***Article Received:** 25 March 2021 **Revised:** 12 April 2021 **Accepted:** 30 May 2021*

differentiation, yielding promising visual results. For invasive weed species in natural environments, the authors of [15] presented a feature-learning technique based on color, edge, and texture information. This approach was expanded in [16] to include spectral, shape, spatial, and texture data. Object-based image analysis (OBIA) was utilized by [17] to classify linear vegetation objects that were difficult to identify using spatial information alone. Their classification was based on the degree of spectral similarity, such as NDVI or ExG values. In [18], a novel approach to weed classification was introduced using OBIA along with a Support Vector Machine (SVM) classifier. This method integrated additional spectral information, shape features, and vegetation indices, which the classifier used to enhance classification results. The study achieved 96% accuracy in sunflower fields, using 90 manually labeled items per class (soil, weeds, and crop), surpassing older pixel-based methods that relied solely on spectral data. This technique successfully identified weeds both between and within crop rows, demonstrating its versatility.

A diverse and extensive dataset of high-quality images is crucial for training accurate and reliable weed recognition models. However, collecting such datasets presents challenges, especially in agricultural contexts where environmental variables can significantly alter the appearance of weeds. Moreover, the process of manually labeling large datasets is often labor-intensive and costly.

Additionally, the performance of weed recognition algorithms plays a significant role in the overall effectiveness of these systems. While traditional computer vision methods—such as image processing, segmentation, and feature extraction—can be effective, they often face limitations when dealing with variations in weed appearance and environmental factors. More advanced machine learning approaches, particularly CNNs, show greater potential for improving weed recognition performance but require substantial amounts of labeled data and can be computationally intensive.

III. Materials and Method

3.1 Dataset

The food crop species chosen for the dataset were selected based on their popularity among consumers in Latvia and the necessity for intensive weed management techniques. The dataset contains two types of images: (i) images of crops and weeds grown in controlled greenhouse conditions, and (ii) images of crops and weeds captured in open field settings. The greenhouse images were taken at the Scientific Institute for Plant Protection Research "Agrihorts" at the University of Life Sciences and Technologies of Latvia in Jelgava, Latvia. Field condition images were captured from three distinct locations in Latvia: Kekava, Rujiena, and Krimulda. All images were taken using a perspective projection over the plants [19].

To construct the dataset, six food crops and eight common weed species were grown in vegetation pots within a greenhouse environment, where conditions were controlled. The plants were cultivated in peat-based substrate, and the seeds were sown in one to two rows with a spacing of 2.0–5.0 cm. Watering occurred once or twice per week, and temperatures were maintained at +20°C during the day and +15°C at night, with humidity levels kept below 50%. LED lamps were used in conjunction with natural sunlight, providing illumination from 6:00 a.m. to 8:00 p.m. The dataset was compiled using a variety of cameras, including Canon EOS 800D, Sony W800 digital cameras, and Intel RealSense D435 cameras.

The dataset comprises 1,118 JPEG images, accompanied by 7,853 XML files with manual annotations. The images are in bitmap color format, using RGB with three samples per point, and are primarily designed to facilitate the identification of food crops and weeds in digital images. The images vary in resolution and size, with common dimensions including 720×1280×3, 1000×750×3, 640×480×3, 640×360×3, and 480×384×3. The dataset is categorized into two main classes: food crops (411 annotations) and weeds (7,442 annotations). Each annotation provides detailed information about the location and size of objects in the image, aiding the development of computer vision algorithms for weed detection and classification.

The diversity of the camera equipment used and the variability in image resolutions and sizes accurately represent real-world scenarios where crops and weeds are encountered. This ensures the dataset is suitable for training and evaluating machine learning models for crop management and weed detection.

The weed species selected for the dataset include: goosefoot (Chenopodium album), catchweed (Galium aparine), field pennycress (Thlaspi arvense), shepherd's purse (Capsella bursa-pastoris), field chamomile (Matricaria perforata), wild buckwheat (Polygonum convolvulus), field pansy (Viola arvensis), and quickweed (Galinsoga parviflora).

The six chosen food crops are: beetroot (Beta vulgaris), carrot (Daucus carota var. sativus), zucchini (Cucurbita pepo subsp. pepo), pumpkin (Cucurbita pepo), radish (Raphanus sativus var. sativus), and black radish (Raphanus sativus var. niger).

This dataset is versatile and can be used for various research applications, including weed-crop interactions, plant

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

development, weed management techniques, and the influence of environmental conditions on plant growth.

3.2 GLCM

The Gray Level Co-occurrence Matrix (GLCM) is a statistical approach used to extract texture characteristics from images, which is particularly relevant in weed detection. Initially developed as a method for texture analysis, the GLCM captures the joint probability distribution of two pixel intensities at a specific spatial distance and direction. Each element in the matrix indicates how often a pair of gray-level values occurs at a given relative position within the image. The matrix is symmetrical, meaning that the probability of a value pair at position (i, j) is equivalent to that at (j, i).

Constructing a GLCM involves four key steps. First, the image undergoes preprocessing to eliminate any noise or artifacts that might distort the calculation of the matrix. Next, the image is quantized into a discrete set of gray-levels, typically using 8 or 16 levels. Then, a distance (d) and direction (θ) are selected to define the spatial relationship between pixel pairs. Common directions used are 0°, 45°, 90°, and 135°, across various distances. Finally, the co-occurrence matrix is computed by counting the occurrences of pixel pairs (i, j) at the specified distance and direction. The resulting matrix is then normalized by dividing each value by the total number of pairs at that distance and direction.

The GLCM can be utilized to derive several texture features, including contrast, energy, homogeneity, and entropy, which are computed as follows:

- Contrast: Measures the intensity difference between neighboring pixels. It is defined as:

$$Contrast = \sum_{i,j} (i-j)^2 P(i,j) \quad (1)$$

  Where $P(i, j)$ is the normalized co-occurrence matrix.

- Energy: Quantifies the uniformity of pixel values and is calculated as:

$$Energy = \sum_{i,j} P(i,j)^2 \quad (2)$$

- Homogeneity: Reflects how close the distribution of pixel values is to the diagonal of the GLCM. It is defined as:

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{(1+|i-j|)} \quad (3)$$

- Entropy: Measures the randomness or unpredictability of the pixel intensity distribution. It is given by:

$$Entropy = -\sum_{i,j} P(i,j) \log_2\big(P(i,j)\big) \quad (4)$$

GLCM provides a straightforward yet effective means of extracting texture features from images and is widely applied in fields such as medical imaging, remote sensing, and computer vision.

3.3 Shi-Tomasi Corner Detector

Corner detection in images is often achieved using the Shi-Tomasi method, a variant of the Harris corner detector. This method evaluates the eigenvalues of the image's second moment matrix to identify corners. The Shi-Tomasi corner detector calculates a numerical score for each pixel by analyzing the eigenvalues of a matrix M, which is computed as:

Let $I(x, y)$ be the intensity of a pixel at location (x, y) in the image. The corner detection method known as Shi-Tomasi calculates a numerical value for every pixel (x, y) by analyzing the eigenvalues of the matrix M. This matrix is specifically defined as follows:

$$M = \sum w(x,y)[\nabla_I(x,y)\nabla_I(x,y)^T] \quad (5)$$

Where $\nabla_I(x, y)$ is the gradient of the image at (x, y) and $w(x, y)$ is a window function that assigns weights to pixels in the neighborhood of (x, y).

The eigenvalues of the matrix M are given by:

$$\lambda_1, \lambda_2 = \frac{1}{2}\left[trace(M) \pm \sqrt{\big(trace(M)^2 - 4 * det(M)\big)}\right] \quad (6)$$

Where $trace(M) = \lambda_1 + \lambda_2$ is the sum of the eigenvalues and $det(M) = \lambda_1\lambda_2$ is the determinant of the matrix.

The Shi-Tomasi corner detector computes a score for each pixel (x, y) based on the smaller of the two eigenvalues:

$$\mathbb{R} = \min(\lambda_1, \lambda_2) \quad (7)$$

This score helps in identifying the strong corner points in the image.

IV. Proposed Methodology

4.1 Image Pre-Processing

Contrast Limited Adaptive Histogram Equalization (CLAHE) is a sophisticated image processing technique designed to improve the contrast of an image. It serves as an enhancement over the conventional Histogram Equalization method.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

The primary principle of Histogram Equalization involves redistributing pixel intensities across an image to achieve a more uniform intensity distribution, thereby enhancing the overall contrast. However, this method can sometimes lead to drawbacks, such as the amplification of noise and the loss of important details within the image.

CLAHE addresses these limitations by applying contrast enhancement in a localized manner. It accomplishes this by segmenting the image into smaller regions known as tiles, performing histogram equalization on each tile independently. By doing so, CLAHE limits the contrast enhancement to local areas, which helps maintain the details in the image.

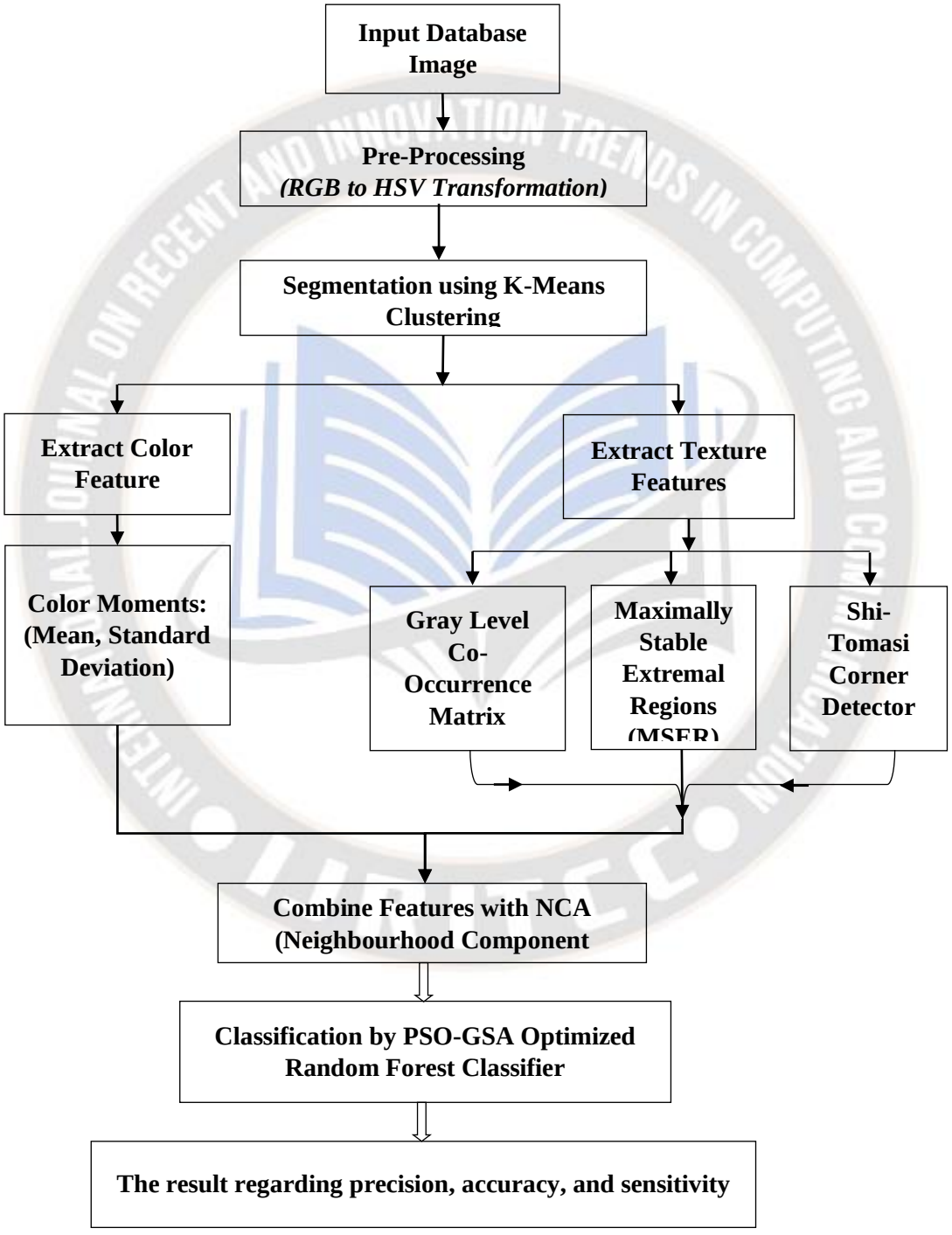

Figure 1: Flow diagram of proposed methodology

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

Furthermore, CLAHE introduces a parameter to limit the amount of contrast enhancement applied to each tile. This parameter ensures that the enhancement remains controlled, preventing excessive contrast adjustments that could lead to noise amplification.

In summary, CLAHE is an effective technique for improving image contrast, especially in cases where traditional histogram equalization does not yield satisfactory results. Its applications are widespread, including medical imaging, satellite imagery, and other domains where contrast enhancement is crucial.

4.2 Feature Extraction

Texture features represent specific characteristics of an image or signal that convey its visual texture. These features are typically obtained through various mathematical methods, including filtering and statistical analysis techniques.

Local features are used in computer vision and image processing to represent specific regions of an image that are distinct from their surroundings. They are often used for tasks such as object recognition, image matching, and tracking.

Local features are typically identified through a process called feature detection, which involves searching for patterns in the image that meet certain criteria, such as having high contrast or being locally unique. Once detected, these features are often described using feature descriptors, which provide a quantitative representation of the feature's properties, such as its location, orientation, and scale.

Local features include blobs, which are regions of an image that have a roughly circular shape and a uniform intensity; corners, which are points where the image gradient changes sharply in two or more directions; and edge pixels, which are points along the boundary between regions of differing intensity or texture.

Local features are important because they provide a way to represent complex visual patterns in a way that is both computationally efficient and robust to changes in image conditions, such as variations in lighting or viewpoint

4.2.1 Maximally Stable Extremal Regions (MSER)

MSER is a method for feature extraction in images that can be used for various applications including object recognition, detection, and tracking. In the context of weed image feature extraction, MSER can be used to extract distinctive features that can be used to distinguish between weeds and crops.

MSER works by detecting regions of the image that are stable with respect to intensity changes. These regions are called extremal regions, and they correspond to connected components of the level sets of the image. An extremal region is a region of the image that has the property that it remains connected for all threshold values of the image intensity.

The MSER algorithm starts by computing the intensity gradient of the image. The extremal regions are then obtained by thresholding the gradient at different levels and extracting connected components of the resulting binary image. The threshold level is increased gradually from 0 to 255, and at each level, the extremal regions are extracted and added to a list of candidate regions.

The final set of regions is obtained by selecting the regions that are stable over a range of threshold levels. The stability is measured by the relative size of the region at different threshold levels. The regions that remain relatively stable over a range of threshold levels are considered to be maximally stable.

The MSER algorithm can be expressed mathematically as follows:

1. Compute the intensity gradient of the image:

$$G(x,y) = \sqrt{(I_x^2(x,y) + I_y^2(x,y)} \tag{8}$$

Where I_x and I_y are the first-order derivatives of the image intensity with respect to x and y, respectively.

2. Threshold the gradient at different levels to obtain a binary image:

$$B(x,y,t) = \begin{cases} 1 & if\ G(x,y) \geq t \\ 0 & Otherwise \end{cases} \tag{9}$$

Where t is the threshold level.

3. Extract connected components of the binary image to obtain candidate extremal regions.
4. Compute the relative size of each candidate extremal region at different threshold levels:

$$r(B,i,j) = \left| \frac{B(i,j,t)}{B(i,j,t-1)} \right| \tag{10}$$

Where i and j are the coordinates of the pixel, t is the current threshold level, and $t-1$ is the previous threshold level.

5. Select the regions that are maximally stable over a range of threshold levels:

$$M(B) = \{ R \mid R\ is\ a\ connected\ component\ of\ B\ and\ r(B,i,j) \leq c\ \forall\ t \} \tag{11}$$

Where c is a constant that controls the stability threshold, and R is a maximally stable extremal region.

In the context of weed image feature extraction, the MSER algorithm can be used to extract distinctive regions that correspond to the leaves or other parts of the weed. These

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

regions can then be used to distinguish between weeds and crops in the image. The MSER algorithm has been shown to be effective for weed detection in various crop fields, and it has the advantage of being computationally efficient and robust to variations in lighting and image noise.

4.2.2 Shi-Tomasi Corner Detector

The score $\mathbb{R}$ measures the corner response at the pixel (x, y) and is used to determine whether the pixel is a corner or not. A high score indicates that the pixel is a corner, while a low score indicates that the pixel is not a corner.

To extract features from an image using the Shi-Tomasi corner detector, the following steps can be performed:

1. Compute the gradient of the image using a derivative filter.
2. Compute the matrix M for each pixel using the gradient information and a window function.
3. Compute the eigenvalues of the matrix M for each pixel.
4. Compute the corner response $\mathbb{R}$ for each pixel based on the eigenvalues.
5. Apply a threshold to the corner response to determine which pixels are corners.
6. Extract the corner locations and use them as features for further processing.

In the context of weed image feature extraction using the Shi-Tomasi corner detector, the extracted corner features can be utilized to identify and classify several types of weeds established on their unique corner structures. The corner features can also be used for object detection and tracking, as well as for image registration and stitching in applications such as precision agriculture.

4.3 Feature Selection

Neighborhood Component Analysis (NCA) is a distance-based feature selection algorithm that can be applied to weed images to select the most informative features for classification tasks. The goal of NCA is to learn a linear transformation of the input features that maximizes the classification accuracy on a training set. The transformation is learned in such a way that the features that are most relevant for the weed classification task are emphasized, while the irrelevant features are suppressed.

The mathematical formulation for NCA for feature selection for weed images is as follows:

Let X be an $n \times d$ matrix of input features, where n is the number of weed images and d is the number of features.

Let y be an $n \times 1$ vector of class labels, where each y_i is an integer indicating the class of the i^{th} weed image.

Let w be a $d \times k$ matrix of weights, where k is the desired number of selected features.

The objective of NCA is to maximize the leave-one-out classification accuracy on the training set, which is defined as:

$$J(w) = \sum_i p_i(y_i|x_i, X\backslash i, w) \tag{12}$$

Where $p_i(y_i|x_i, X\backslash i, w)$ is the probability that the i^{th} weed image is correctly classified when the i^{th} feature is removed from X, and $X\backslash i$ is the matrix X with the i^{th} row removed. The probability $p_i(y_i|x_i, X\backslash i, w)$ is computed using the softmax function:

$$p_i(y_i|x_i, X\backslash i, w) = \frac{exp(w_i^T x_i)}{xp(w_j^T x_i)} \tag{13}$$

where w_i is the i^{th} column of the matrix w, and x_i is the i^{th} row of the matrix X.

The input features for weed images can include color, texture, shape, and other characteristics. The exact features used will depend on the specific classification task and the available data. For example, color features can be extracted using color histograms, texture features can be extracted using Gabor filters or local binary patterns, and shape features can be extracted using morphological operations or geometric descriptors.

The optimization problem can be solved using gradient descent or other optimization methods. The derivative of the objective function in relation to the weights w can be expressed as:

$$\nabla_w J(w) = \sum_i p_{i(y_i|x_i, X\backslash i, w)} - \sum_j p_i(y_i|x_i, X\backslash i, w) p_j(y_j|x_j, X \backslash i, w)) x_i x_i^T w \tag{14}$$

The weight matrix w is updated iteratively by taking small steps in the direction of the negative gradient, until convergence is reached.

Finally, the k features that have the highest absolute weights in the learned weight matrix w are selected as the most informative features for the weed classification task. These features can then be used as inputs to a classification algorithm to classify new weed images.

4.4 Classification

The machine learning algorithm called Random Forest is commonly used for carrying out classification and regression tasks. It functions by constructing numerous decision trees during the training phase and then provides the class that represents the modal value of the classes (in the case of classification) or the mean prediction (in the case of regression) of the individual trees. Metaheuristics optimization algorithms

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

such as PSO and GSA are utilized to discover the optimal solution for a specified problem.

To optimize the accuracy of a Random Forest model using PSO-GSA optimization, the following procedures can be undertaken:

1. Define the objective function: The objective function is the measure utilized to assess the performance of the Random Forest model. In this instance, classification accuracy is utilized as the objective function.
2. Define the search space: The search space pertains to the range of values explored by the PSO-GSA algorithm to determine the optimal solution. In the case of a Random Forest model, the search space includes the number of trees, the depth of each tree, and the number of features used for each split.
3. Initialize the PSO-GSA algorithm: The PSO-GSA algorithm necessitates initializing the population size, the maximum number of iterations, and the initial positions of the particles.
4. Evaluate the fitness of the particles: The fitness of each particle is assessed by training a Random Forest model utilizing the particle's position within the search space and determining its accuracy using the objective function.
5. Update the position of the particles: The position of each particle is updated using the PSO-GSA algorithm, which considers the particle's best position, the swarm's best position, and the current position of the particle.
6. Repeat steps 4 and 5 until convergence: The PSO-GSA algorithm is repeated until either the maximum number of iterations is attained or a satisfactory accuracy is achieved.

The PSO-GSA algorithm is a hybrid swarm algorithm that combines the advantages of two existing swarm algorithms: PSO and GSA. The hybridization is achieved by using a low-level co-evolutionary heterogeneous hybrid, in which the two algorithms work together in parallel.

The main objective of the PSO-GSA algorithm is to add the social thinking ability ($gbest$) of PSO to the local search ability of GSA. This is accomplished through the following equation:

$$X(i,j) = X(i,j) + V(i,j) \tag{15}$$

Where $X(i,j)$ is the j^{th} dimension of the position vector of particle i, and $V(i,j)$ is the j^{th} dimension of the velocity vector of particle i. The velocity of each particle is updated using the following equation:

$$V(i,j) * wV(i,j) + c_1 r_1 * \big(pbest(i,j) - X(i,j)\big) + c_2 r_2 \big(gbest(j) - X(i,j)\big) + a * G(j) \tag{16}$$

Where w is the inertia weight, c_1 and c_2 are the cognitive and social acceleration coefficients, respectively. r_1 and r_2 are random numbers between 0 and 1. $pbest(i,j)$ is the best position of particle i up to the current iteration in the j^{th} dimension, and $gbest(j)$ is the best position of all particles up to the current iteration in the j^{th} dimension. $G(j)$ is the gravitational force in the j^{th} dimension, which is calculated using the following equation:

$$G(j) = \sum G(i,j) \tag{17}$$

Where $G(i,j)$ is the gravitational force between particle i and the other particles in the j^{th} dimension. The PSO-GSA algorithm uses the gravitational constant (G) to regulate the search behavior of the particles. The value of G is adjusted dynamically during the search process using the following equation:

$$G = G_0 e^{-at} \tag{18}$$

Where G_0 is the initial gravitational constant, t is the current iteration, and a is a constant parameter that controls the rate of decay of G.

In summary, the PSO-GSA algorithm combines the social thinking ability of PSO with the local search ability of GSA to improve the convergence rate and search accuracy of the algorithm. The algorithm has been applied to various optimization problems and has shown promising results.

Once the PSO-GSA optimization is complete, you can use the hyperparameters of the global best solution to train a random forest classifier on your data set and make predictions on new data.

V. Simulation and Results

5.1 K-Fold Forward Cross-Validation (K-Fold FCV)

This approach improves on the classic k-fold cross-validation for assessing model prediction capacity. Rather of splitting the dataset at random, all samples are first organised based on the material's property beliefs and then uniformly dispersed into k subsets.

The entire procedure is as follows:

1. Categorize all samples in ascending or descending order according to their property values.
2. Separate the categorized samples into k equal subsets labelled $S_1, S_2, \ldots, S_k$.
3. Designate S_2 as the validation set and S_1 as the training set, beginning with the second subset S_2. Train a model on S_1 and then test it on S_2.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

4. For the following cycle, make S_3 the validation set and all subsets before S_3 the training set. Train a model on S_1 and S_2 and then test it on S_3.
5. Step 4 should be repeated until all S_2 through S_k have been evaluated. Calculate the total performance of all models. The option to rank the samples ascending or descending is based on whether the model is predicted to extrapolate higher or lower.

When k is relatively big, the initial size of the training set may be insufficient. A minimum size for the training data might be defined to minimise distortions induced by this problem.

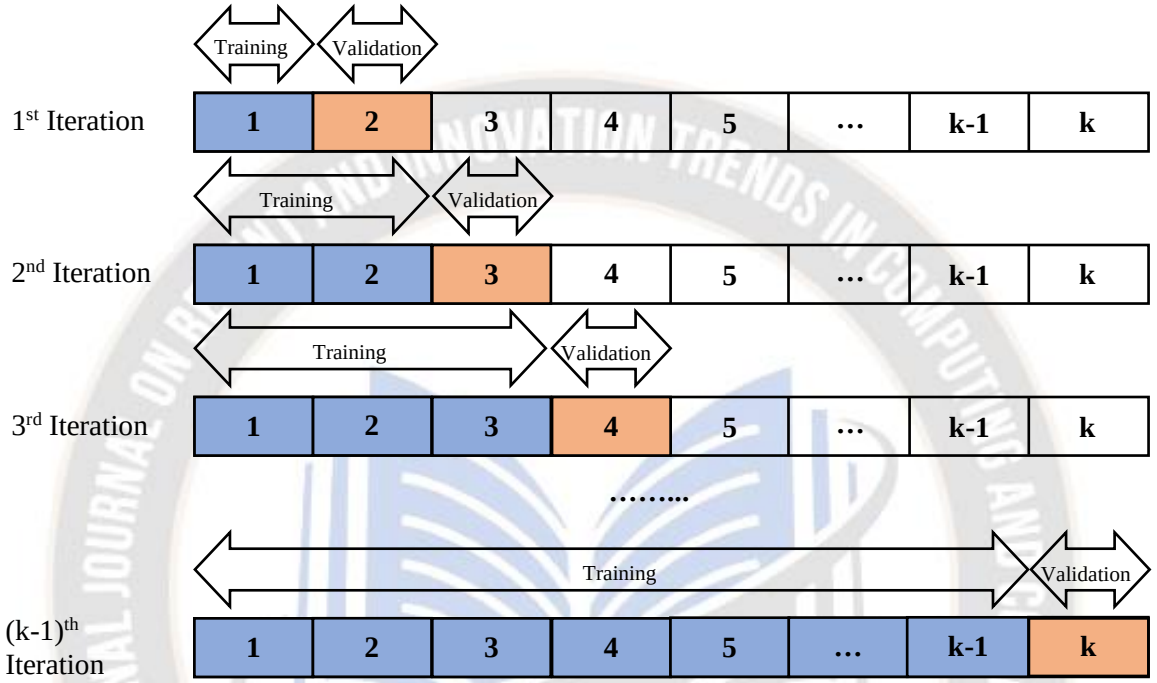

Fig.2. k-fold forward cross validation

5.2 Evaluation Parameters

Table 1. Evaluation parameters

TP (True Positive)	"Indicated the weed with that were classified as correctly classified"
TN (True Negative)	"Indicated the weed with that were classified as not classified correctly"
FP (False Positive)	"Indicated the weed with that were classified as incorrectly classified"
FN (False Negative)	"Indicated the weed with that were classified as not classified incorrectly"

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

$$Precision = \frac{TP}{TP + FP} \quad (20)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (21)$$

$$Specificity = \frac{TN}{TN + FN} \quad (22)$$

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \quad (23)$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN} \quad (24)$$

$$F - Score = \frac{2TP}{2TP + FP + FN} \quad (25)$$

$$Matthews\ Correlation\ Coefficient\ (MCC) = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}} \quad (26)$$

$$Kappa\ Statistics = \frac{2(TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TN + FN) \times (FN + TN)} \quad (27)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

5.3 Results

Fig. 3. Input image

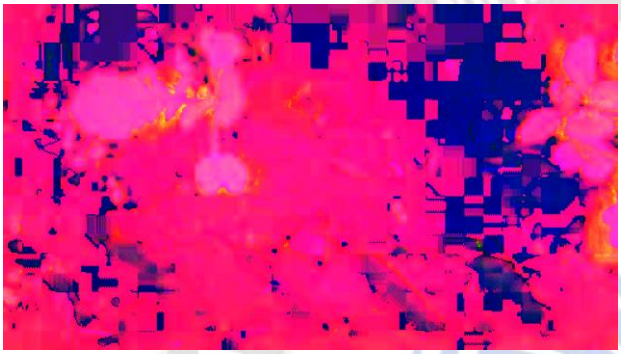

Fig. 4. HSV color space conversion

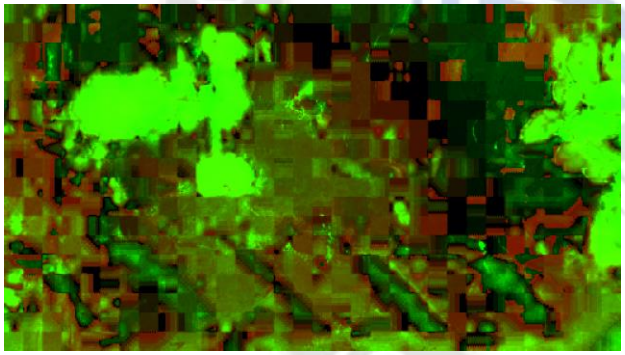

Fig. 5. Saturation enhanced HSV

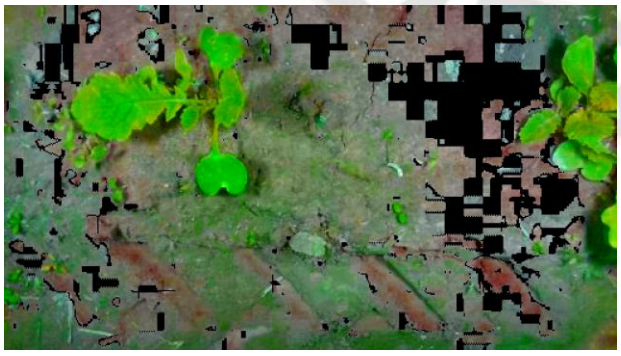

Fig. 6. Enhanced RGB image

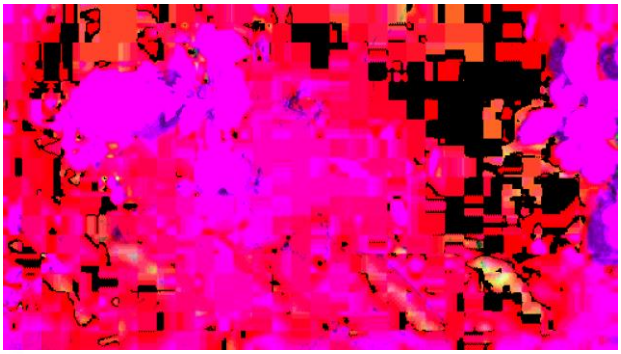

Fig. 7. LAB color space conversion

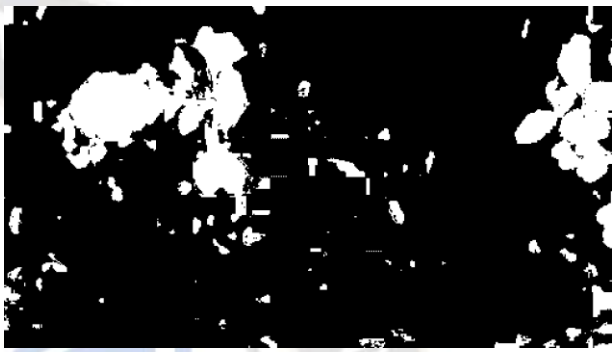

Fig. 8. Binary mask extracted from LAB image

Fig. 9. Binary mask after noise removal

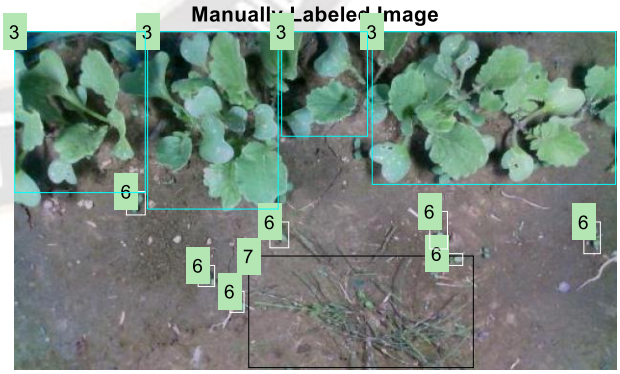

Fig. 10. Manually labelled image

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

Fig. 11. Input image

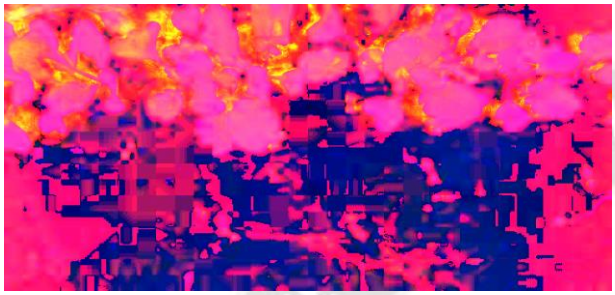

Fig. 12. HSV color space conversion

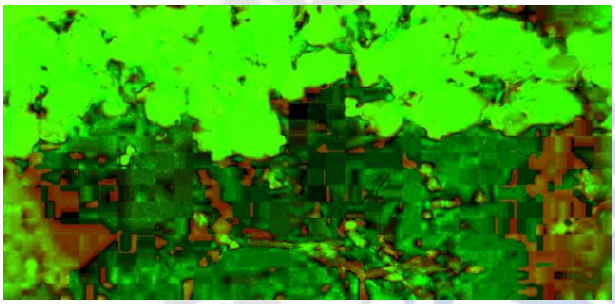

Fig. 13. Saturation enhanced HSV

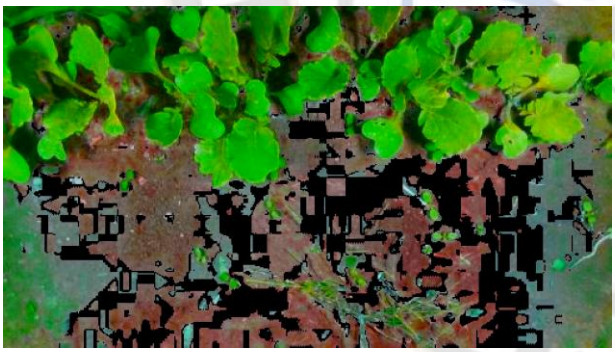

Fig. 14. Enhanced RGB image

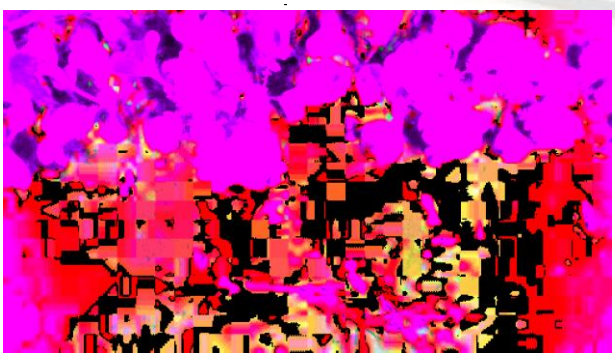

Fig. 15. LAB color space conversion

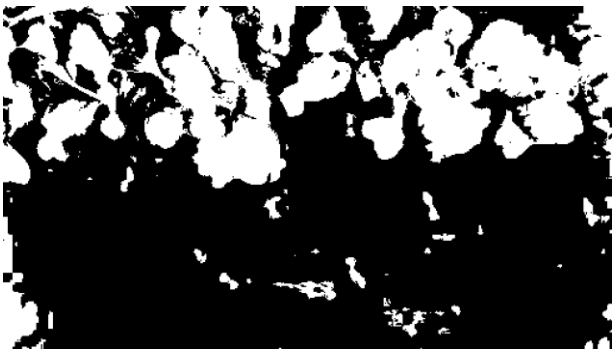

Fig. 16. Binary mask extracted from LAB image

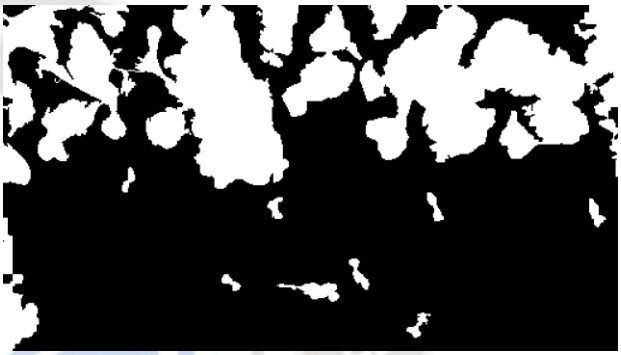

Fig. 17. Binary mask after noise removal

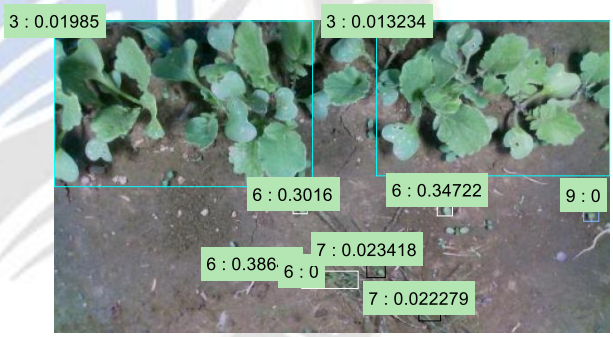

Fig. 18. Manually labelled image

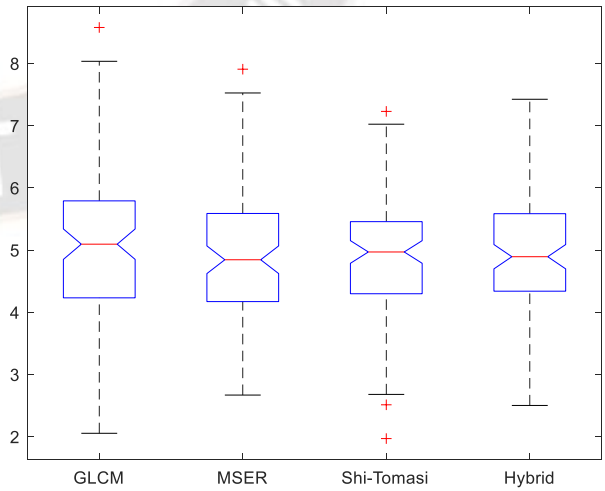

Fig. 19. Box plot comparison of extracted features

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 6
Article Received: *25 March 2021* ***Revised:*** *12 April 2021* ***Accepted:*** *30 May 2021*

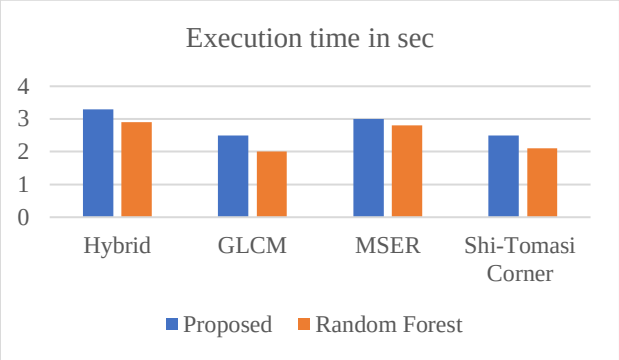

Fig. 20. Execution time of the proposed and random forest classifier on various features in seconds

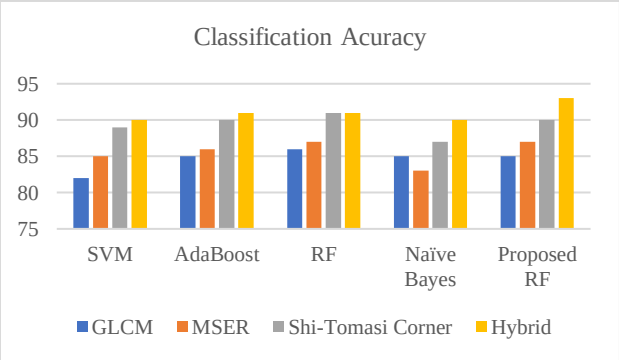

Fig. 21. Accuracy comparison with various classifiers on features learned on classifiers

Table 2. Average feature computation time for an image

Features	Number of features	Time (secs)
Hybrid Features after NCA	1000	0.9287
GLCM	16	0.2200
MSER	60	0.4700
Shi-Tomasi	40	0.2627

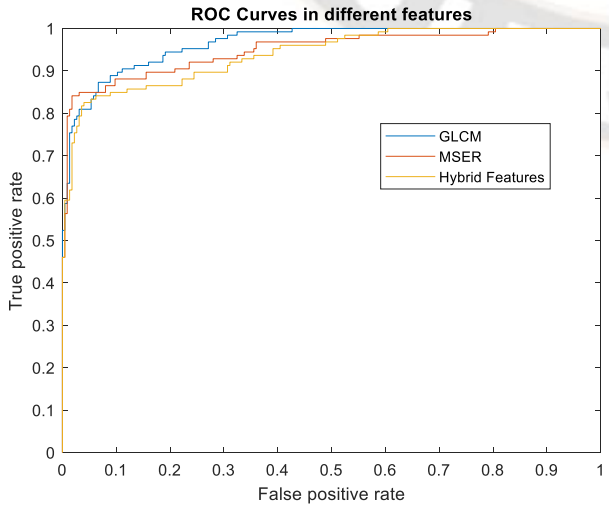

Fig. 22. ROC curves in different features

Table 3. Comparative analysis of classification result with various parameters

Parameters	GLCM	Si-Tomasi	MSER	Hybrid
Accuracy	0.9091	0.9360	0.9375	0.9507
Error	0.0909	0.0640	0.0625	0.0493
Sensitivity	0.9600	0.9592	0.9500	0.9664
Specificity	0.9800	0.9143	0.9857	0.9355
Precision	0.9000	0.9126	0.9333	0.9351
False Positive Rate	0.0200	0.0857	0.0143	0.0645
F1-Score	0.9111	0.9353	0.9314	0.9505
Matthews Correlation Coefficient	0.8998	0.8730	0.9238	0.9018
Kappa	0.7159	0.8720	0.8047	0.9013

VI. Conclusion

The image classification system developed for distinguishing between crops and weeds employs a blend of hybrid features along with feature selection through neighborhood component analysis. Utilizing hybrid features enables a more comprehensive representation of the data, while neighborhood component analysis effectively identifies the most pertinent features for accurate classification.

The system leverages a random forest classifier optimized by the PSO-GSA algorithm to train and categorize the images. This approach enhances classification accuracy by integrating the gravitational search algorithm (GSA) with particle swarm optimization (PSO) to refine the random forest model.

In summary, the proposed methodology presents a robust solution for accurately classifying images of crops and weeds. The combination of hybrid features and neighborhood component analysis ensures the utilization of the most significant information for classification, while the PSO-GSA-optimized random forest classifier provides a strong and efficient framework for data training and classification.

It is important to highlight that the system's accuracy could be affected by the quality of the input images and the specific environmental conditions of the crops and weeds. Therefore, additional testing and assessment of the system in various settings will be essential to validate its effectiveness in practical applications.

References

[1] Monteiro, A. and Santos, S., 2022. Sustainable approach to weed management: The role of

precision weed management. Agronomy, 12(1), p.118.

[2] Chauhan, B.S., 2020. Grand challenges in weed management. Frontiers in Agronomy, 1, p.3.

[3] Gerhards, R., Andujar Sanchez, D., Hamouz, P., Peteinatos, G.G., Christensen, S. and Fernandez-Quintanilla, C., 2022. Advances in site-specific weed management in agriculture—A review. Weed Research, 62(2), pp.123-133.

[4] Hasan, A.M., Sohel, F., Diepeveen, D., Laga, H. and Jones, M.G., 2021. A survey of deep learning techniques for weed detection from images. Computers and Electronics in Agriculture, 184, p.106067.

[5] Mishra, A.M. and Gautam, V., 2021. Weed Species Identification in Different Crops Using Precision Weed Management: A Review. In ISIC (pp. 180-194).

[6] Subeesh, A., Bhole, S., Singh, K., Chandel, N.S., Rajwade, Y.A., Rao, K.V.R., Kumar, S.P. and Jat, D., 2022. Deep convolutional neural network models for weed detection in polyhouse grown bell peppers. Artificial Intelligence in Agriculture, 6, pp.47-54.

[7] Sohail, R., Nawaz, Q., Hamid, I., Gilani, S.M.M., Mumtaz, I., Mateen, A. and Chauhdary, J.N., 2021. An analysis on machine vision and image processing techniques for weed detection in agricultural crops. Pak. J. Agri. Sci, 58(1), pp.187-204.

[8] Ahmad, J., Muhammad, K., Ahmad, I., Ahmad, W., Smith, M.L., Smith, L.N., Jain, D.K., Wang, H. and Mehmood, I., 2021. Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems. Computers in Industry, 98, pp.23-33.

[9] Zhang, W., Miao, Z., Li, N., He, C. and Sun, T., 2022. Review of Current Robotic Approaches for Precision Weed Management. Current Robotics Reports, 3(3), pp.139-151.

[10] Alagbo, O., Spaeth, M., Saile, M., Schumacher, M. and Gerhards, R., 2022. Weed Management in Ridge Tillage Systems—A Review. Agronomy, 12(4), p.910.

[11] Li, Y., Guo, Z., Shuang, F., Zhang, M. and Li, X., 2022. Key technologies of machine vision for weeding robots: A review and benchmark. Computers and Electronics in Agriculture, 196, p.106880.

[12] Rai, N., Zhang, Y., Ram, B.G., Schumacher, L., Yellavajjala, R.K., Bajwa, S. and Sun, X., 2023. Applications of deep learning in precision weed management: A review. Computers and Electronics in Agriculture, 206, p.107698.

[13] Kaur, B., Goyal, B., Dogra, A. and Singh, R., 2023. Citrus fruit content and disease detection using adaptive contrast enhancement. Materials Today: Proceedings.

[14] Louargant, M., Jones, G., Faroux, R., Paoli, J.N., Maillot, T., Gée, C. and Villette, S., 2022. Unsupervised classification algorithm for early weed detection in row-crops by combining spatial and spectral information. Remote Sensing, 10(5), p.761.

[15] Banan, A., Nasiri, A. and Taheri-Garavand, A., 2020. Deep learning-based appearance features extraction for automated carp species identification. Aquacultural Engineering, 89, p.102053.

[16] Khan, S., Tufail, M., Khan, M.T., Khan, Z.A. and Anwar, S., 2021. Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. Precision Agriculture, 22(6), pp.1711-1727.

[17] Anderson, C.J., Heins, D., Pelletier, K.C. and Knight, J.F., 2023. Improving Machine Learning Classifications of Phragmites australis Using Object-Based Image Analysis. Remote Sensing, 15(4), p.989.

[18] Patel, S., Swaminarayan, P., Pabla, S., Mandla, M. and Narendra, H., 2022. Machine Learning-Based Multi-temporal Image Classification Using Object-Based Image Analysis and Supervised Classification. In Smart Trends in Computing and Communications: Proceedings of SmartCom 2022 (pp. 223-233). Singapore: Springer Nature Singapore.

[19] Sudars, K., Jasko, J., Namatevs, I., Ozola, L. and Badaukis, N., 2020. Dataset of annotated food crops and weed images for robotic computer vision control. Data in brief, 31, p.105833.