JOURNAL OF BASIC SCIENCE AND ENGINEERING

AI-POWERED WEED DETECTION SYSTEM FOR PRECISION FARMING

S. Anand[1] and C. Ashok Kumar[2]

[1]Research Scholar, Department of Computer and Information Science, Faculty of Science,
Annamalai University, Tamilnadu, India

[2]Assistant Professor, Department of Computer Science, Dr. M.G.R. Government Arts and Science College for Women, Viluppuram, Tamilnadu, India

Abstract

In precision agriculture, where maximizing crop yield while minimizing resource usage is paramount, the need for efficient weed detection models is critical. Weeds pose a significant threat to crop productivity, competing for essential resources such as nutrients, water, and sunlight. Traditional methods of weed control, such as manual labor or blanket herbicide application, are time-consuming, labor-intensive, and often result in overuse of chemicals, leading to environmental degradation and economic inefficiency. This paper presents a Hyperparameter-Tuned Deep Learning model for Weed Detection and Classification (HPTDL-WDAC) suitable for Precision Farming applications. The proposed HPTDL-WDAC system integrates advanced techniques from computer vision and deep learning to accurately identify and classify weeds in agricultural fields. The workflow begins with pre-processing steps aimed at enhancing image quality and reducing noise. Specifically, a Gaussian Filter (GF) is employed to effectively remove noise from input images, followed by resizing to standard dimensions and class labelling for subsequent analysis.

For object detection and classification, the RetinaNet model is employed. RetinaNet's innovative architecture, featuring a focal loss mechanism, enables robust detection of weed instances amidst varying backgrounds and lighting conditions. Notably, the hyperparameters of the RetinaNet model are fine-tuned using the ADAM optimizer, optimizing its performance for the specific task of weed detection and classification in precision farming scenarios. A thorough simulation analysis of the HPTDL-WDAC technique was conducted using a benchmark dataset. Experimental results demonstrate the effectiveness of the proposed system in accurately detecting and classifying weeds in various agricultural systems. This shows that it exhibits improved results compared to recent approaches on various metrics.

Keywords: Precision agriculture, Weed classification, Gaussian Filter, RetinaNet, ADAM optimizer, Hyperparameters

1. Introduction

Precision farming, also known as precision agriculture, is an approach to farming that utilizes technology and data-driven methods to optimize crop production while minimizing waste and environmental impact. It involves the use of various technologies such as GPS, sensors, drones, and AI to collect and analyze data about soil conditions, weather patterns, crop health, and other factors that affect agricultural productivity. By precisely tailoring inputs such as water, fertilizers, and pesticides to the specific needs of individual plants or sections of fields, precision farming aims to maximize yields, reduce costs, and minimize the negative impacts of agriculture on the environment [1].

Weeds pose a significant challenge in modern agriculture and can have detrimental effects on crop yield, quality, and overall farm profitability. Weeds compete with crops for essential resources such as nutrients, water, and sunlight, leading to reduced growth and yield. Additionally, some weeds can release harmful chemicals or serve as hosts for pests and diseases, further impacting crop health. Traditional methods of weed control, such as manual labour or blanket herbicide application, are often inefficient, labour-intensive, and can have adverse environmental consequences such as soil erosion, water contamination, and biodiversity loss [2]. The need for effective weed management models in modern agriculture is therefore paramount. Weed management models leverage advanced technologies such as computer vision, machine learning, and robotics to accurately detect, identify, and control weeds in agricultural fields. These models enable farmers to precisely target interventions, such as selective herbicide application or mechanical weeding, based on real-time data and analysis of weed distribution and abundance. By optimizing weed control practices, weed management models help farmers reduce reliance on chemical inputs, minimize environmental impact, and improve overall farm productivity and profitability [3].

1.1 Paper Contributions

This paper proposes a novel AI-powered weed detection and classification model for precision farming. The methodology begins with pre-processing steps aimed at optimizing the quality of input images. To this end, a Gaussian Filter is applied to remove noise, ensuring clearer and more accurate detection results. Additionally, resizing and class labelling are performed to standardize the data and facilitate subsequent analysis. Central to the weed detection system is the utilization of the RetinaNet model for object detection and classification. RetinaNet, renowned for its efficiency and accuracy in detecting objects of varying sizes and orientations, is well-suited for the dynamic and complex environments encountered in agricultural settings. Importantly, the hyperparameters of the RetinaNet model are fine-tuned using the ADAM optimizer, optimizing its performance for the specific task of weed detection and classification. By integrating these

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

advanced technologies, the proposed system offers farmers a powerful tool for proactive weed management in precision farming.

1.2 Paper Organizations

The remainder of the investigation is structured as follows: Section 2 discusses the research works published in this field, Section 3 provides a detailed explanation of the functionality of the HPTDL-WDAC technique, and Section 4 presents the experimental findings. Finally, Section 5 offers concluding remarks for the study.

2. Related works

Z. Wu et al. [4] provides a thorough and insightful overview of the various computer vision techniques applied to weed detection in precision agriculture. The authors effectively categorize and analyze different methodologies, including image preprocessing, feature extraction, machine learning, deep learning, and sensor technologies. Their critical evaluation of these methods, along with a discussion of performance metrics, highlights the advancements and challenges in the field. A.H. Al-Badri et al. [5] proposes an extensive review of the application of machine learning techniques for weed classification. The authors thoroughly examine the current methodologies, highlighting their strengths and limitations, and discuss the various challenges faced in the field, such as dataset quality, computational requirements, and model interpretability. Jingning Yu [6] provides an in-depth analysis of using Gaussian filters to enhance image quality by mitigating the effects of Gaussian and pepper noise. The author explores the mathematical foundations and implementation details of Gaussian filtering, demonstrating its effectiveness in noise reduction and image restoration. R. Punithavathi et al. [7] introduces a sophisticated model integrating computer vision and deep learning for effective weed detection in precision agriculture. The authors detail a comprehensive approach, including noise removal using WF, followed by the application of the Multi-Scale Faster RCNN model for object detection, and ELM for classification. Hyperparameter tuning with the Farmland Fertility Optimizer further enhances model performance. The study demonstrates significant improvements over existing methods, making a notable contribution to advancing sustainable farming practices through precise and automated weed management. N. Rai et al. [8] provides a comprehensive review of the state-of-the-art deep learning techniques applied to precision weed management. The authors systematically explore various deep learning models and their applications in weed detection, classification, and management, highlighting their advantages, limitations, and potential for improving agricultural productivity.

N. Mohanad et al. [9] develops an innovative application of the RetinaNet deep learning architecture for dual tasks of object detection and distance estimation. The authors detail the methodology, highlighting the robustness and accuracy of RetinaNet in identifying objects within an image and simultaneously estimating their distances. Through extensive experimentation, the

study demonstrates the model's effectiveness and potential applications in various fields, including autonomous navigation and surveillance. Mouna Afif et al. [10] offers an insightful evaluation of the RetinaNet deep learning model for its applicability in assisting navigation for blind and visually impaired individuals. The authors rigorously assess the performance of RetinaNet in detecting indoor objects, focusing on its accuracy, reliability, and real-time processing capabilities. R. Kamath et al. [11] proposes an innovative approach for differentiating between paddy crops and weeds using a multiple classifier system. The authors explore the integration of various machine learning classifiers to enhance the accuracy and robustness of weed detection in paddy fields. N. Islam et al. [12] performs a thorough investigation into the application of image processing and machine learning for early weed detection in chilli farms. The authors present a robust methodology combining advanced image processing techniques with machine learning algorithms to accurately identify and differentiate weeds from chilli plants at an early growth stage. O.G. Ajayi et al. [13] offers a comprehensive evaluation of the YOLO v5 model for classifying crops and weeds using images captured by unmanned aerial vehicles (UAVs). The authors meticulously assess the model's performance in terms of accuracy, speed, and reliability, demonstrating its effectiveness in real-time agricultural applications.

3. The Proposed Model

The present research introduces a novel HPTDL-WDAC technique designed to effectively distinguish between plants and weeds in precision agriculture. This technique incorporates several subprocesses, including GF-based preprocessing, RetinaNet-based object detection and classification, and ADAM-based parameter optimization. The proposed model successfully discerns weeds from crops, thereby reducing herbicide usage and enhancing productivity. Figure 1 provides an overview of the entire process of the HPTDL-WDAC technique.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

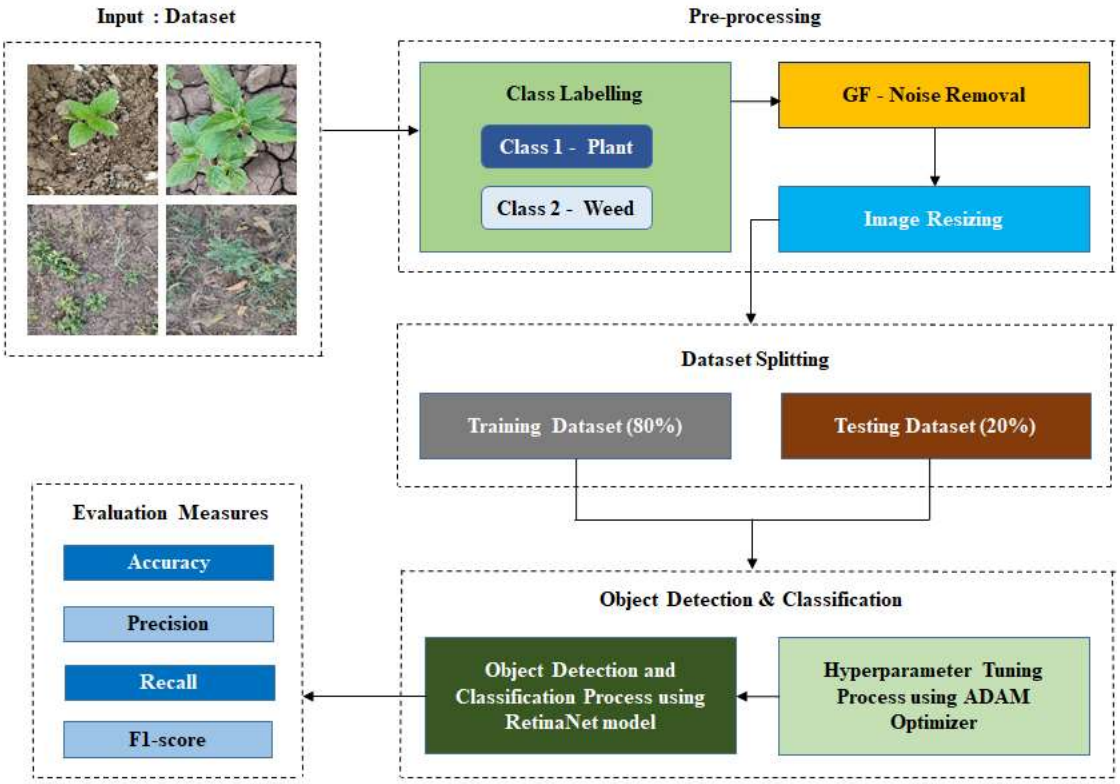


Figure 1: Overall process of HPTDL-WDAC technique

3.1 Image Pre-processing

Image pre-processing is a crucial step in preparing raw image data for subsequent analysis and interpretation. It involves several operations aimed at enhancing image quality, reducing noise, and standardizing data for further processing. This step is essential in weed classification systems to ensure the accuracy and reliability of the classification process. In the proposed model, pre-processing comprises three essential processes: class labelling, noise removal, and subsequent image resizing to facilitate further processing.

3.1.1 Class Labelling

Class labelling is a crucial step in the pre-processing of image data, particularly in supervised learning tasks such as object detection or classification. In the context of weed classification, class labelling involves assigning categorical labels to images to indicate the presence of either plants or weeds. This labelling is essential for training machine learning algorithms to recognize and distinguish between different classes of objects in images. By providing ground truth labels for each image in the dataset, class labelling enables the model to learn the relationship between input features and their corresponding class labels. This supervised learning process allows the model

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

to generalize its understanding of different classes of objects and make accurate predictions on unseen data. Ultimately, accurate class labelling is essential for the development of robust and reliable machine learning models for weed classification in precision agriculture.

3.1.2 Noise Removal using GF technique

At the outset, the Gaussian filter approach can be employed to eliminate noise present in the image. Noise removal is an image pre-processing technique designed to enhance the features of the image corrupted by noise. The Gaussian filter operates by convolving the image with a Gaussian kernel, effectively smoothing out high-frequency noise components while preserving the underlying structures and edges in the image.

The 2D Gaussian function is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (1)$$

Where:

$G(x,y)$ represents the value of the Gaussian kernel at position (x,y)

σ is the standard deviation of the Gaussian distribution,

x and $y$are the spatial coordinates within the kernel.

The convolution operation involves sliding the Gaussian kernel over the image and computing the weighted sum of pixel values within the kernel window at each position. It can be represented as:

$$I_{smoothed}(x,y) = \sum_{i=-k}^{k}\sum_{j=-k}^{k} I(x-i, y-j).G(i,j) \quad (2)$$

Where:

$I(x,y)$ is the original pixel intensity at position (x,y)

$G(i,j)$ is the value of the Gaussian kernel at position (i,j)

k is the size of the Gaussian kernel

After applying the Gaussian filter, high-frequency noise components in the image are attenuated, resulting in a smoother and cleaner image. This noise reduction enhances the quality of the image and improves the performance of subsequent processing tasks, such as weed detection and classification.

3.1.3 Image Resizing

Image resizing is performed to standardize the dimensions of input images, ensuring consistency across the dataset and facilitating efficient processing. Resizing involves scaling the image to a predefined width and height while preserving the aspect ratio. The image resizing can be represented as:

$$Resized\_Image(i) = resize(Original\_Image(i), width, height) \quad (3)$$

Where Original_Image(i) represents the i^{th} original image, and width and height denote the desired dimensions of the resized image.

3.2 Dataset Splitting

Dataset splitting is a critical step in machine learning and statistical modeling, where the available dataset is divided into separate subsets for training, and testing. Training set used to train the model and the testing used to evaluate the model's performance and assess its generalization to unseen data. Let's denote the entire dataset as D containing N samples. The dataset splitting process can be represented as follows:

$$Training\ Set(D_{train}) = \{(x_i, y_i)\}_{i=1}^{N_{train}} \quad (4)$$

$$Testing\ Set(D_{test}) = \{(x_i, y_i)\}_{i=1}^{N_{test}} \quad (5)$$

Dataset splitting allows us to assess how well our model generalizes to unseen data. By evaluating the model on a separate testing set, we obtain an unbiased estimate of its performance. Training a model on the entire dataset without validation or testing can lead to overfitting, where the model learns to memorize the training data rather than capturing underlying patterns. Dataset splitting helps mitigate overfitting by providing a separate testing set for evaluation. This is crucial for deploying the model in real-world applications.

3.3 Optimized RetinaNet based Object Detection and Classification

This manuscript employs the RetinaNet technique for the effective detection and classification of weed and crop images. The RetinaNet deep learning architecture is a sophisticated model tailored for high-accuracy object detection, making it ideal for weed detection and classification in precision agriculture. The process begins with input images of agricultural fields, which are passed through the backbone network, typically a pre-trained ResNet-50 or ResNet-101. This network extracts hierarchical features from the images at various levels of abstraction. These features are then fed into a Feature Pyramid Network (FPN), which constructs a multi-scale feature pyramid, enabling the detection of weeds at different sizes and scales. The model includes two specialized subnets: the classification subnet and the box regression subnet. The classification subnet processes the feature maps to predict the probability of each anchor box containing a weed, plant, or background. The box regression subnet predicts the coordinates of the bounding boxes around the detected objects. The final output of RetinaNet consists of class probabilities and precise bounding box coordinates for each detected weed and plant, enabling

accurate and effective weed management by allowing for targeted interventions in the agricultural fields. The entire detection process is depicted in Figure 2.

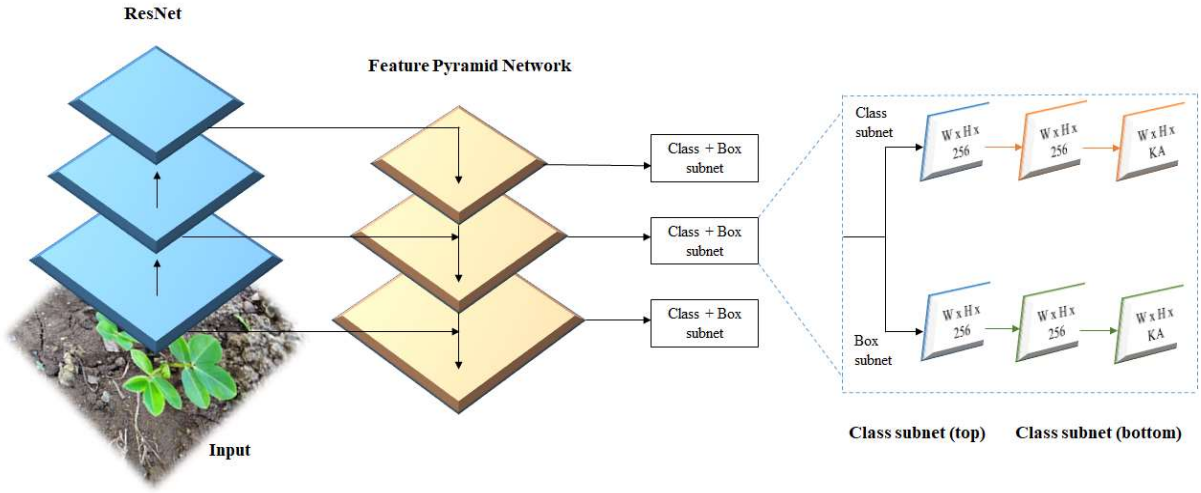


Figure 2: RetinaNet Model Architecture

3.3.1 Backbone Network

The backbone network in the RetinaNet model is typically a pre-trained convolutional neural network such as ResNet-50 or ResNet-101. This network is responsible for extracting hierarchical feature maps from the input image. This manuscript utilizes the ResNet-50 as backbone network of the proposed model. The backbone network processes an input image I through a series of convolutional layers, which can be described as follows:

- The input image I is first convolved with a set of filters to produce the initial feature map F_0:

$$F_0 = Conv(I, W_0) + b_0 \quad (6)$$

where W_0 and b_0 are the weights and biases of the initial convolutional layer.

- This feature map F_0 is then passed through a non-linear activation function (typically ReLU) and pooling layer:

$$F_0 = ReLU(F_0) \quad (7)$$

$$F_0 = MaxPool(F_0) \quad (8)$$

- The feature maps are further processed through a series of residual blocks. Each residual block consists of multiple convolutional layers with skip connections to allow for easier gradient flow. For a residual block at layer l:

$$F_l = ReLU(Conv(F_{l-1}, W_l) + b_l \quad (9)$$

Here, F_{l-1} is the input feature map to the l-th residual block, and W_l and b_l are the weights and biases for the convolutions within the block.

- The output of the residual block is added to its input through a skip connection:

$$F_l = F_l + F_{l-1} \quad (10)$$

- The backbone network outputs feature maps from different layers, typically after each stage of residual blocks. These feature maps correspond to different levels of the network, capturing different spatial resolutions and feature complexities. The feature pyramid in the FPN is built by these feature maps.

3.3.2 Feature Pyramid Network

The Feature Pyramid Network (FPN) in the RetinaNet model constructs a multi-scale feature representation from the backbone network's outputs, enabling robust detection of objects of various sizes. Let's denote the feature maps extracted from the backbone network as C_2, C_3, C_4, and C_5. These feature maps correspond to different levels of the backbone network, where C_2 has the highest spatial resolution and C_5 has the lowest.

Top-Down Pathway: The FPN starts by creating a top-down pathway, where higher-level (coarser) feature maps are upsampled and merged with lower-level (finer) feature maps.

Lateral Connections: Lateral connections are used to combine the upsampled feature maps with the corresponding feature maps from the backbone network.

Initial Coarser Feature Map

$$P_5 = Conv1x1(C_5) \quad (11)$$

Here, $Conv1x1$ denotes a 1x1 convolution used to reduce the number of channels.

Subsequent Feature Maps

For levels P_l where l=4,3,2:

$$P_l = Conv1x1(C_l) + UpSample(P_{l+1}) \quad (12)$$

$Conv1x1(C_l)$: 1x1 convolution to reduce the number of channels in C_l.

$UpSample(P_{l+1})$: Upsampling the coarser feature map P_{l+1} by a factor of 2.

The construction of the finer features maps of FPN is

$$P_4 = Conv3x3(Conv1x1(C_4) + UpSample(P_5)) \quad (13)$$

$$P_3 = Conv3x3(Conv1x1(C_3) + UpSample(P_4)) \quad (14)$$

$$P_2 = Conv3x3(Conv1x1(C_2) + UpSample(P_4)) \quad (15)$$

Final Feature Maps: After combining the feature maps using the lateral connections, each P_l undergoes a 3x3 convolution to generate the final feature maps, which helps to reduce aliasing artifacts from the upsampling:

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

$$P_l = Conv3x3(P_l) \quad (16)$$

where $Conv3x3$ denotes a 3x3 convolution operation.

Output of FPN

The output of the FPN is a set of feature maps P_2, P_3, P_4, and P_5. These feature maps capture rich, multi-scale information, allowing the subsequent RetinaNet subnets to perform robust object detection and classification across different object sizes.

3.3.3 Classification Subnet

The classification subnet is applied to each level of the feature pyramid, producing outputs for each spatial location on the feature map. It predicts the probability of each anchor box containing a specific object class (weed or plant). The steps involved in the classification subnet are explained below.

Input Feature Maps: The input to the classification subnet consists of the feature maps P_l from each level l of the FPN.

Convolutional Layers: The classification subnet typically consists of a series of shared convolutional layers, followed by a final convolutional layer that outputs the class probabilities. Let's denote the shared convolutional layers as a series of k convolutional operations.

$$H_l^{(i)} = ReLU\left(Conv\left(H_l^{(i-1)}, W^{(i)}\right) + b^{(i)}\right), i = 1,2,\dots,k \quad (17)$$

Where $H_l^{(0)} = P_l$

Final Classification Layer: The final layer produces the class scores for each anchor box. If there are A anchors per spatial location and C object classes, the final output of the classification subnet has dimensions ($H \times W \times A \times C$) for each level l, where H and W are the height and width of the feature map. The final classification layer applies a convolution to produce the class scores:

$$C_l = Conv\left(H_l^{(k)}, W_c\right) + b_c \quad (18)$$

Here, C_l is the class score tensor for feature map level l, W_c is the weight tensor of the final convolutional layer, and b_c is the bias.

Sigmoid Activation: The class scores are converted to probabilities using the sigmoid function.

$$P(c|x) = Sigmoid(C_l) \quad (19)$$

where $P(c|x)$ is the predicted probability for class c given the feature map x.

These predictions are then used to determine the presence and types of objects (weeds, plants) in the input images, facilitating effective weed detection and classification in precision agriculture applications.

3.3.4 Box Regression Subnet

The box regression subnet is responsible for predicting the coordinates of bounding boxes for objects detected in the input image. This subnet processes the feature maps from each level of the Feature Pyramid Network to produce these bounding box predictions. The following describes the procedures that make up the box regression subnet.

Input Feature Maps: The input to the box regression subnet consists of the feature maps P_l from each level l of the FPN.

Convolutional Layers: Similar to the classification subnet, the box regression subnet consists of a series of shared convolutional layers. Let's denote these shared convolutional operations as a series of k convolutional layers.

$$H_l^{(i)} = ReLU\left(Conv\left(H_l^{(i-1)}, W^{(i)}\right) + b^{(i)}\right), i = 1,2,\dots,k \quad (20)$$

Where $H_l^{(0)} = P_l$

Final Box Regression Layer: The final layer produces the bounding box coordinates for each anchor box. If there are A anchors per spatial location, the final output of the box regression subnet has dimensions ($H \times W \times A \times 4$) for each level l, where H and W are the height and width of the feature map and 4 corresponds to the 4 coordinates of the bounding box (x, y, w, h). Where (x, y) represents the coordinates of the center of the bounding box, and w and h denote the width and height of the bounding box respectively. The final box regression layer applies a convolution to produce the bounding box coordinates:

$$B_l = Conv\left(H_l^{(k)}, W_b\right) + b_b \quad (21)$$

Here, B_l is the bounding box coordinate tensor for feature map level l, W_b is the weight tensor of the final convolutional layer, and b_b is the bias.

In precision agriculture applications, these predictions are used to localize the objects (plants, weeds) identified in the input images, enabling precise and accurate weed detection and classification.

3.3.5 Focal Loss

Focal Loss is a specialized loss function used in the RetinaNet model to address the issue of class imbalance during training, particularly in object detection tasks where there are many more background examples than object examples. The focal loss modifies the standard cross-entropy loss by adding a factor that down-weights the loss assigned to well-classified examples. This ensures that the model focuses more on hard-to-classify examples. As per the proposed model, the focal loss can be defined as follows:

- p_t be the model's estimated probability for the ground truth class. If y=1 (positive class), then p_t=p. If y=0 (negative class), then p_t=1−p.
- α be the weighting factor for the positive class to address class imbalance.
- γ be the focusing parameter that reduces the loss contribution from easy examples and extends the range in which an example receives a low loss.

The focal loss FL is given by:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (22)$$

Where,

$p_t = \begin{cases} p & if\ y=1 \\ 1-p & if\ y=0 \end{cases}$ Here, p is the predicted probability of the class being 1.

$\alpha_t = \begin{cases} \alpha & if\ y=1 \\ 1-\alpha & if\ y=0 \end{cases}$ Here, the weighting factor α helps balance the importance of positive and negative examples.

$(1-p_t)^{\gamma}$ reduces the relative loss for well-classified examples, focusing more on hard examples. $\log(p_t)$ is the standard log-loss for the correct class.

3.3.6 Non-Maximum Suppression

Non-Maximum Suppression (NMS) is a crucial post-processing step in object detection models, including RetinaNet. It helps to filter out multiple detections of the same object by retaining the most confident one and suppressing the others based on their overlap. The procedures followed in the NMS are described below.

Initialization

- Let $\{b_1,b_2,...,b_N\}$ be the set of predicted bounding boxes.
- Let $\{s_1,s_2,...,s_N\}$be the corresponding confidence scores for these bounding boxes.

Sorting

- Sort the bounding boxes by their confidence scores in descending order. Assume after sorting, the indices are rearranged such that $s_1 \geq s_2 \geq ... \geq s_N$.

Intersection over Union (IoU)

- Compute the IoU for each pair of bounding boxes to determine their overlap. The IoU between two bounding boxes b_i and b_j is defined as:

$$IoU(b_i, b_j) = \frac{Area(b_i \cap b_j)}{Area(b_i \cup b_j)} \quad (23)$$

Here, $Area(b_i \cap b_j)$ is the area of the intersection of b_i and b_j, and $Area(b_i \cup b_j)$ is the area of their union.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Algorithm

- Initialize an empty list to hold the indices of the final bounding boxes: $Selected = []$
- For each bounding box b_i in the sorted list:
  - Compare b_i with all previously selected boxes using IoU.
  - If b_i has a high overlap with any selected box, discard b_i.
  - If b_i has a low overlap with all selected boxes, add b_i to the list of selected boxes: $Selected = Selected \cup \{i\}$

Output

- The final list of indices $Selected$ corresponds to the bounding boxes that are retained after NMS.

In the proposed model, NMS helps to

- Eliminate redundant detections of the same object, which can occur due to the dense sampling of anchor boxes.
- Ensure that only the most confident detection is kept for each object, improving the clarity and accuracy of the final detections.
- Reduce the number of false positives, which is critical for accurate object detection and classification, particularly in applications such as precision agriculture where distinguishing between weeds and crops accurately is essential.

3.4 Hyperparameter Tuning using ADAM Optimizer

Hyperparameter tuning in deep learning models like RetinaNet is essential for optimizing performance. The ADAM (Adaptive Moment Estimation) optimizer is widely used due to its adaptive learning rate and efficient computation. It combines the advantages of RMSProp and AdaGrad algorithms. The update rules for ADAM are as follows:

Initialize parameters:

- Learning rate: α
- Exponential decay rates for moment estimates: β_1, β_2
- Small constant for numerical stability: ϵ

Initialize first moment m and second moment v to 0:

m_0=0, v_0=0

Compute biased estimates of first and second moments:

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

For each parameter θ_t:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (24)$$

$$\vartheta_t = \beta_2 \vartheta_{t-1} + (1-\beta_2) g_t^2 \quad (25)$$

where g_t is the gradient of the loss function at time step t.

Compute bias-corrected first and second moments:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (26)$$

$$\widehat{\vartheta}_t = \frac{\vartheta_t}{1-\beta_2^t} \quad (27)$$

Update parameters:

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{\vartheta}_t + \epsilon}} \quad (28)$$

3.4.1 Applying ADAM in RetinaNet Hyperparameter Tuning

Learning Rate (α): The learning rate is crucial for the convergence speed and stability of the training process.
Batch Size: The batch size affects the stability of the training process and the memory usage.
Epochs: The number of epochs determines how many times the model will iterate over the entire training dataset. It is tuned to ensure the model adequately learns without overfitting.
Anchor Scales and Aspect Ratios: These hyperparameters are specific to RetinaNet and determine the shapes and sizes of the anchor boxes.
Focal Loss Parameters (γ and α): The parameters of the focal loss function need to be tuned to balance the contribution of easy and hard examples.

During training, for each mini-batch of training examples, compute the gradient g_t of the loss with respect to each model parameter θ_t. Then, update θ_t using the ADAM update rules. Repeat this process for each epoch until the model converges.

By using ADAM, the model parameters θ_t are updated adaptively, leading to potentially faster and more stable convergence compared to standard SGD, especially when dealing with the complex, high-dimensional parameter space of RetinaNet.

4. Results and Discussion

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

This section of the manuscript presents a detailed exposition of the proposed approach, supported by a comprehensive evaluation. The effectiveness of our strategy is validated through careful consideration of various metrics, including accuracy, precision, recall, and F1-score. Notably, the proposed technique stands out from existing methods such as Inception-v3, VGG16, and RCNN. The HPTDL-WDAC approach is specifically tailored for plant and weed image data collected from [24], consisting of a total of 1300 images. These images are divided into two categories: plant and weed, with 634 and 666 images, respectively. Figure 3 illustrates examples of plant and weed images.

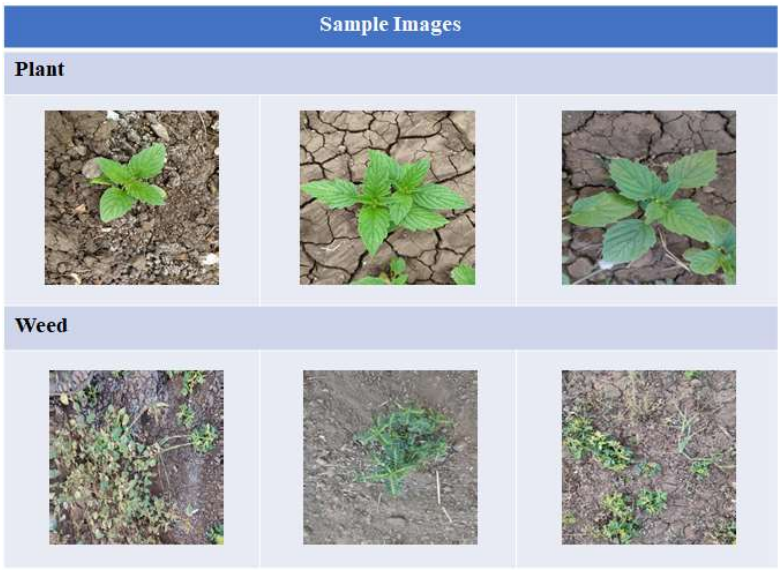


Figure 3: Sample images for three different classes

Out of the 1300 collected images, 1040 are used for training the proposed classifier, while the remaining images are reserved for testing. The implementation details of the HPTDL-WDAC technique are outlined in Table 1. Additionally, Figure 4 presents the various stages of the proposed model, starting with image preparation, followed by splitting the dataset into 80% for training and 20% for testing, and finally training and testing the model. Figure 5 displays the detection and classification results for the test images. The validation of the approach is confirmed through empirical evaluations, with detailed metrics provided in this section.

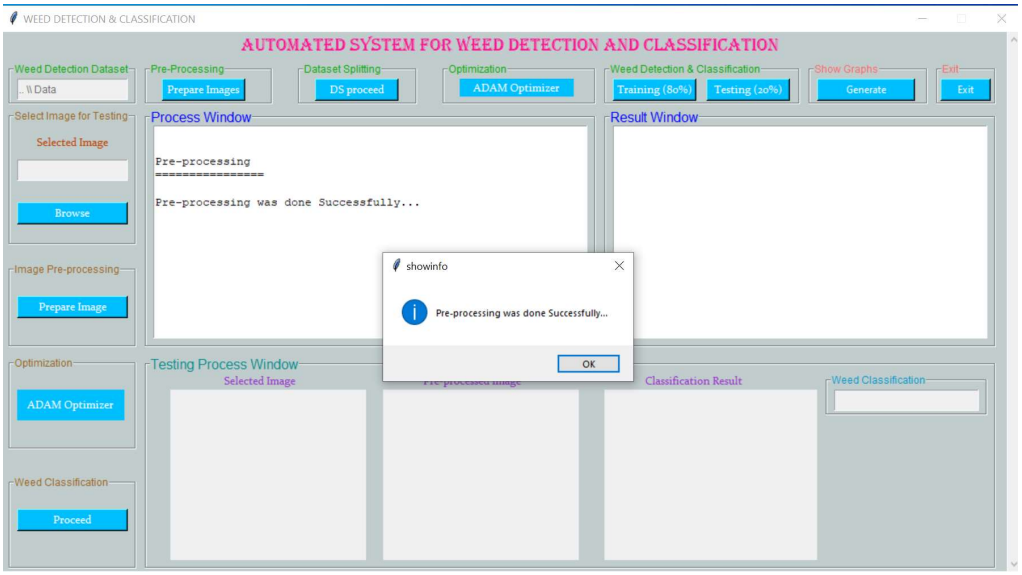
Table 1: Simulation Variables

S. No.	Method	Description	Value
1	HPTDL-WDAC	Learning Rate	1e-5
2		Batch Size	32

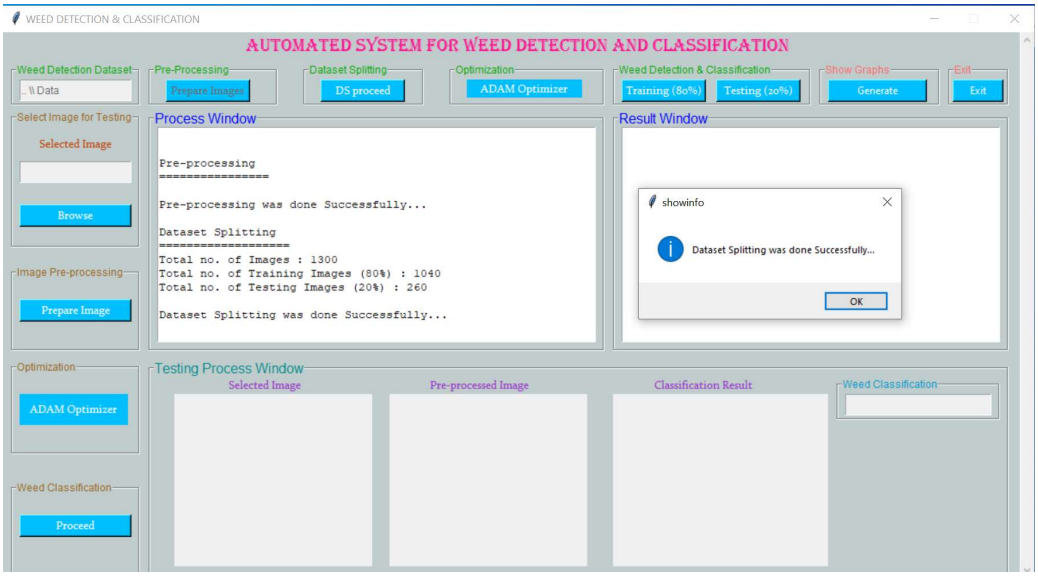
Vol. 21, No. 1, (2024)
ISSN: 1005-0930

3		Epochs	100
4		Anchor Scales	[0.1,0.2,0.4]
5		Aspect Ratios	[0.5,1.0,2.0]
6		Focal Loss Parameters	γ =1.0 α = 0.5

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

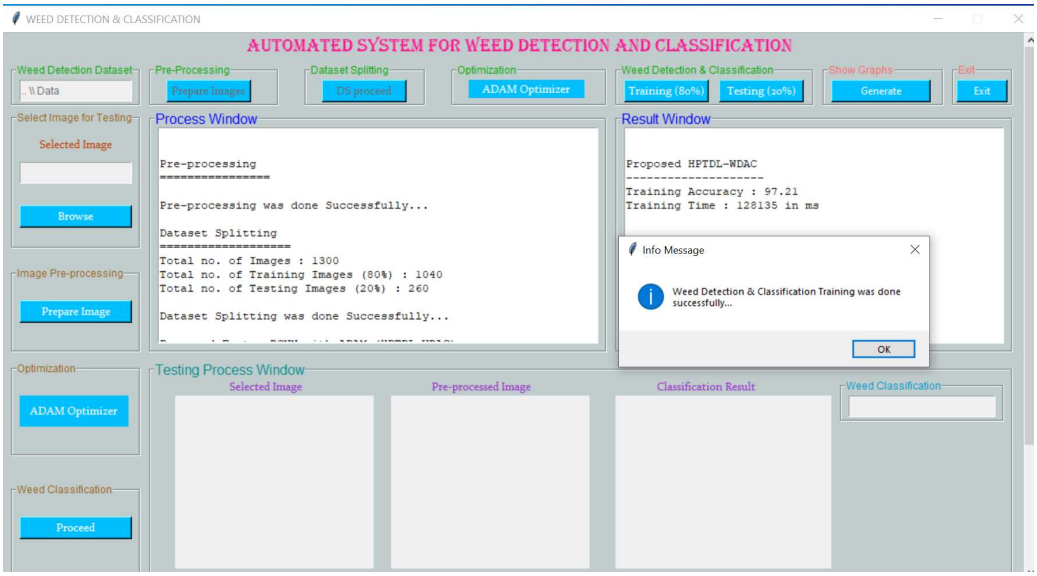


(a)

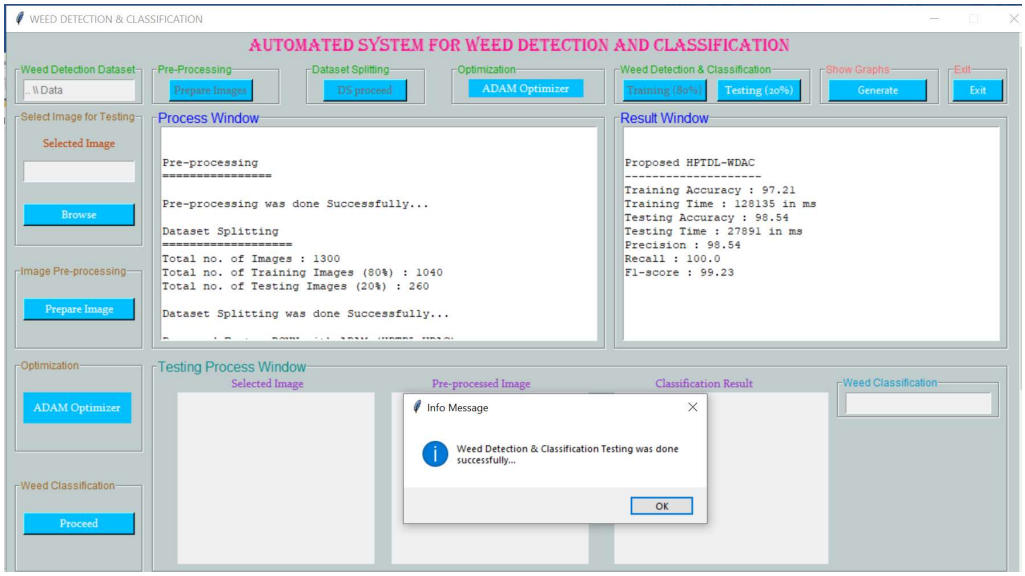


(b)

Vol. 21, No. 1, (2024)
ISSN: 1005-0930



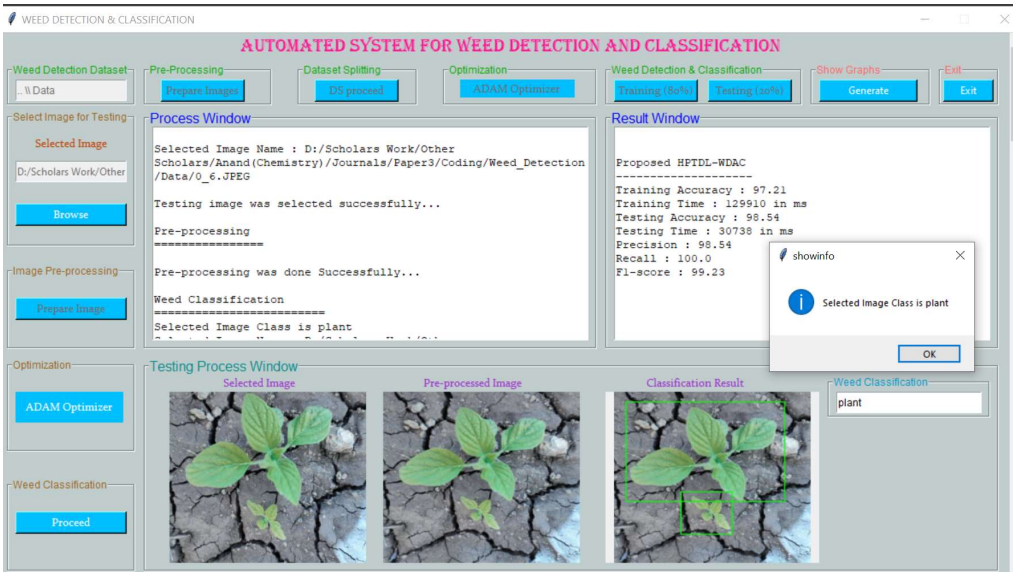
(c)



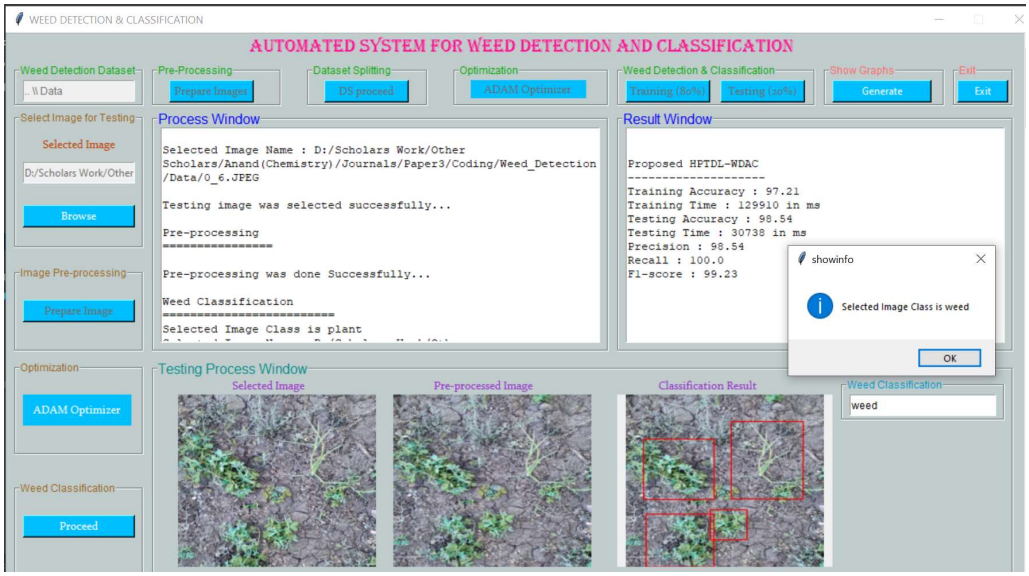
(d)

Figure 4: Train and Test the model with the dataset (a) Prepare images (b) Split the dataset (c) Results attained during training (d) Results attained during testing

Vol. 21, No. 1, (2024)
ISSN: 1005-0930



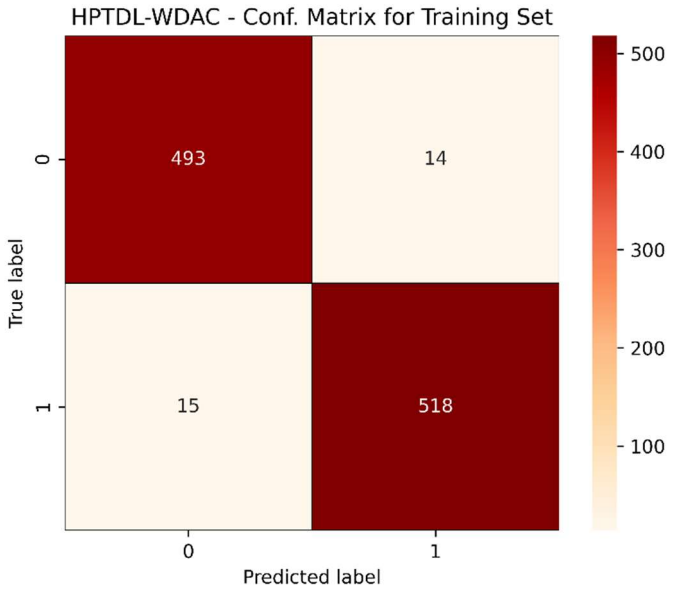
(a)



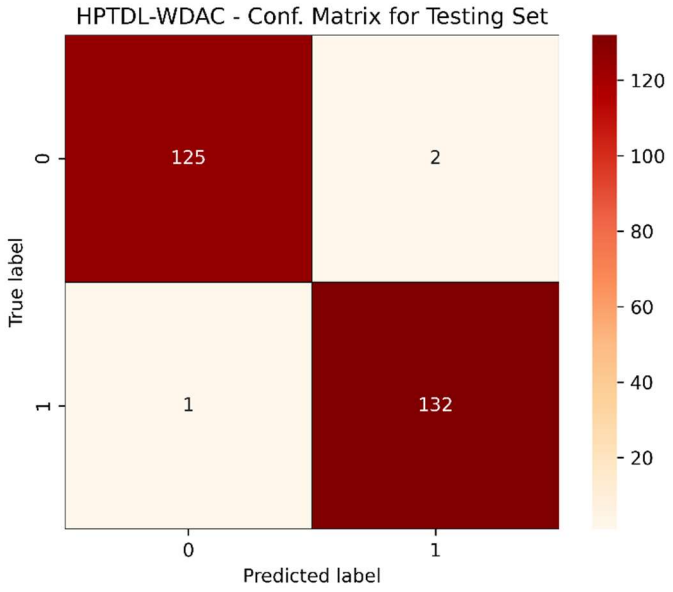
(b)

Figure 5: Object detection and classification results for a test image (a) Plant (b) Weed

Figure 6 demonstrates a set of results attained by the HPTDL-WDAC algorithm under training and testing phase.



(a)



(b)

Figure 6: Confusion matrix for Classification analysis of HPTDL-WDAC approach (a) Training phase (b) Testing phase

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Figure 7 illustrates the detailed classification outcomes produced by the HPTDL-WDAC model. The empirical results demonstrate that the HPTDL-WDAC approach consistently performs well across all evaluated metrics. Specifically, on the testing dataset, the HPTDL-WDAC method achieved an accuracy of 98.85%, a precision of 98.43%, a recall of 99.21%, and an F1-score of 98.82%. These metrics underscore the model's robustness and effectiveness in distinguishing between plants and weeds.

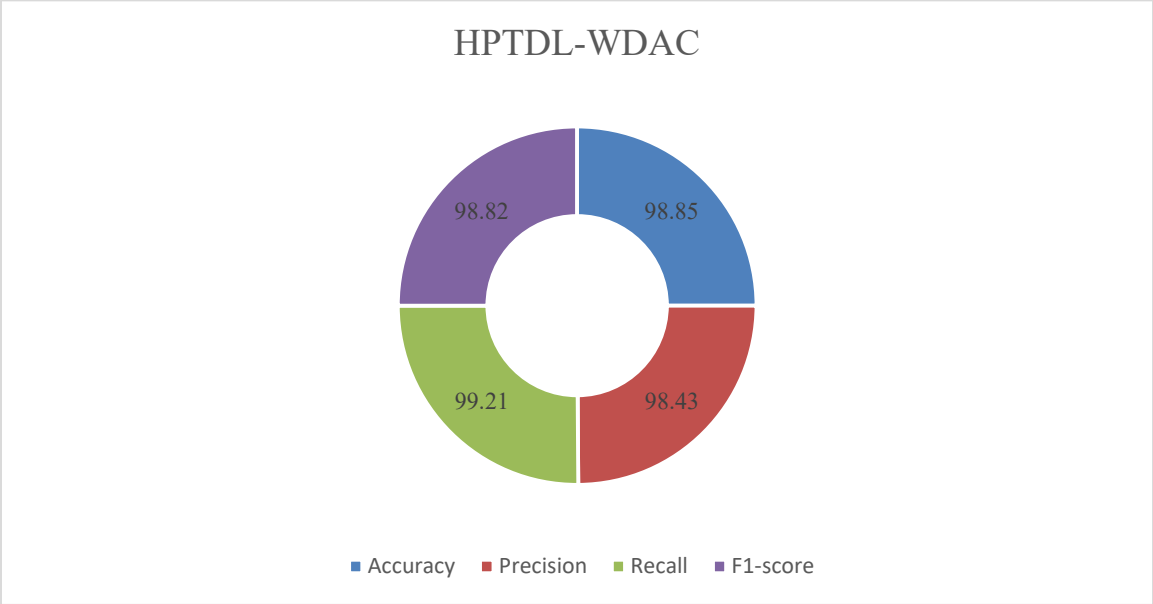


Figure 7: Result analysis of HPTDL-WDAC approach

To highlight the enhanced performance of the HPTDL-WDAC approach, a concise comparative analysis with recent models is presented in Table 2 and Figure 8. The experimental findings reveal that the Inception-v3, VGG-16, and RCNN models achieved accuracies of 89.67%, 93.85%, and 95.13% respectively. However, the HPTDL-WDAC model demonstrated superior performance, achieving a notably higher accuracy of 98.85%. These comprehensive results and subsequent discussion clearly demonstrate that the HPTDL-WDAC model outperforms other models by a significant margin.

Table 2: Comparison of HPTDL-WDAC technique with existing algorithms

Models	Accuracy	Precision	Recall	F1-score
Inception-v3	89.67	89.23	90.75	89.62
VGG-16	93.85	93.61	94.12	93.81
RCNN	95.13	94.92	95.97	95.08

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

HPTDL-WDAC	98.85	98.43	99.21	98.82

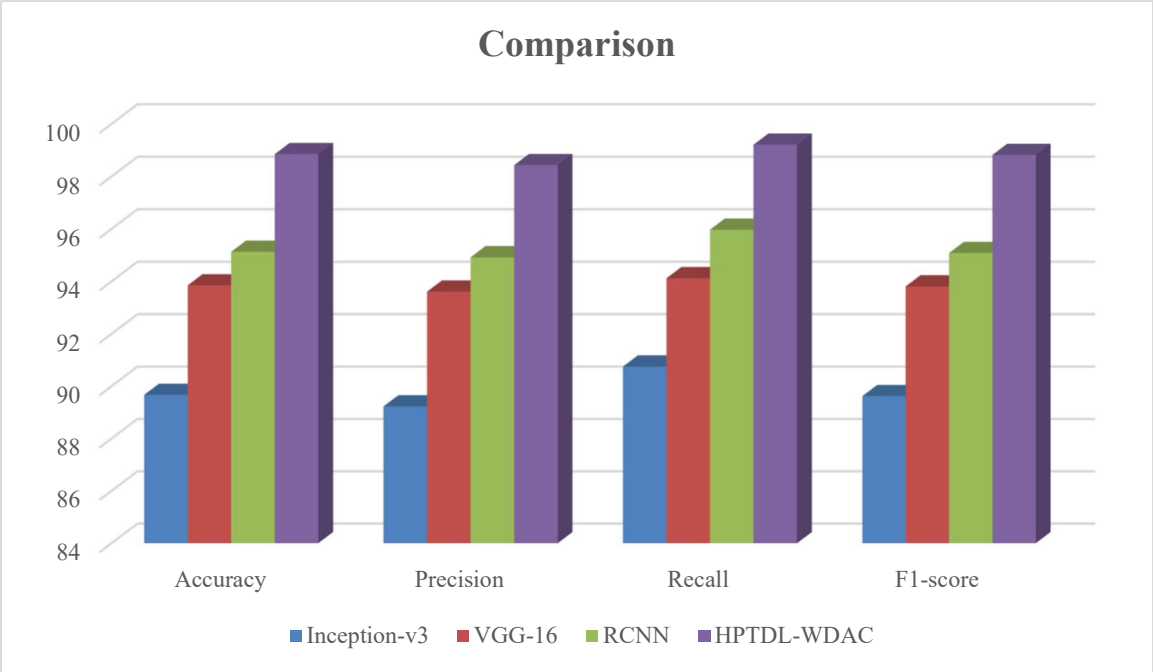


Figure 8: Comparative analysis of Proposed HPTDL-WDAC Model with Existing models

5. Conclusion

The study presented an advanced AI-powered weed detection system for precision farming, employing a series of sophisticated techniques to enhance the accuracy and efficiency of weed detection and classification. The pre-processing stage involved noise removal using a Gaussian Filter, which significantly improved the quality of the images by eliminating noise. Additionally, image resizing and class labelling were performed to standardize the inputs for the model. The core of the system is the RetinaNet model, renowned for its robust object detection capabilities. By leveraging the ADAM optimizer, the hyperparameters of the RetinaNet model were finely tuned, ensuring optimal performance. This comprehensive approach resulted in superior detection and classification accuracy, effectively distinguishing between plants and weeds. The empirical evaluation demonstrated that the proposed HPTDL-WDAC technique outperforms existing methods, such as Inception-v3, VGG-16, and RCNN, with a notable accuracy of 98.85%. The findings underscore the potential of integrating advanced deep learning models with optimized pre-processing techniques to address the challenges in precision agriculture. This system not only enhances weed detection but also contributes to reduced herbicide usage, promoting more sustainable farming practices. As part of the future study, the efficiency of the

proposed method can be increased by integrating this weed detection model with autonomous agricultural equipment such as drones or robotic weeding tools to automate the weeding process.

References

1. Shijian Yuan and Xiaobo Fan, "Developments and perspectives on the precision farming process for ultra-large size integrated components", International Journal of Extreme Manufacturing, vol. 1, no. 2, pp. 1-18, 2019.
2. David P. Horvath, Sharon A. Clay, Clarence J. Swanton, James V. Anderson, Wun S. Chao, "Weed-induced crop yield loss: a new paradigm and new challenges", Trends in Plant Science, vol. 28, no. 5, pp. 567-582, 2023.
3. Vishnu Moond, Narinder Panotra, Ashoka P., D. R. K. Saikanth, Gurinder Singh, N. Prabhavathi, and Badal Verma, "Strategies and Technologies in Weed Management: A Comprehensive Review", Current Journal of Applied Science and Technology, vol, 42, no. 29, pp. 20-29, 2023.
4. Z. Wu, Y. Chen, B. Zhao, X. Kang and Y. Ding, "Review of weed detection methods based on computer vision", Sensors, vol. 21, no. 11, pp. 3647-3668, 2021.
5. A.H. Al-Badri, N.A. Ismail, K. Al-Dulaimi, G.A. Salman, A. Khan, A. Al-Sabaawi, and M.S.H. Salam, "Classification of weed using machine learning techniques: a review—challenges, current and future potential techniques", Journal of Plant Diseases and Protection, vol. 129, no. 4, pp. 745–768, 2022.
6. Jingning Yu, "Based on Gaussian filter to improve the effect of the images in Gaussian noise and pepper noise", Journal of Physics: Conference Series, CONF-SPML-2023, pp. 2580, 2023.
7. R. Punithavathi, A. Delphin Carolina Rani, K. R. Sughashini, Chinnarao Kurangi, M. Nirmala, Hasmath Farhana Thariq Ahmed, and S. P. Balamurugan, "Computer Vision and Deep Learning-enabled Weed Detection Model for Precision Agriculture", Computer Systems Science & Engineering, vol. 44, no. 3, pp. 2759-2774, 2023.
8. N. Rai, Y. Zhang, B.G. Ram, L. Schumacher, R.K. Yellavajjala, S. Bajwa, and X. Sun, "Applications of deep learning in precision weed management: a review", Computers and Electronics in Agriculture, vol. 206, no. 1, pp. 107698, 2023.
9. Mohanad N. Alhasanat, Moath H. Alsafasfeh, Abdullah E. Alhasanat, and Saud G. Althunibat, "RetinaNet-Based Approach for Object Detection and Distance Estimation in an Image", International Journal on Communications Antenna and Propagation, vol. 11, no. 1, pp. 19341, 2021.
10. Mouna Afif, Riadh Ayachi, Yahia Said, Edwige Pissaloux and Mohamed Atri, "An Evaluation of RetinaNet on Indoor Object Detection for Blind and Visually Impaired Persons Assistance Navigation", Neural Processing Letters, vol. 51, no. 1, pp. 2265-2279, 2020.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

11. R. Kamath, M. Balachandra and S. Prabhu, “Paddy crop and weed discrimination: A multiple classifier system approach,” International Journal of Agronomy, vol. 21, no. 3 and 4, pp. 1–14, 2020.
12. N. Islam, M. M. Rashid, S. Wibowo, C. Y. Xu, A. Morshed et al., “Early weed detection using image processing and machine learning techniques in an Australian CHILLI FARM,” Agriculture, vol. 11, no. 5, pp. 387, 2021.
13. O.G. Ajayi, J. Ashi, and B. Guda, “Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images”, Smart Agriculture Technology, vol. 5, no, 1, pp. 100231, 2023.
14. A. N. Veeranampalayam Sivakumar, J. Li, S. Scott, E. Psota, A. Jhala et al., “Comparison of object detection and patch-based classification deep learning models on mid- to late-season weed detection in UAV imagery,” Remote Sensing, vol. 12, no. 13, pp. 2136, 2020.
15. Zhang S., Huang W., and Wang Z., “Combing modified Grabcut, K-means clustering and sparse representation classification for weed recognition in wheat field”, Neurocomputing, vol. 452, no. 10, pp. 665-674, 2021.
16. N. Al-Qubaydhi, A. Alenezi, T. Alanazi, A. Senyor, N. Alanezi, B. Alotaibi, M. Alotaibi, A. Razaque, A.A. Abdelhamid, and A. Alotaibi, “Detection of unauthorized unmanned aerial vehicles using YOLOv5 and transfer learning”, Electronics, vol. 11, no. 17, pp. 2669, 2022.
17. I. Gallo, A.U. Rehman, R.H. Dehkordi, N. Landro, R. La Grassa, and M. Boschetti, “Deep object detection of crop weeds: performance of YOLOv7 on a real case dataset from UAV images”, Remote Sensing, vol. 15, no. 2, pp. 539, 2023.
18. Mohd Anul Haq, “CNN based automated weed detection system using uav imagery”, Computer Systems Science and Engineering, vol. 42, no. 2, pp. 837-849, 2022.
19. J. Chen, H. Wang, H. Zhang, T. Luo, D. Wei, T. Long, and Z. Wang, “Weed detection in sesame fields using a YOLO model with an enhanced attention mechanism and feature fusion”, Computers and Electronics in Agriculture, vol. 202, no. 1, pp. 107412, 2022.
20. X. Jin, T. Liu, Y. Chen, and J. Yu, “Deep learning-based weed detection in turf: a review”, Agronomy, vol. 12, no. 12, pp. 3051, 2022.
21. M.H. Saleem, J. Potgieter, and K.M. Arif, “Automation in agriculture by machine and deep learning techniques: a review of recent developments”, Precision Agriculture, vol. 22, no. 1, pp. 2053–2091, 2021.
22. S.J. Rani, P.S. Kumar, R. Priyadharsini, S.J. Srividya, and S. Harshana, “Automated weed detection system in smart farming for developing sustainable agriculture”, Int. J. Environmental Science and Technology, vol. 19, no. 9, pp. 9083–9094, 2022.
23. T.A. Shaikh, T. Rasool, and F.R. Lone, “Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming”, Computers and Electronics in Agriculture, vol. 198, no. 1, pp. 107119, 2022.
24. https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes