



Research Article | Open Access |

A Deep Learning Framework for Smart Agriculture: Real-Time Weed Classification Using Convolutional Neural Network

Sushilkumar S. Salve,* Sourav S. Chakraborty, Sanskar Gandhewar and Shrutika S. Girhe

Department of Electronics and Telecommunication, Sinhgad Institute of Technology, Lonavala, Maharashtra, 410401, India

*Email: sushil.472@gmail.com (S. S. Salve)

Abstract

Agricultural sector being the foundation of food supply and raw material production contributes significantly to the GDP growth and value chain. Thus, the effective elimination of weed in modern day agriculture is essential as the current world scenario demands for efficient and resourceful ways for crop cultivation and harvesting. The urgent need for elimination of weeds arises due to their tendency to extricate all the essential minerals and moisture that the crops require for their appropriate growth. The main objective of this study is to successfully acquire live video feed as input, classification into categories of crop, weed and none. Finally, upon detection of weed the spraying mechanism releases a pre-determined amount of herbicide upon the weed. A total of 5471 image samples were captured to train the CNN model. The prototype mentioned in this paper uses Convolutional Neural Network (CNN) technique for feature extraction, Fully Connected Layers or Dense Layers (FCLs) for classification using SoftMax as the activation function respectively. The activation function also here is being used to remove all negative (less significant) values. Also, a comprehensive comparison was made between the CNN and YOLOv4 technique and performance parameters of both were evaluated. The CNN technique achieved an accuracy of 95.50% whereas YOLOv4 achieved 91.00%. Finally, the F1 Score was evaluated to be 96.25% and 91.96% respectively. Compared to existing models, our prototype demonstrated higher accuracy and real-time adaptability in field conditions, proving suitable for autonomous weed management systems. Unlike earlier systems that depended mostly on stored images or fixed datasets, our approach stands out by using a live video feed to identify weeds in real-time. It's built on a mobile platform that can automatically spray herbicides, making precision farming possible without the need for constant human supervision.

Keywords: Computer vision; Convolutional neural network; Deep learning; SoftMax; Max pooling; Weed detection; Image pre-processing.

Received: 05 April 2025; Revised: 15 May 2025; Accepted: 27 May 2025; Published Online: 30 May 2025.

1. Introduction

In crop fields, weeds are naturally occurring plants that compete with crops for vital resources like space, light, moisture, and air, which may lower crop yield. Effective weed control is essential during cultivation because they impede crop growth.^[1] Farmers may experience lower yields and financial losses as a result of weeds competing for resources with cash crops. The impact of weeds varies depending on the crop type and the farm's geographical

location.^[2] Weeds can reduce yield by up to 34% if they are not controlled, whereas animal pests and diseases cause yield loss of 18% and 16%, respectively. Weed infestations can result in crop losses of roughly 23% to 44% in typical crop fields.^[1] Simultaneously, the agricultural sector is under pressure to achieve steadily rising yields as the demand for more food production rises at the same time.^[3] This emphasizes how precision farming and robotics are necessary to increase yield while lowering dependency on

conventional farming practices. Modern technology has made it possible for autonomous machines to carry out agricultural tasks effectively. High-quality crops can be produced with little human labor when robotics and intelligent machinery are integrated into agriculture.^[3,4]

A weed detection system uses machine learning algorithms to identify unwanted plants in an agricultural field. Farmers can reduce their use of weed and herbicides, which can be harmful to the environment and public health. Plans for targeted weed control can be created by utilizing the information on the types of the weeds that the detecting system can supply.^[5] A new technology that has the potential to completely transform agriculture is machine learning-based weed detection. The system's purpose is to locate and identify weeds in a field so that farmers can take specific action to get rid of them, gather live videos and photos of a field, apply machine learning techniques to the same, and then determine the weeds. Numerous methods, such as object detection, feature extraction, segmentation, and Classification, can be used to complete this process. We decided to use a live feed CNN technique to address this problem it's more like analyzing the input dataset to find the weeds.^[5,6]

The weeds within rows might not be accurately removed by conventional machinery. Sunil G C *et al.* emphasized while introducing their study on the thought that the herbicide which is sprayed uniformly across the field, treating weeds and crops alike, at a set pace when compared to site-specific herbicide applications prove less feasible as blanket herbicide applications may have a more negative impact on the ecosystem. As a result, applying an herbicide selectively to areas of concern may improve precision while lowering input costs and environmental problems.

Umamaheswari S *et al.*^[7] mentioned about the field of robotic farming and precision agriculture that needs to advance in response to current problems with the lack of agricultural labour and resources, the emergence of new crop diseases, and weeds. The issues of climate change and sustainable agriculture are intimately tied to the challenge of effective weed classification and detection. According to various resources and findings the study suggests, existing species may be exposed to new and hybrid weeds as a result of climate change. Because weeds can hinder the growth of farm crops, it is crucial to create new technologies that aid in identifying them. Identifying weeds can also help remove them, which lowers the need for pesticides and offers effective substitutes when the crops are harvested.

O. M. Olaniyi *et al.*^[8] mentioned about the various ways of weed eliminations as people have become more civic and knowledgeable about weeds, experts have been looking for ways to eradicate the infamous pest with the least amount of harm to the plant. The three main strategies for controlling weeds are cultural, chemical, and automated approaches. Bush fallowing, mulching, fire clearance, early flooding, hand weeding, shifting crops, and maintaining a clean reaper

are all components of the cultural approach of weed management. This approach has significant labour costs and drawbacks. Applying herbicides is thought to be a significant alternative to hand weeding. However, excessive herbicide use can result in harvest losses, harm to the environment, high production costs, and the development of herbicide resistance. Without getting to the weeds, some of these pesticides even wind up on the soil and food crops. Since spraying food crops is viewed as a risk to the safety of the food being consumed, a thorough weed control method is necessary.

On the other hand, as specified by P. Kavitha Reddy *et al.*,^[5] deep learning techniques particularly those that use neural networks have become increasingly popular in recent years. These methods use big datasets of tagged images to train and intricate neural network models. The neural network automatically collects pertinent information and classifies the input photos using iterative learning procedures. The YOLO algorithm is a well-known implementation of the convolutional neural network (CNN), which is the foundation of deep learning techniques in computer vision (CV).

In this paper, a low cost and robust weed detection, live video-based and elimination system with automated spraying using Convolutional Neural Network (CNN) as the main computing algorithm, SoftMax and ReLU as activation functions and classification of the same using Fully Connected Layers (FCLs) is given along with a detailed comparison of YOLOv4 with the proposed method.

The major reason for why CNN was selected is due to its ability to focus on fine-grained feature learning, especially useful in identifying small or overlapping weed patterns. YOLOv4 was chosen for comparison due to its real-time detection speed. Other models like Faster R-CNN or ViT were not used due to higher computational demands unsuitable for edge deployment on Raspberry Pi 4. The two activation functions SoftMax and ReLU were selected for their simplicity, speed, and established use in CNN architectures. Alternatives like Swish or Leaky ReLU can improve performance but require higher computational cost and tuning.

2. Materials and methods

This particular section describes the materials and design required for the successful development of the particular proposed system. Here a detailed overview of the components, methodology utilized and many other specifications are mentioned. The system prototype well integrates the combination of Internet of Things (IOT) with image processing, feature extraction, deep learning algorithm and identification along with precision spraying unit.

2.1 System overview

The proposed system is implemented using Convolutional

Neural Network to develop and cultivate a robust, multi-scalable and versatile weed detection system that produces accurate results in real time using live video feed via a webcam. The input dataset then goes through various processes and at the end determines the result based on three particular parameters i.e., i) weed, ii) crop, iii) none. The various processes particularly include, Image Acquisition, feature extraction, classification and training of the model.

A generalized block diagram is represented as Fig. 1 that gives an idea regarding the actual flow of the components within the proposed system, and their particular task involved in the accurate execution. The proposed prototype contains various components mounted on a robust wooden platform which are powered by a 12V DC adapter.

The main microcontroller unit i.e., Raspberry Pi 4 Model B is powered by a 5V USB-C type charger. The output can be observed on a desktop monitor via connection with an HDMI cable. Fig. 2 Carefully shows stage by stage deployment and implementation of a particular CNN based weed detection system using Max Pooling, ReLU, Dropout, Fully Connected Layers (FCLs) and SoftMax for multiple stages of detection and processing of the input dataset.^[9,10]

2.2 Working principle

2.2.1 Hardware

The “A Deep Learning Framework for Smart Agriculture: Real-Time Weed Classification Using CNN” uses a robust and sturdy navigable prototype that enables the system to be mounted of a hard bound wooden base with a four-wheel chassis. The two forward wheels are attached with two 12V DC geared motors of 300 r.p.m each and the two rear wheels are attached as dummy wheels for support. As they support heavier load i.e., in this case a wooden platform 12V DC geared motors are used. These motors are then connected to an L293D module. This L293D module is a motor driver module which is widely used in embedded systems to control the direction of DC motors and stepper motors. This module is capable of driving two DC motors independently in both forward and reverse direction. This adds precision and control to the whole system and grants mobility across the

field. Both the L293D module and the DC motors are powered using a 12V DC power supply. The L293D is also interfaced with the Raspberry Pi 4 model B as the master control unit.

A Bluetooth module i.e., HC-04 is also interfaced to the microcontroller for controlling the directions provided by the motor driver module. This Bluetooth module supports V2.0+ EDR (Enhanced Data Rate) up to 3 Mbps modulation along with 2.4 GHz radio transceiver and baseband. A python program is being compiled and executed by the microcontroller that enables the user to connect with the Bluetooth module using the application “Serial Bluetooth Terminal”, where user can give commands in the form of numbers for specifying movements in specific direction (i.e., 1 = forward, 2 = reverse, 3 = left, 4 = right, 5 = terminate).

A single-channel relay module is also being used and interfaced with the microcontroller in order to control the fluid pump inside the sprayer prototype. It is rated for switching up to 10A at 250V AC or 24V DC. This is also powered using the 12V DC power supply which was used to power the motors and driver module. Now in this relay when the input (IN) is driven LOW, at that time the relay coil energizes and it switches the normally closed (NC) contact point to normally open (NO) contact. This action effectively contributes in turning a connected device (i.e., the fluid pump) on or off at specified intervals upon weed detection. The camera module used in the particularly developed system is the Xiaomi Mi HD USB 2.0 Web-Cam. It can capture live video feeds up to a resolution of 1280×720p HD and has a frame rate of 30 FPS. With up to a 90° wide angle field of view it has no driver requirements, thus compatible with the Raspberry Pi 4 microcontroller. The OpenCV library efficiently helps the prototype to capture and process the live feed of input dataset for image pre-processing.

The Raspberry Pi 4 model B microcontroller acts as the heart and brains of the system. This is basically a card sized mini-computer that operates using its own software, performing tasks that an actual desktop can perform independently including browsing, media playback and major IOT development. It has a 4 GB LPDDR4-3200

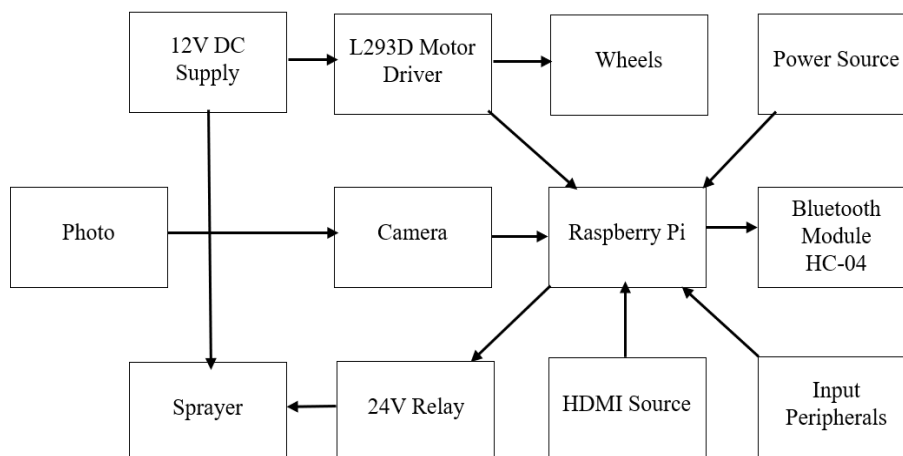


Fig. 1: Block diagram of proposed prototype.

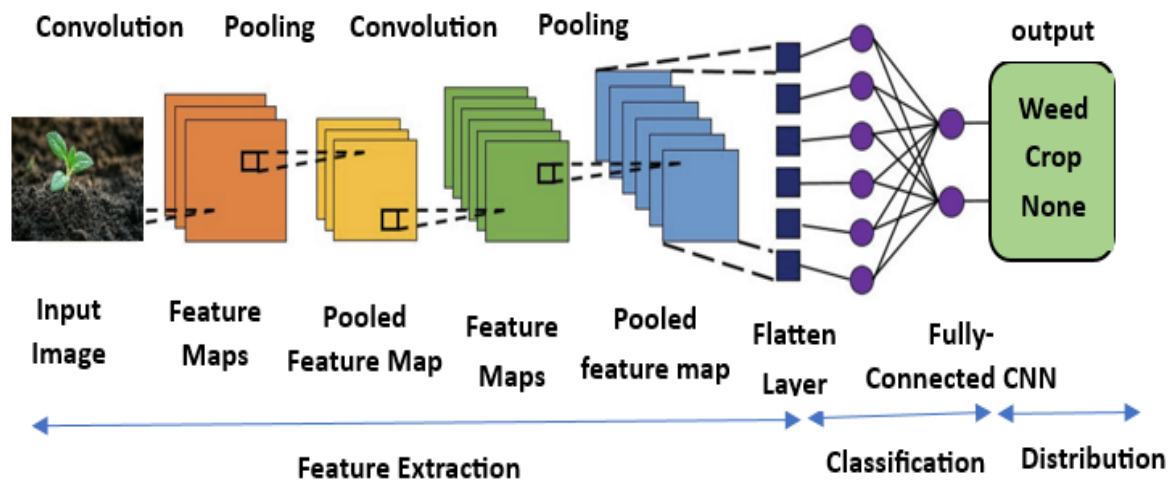


Fig. 2: Schematic of the proposed system overview.

SDRAM and has a microSD card slot that comprises of the actual controller software. It consists of four USB ports two with USB 3.0 capabilities and the other two with USB 2.0 support. Two micro-HDMI slots are provided for interfacing with external display peripherals supporting resolutions up to 4K 60 FPS. The power supply is provided via a 5V DC USB-C type connector, and has an ambient operating temperature within the range of 0°C to 50°C.

Fig. 3 gives a glimpse of the proposed system that is being cultivated and developed for the comprehensive study of

both the algorithms. The image gives a clear idea about all the particular components and their whereabouts in the particular model.

2.3 Software

In regard to the proposed model in this study, Debian GNU/Linux 10 (buster) has been installed onto the Raspberry Pi 4 Model B as its operating system which uses Python IDLE as compiler to script and execute the python code for the implementation and training of CNN and YOLOv4

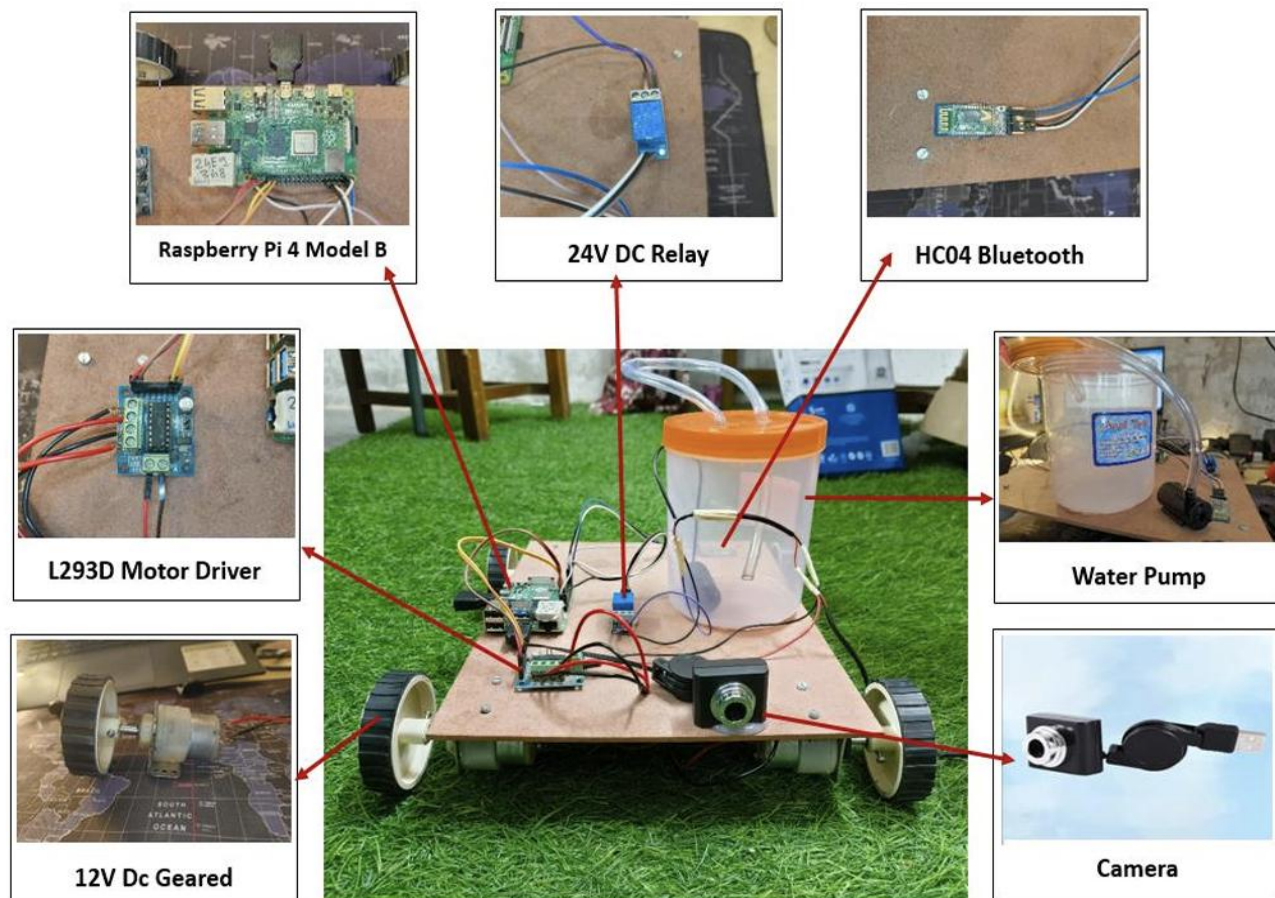


Fig. 3: A schematic representation of the proposed prototype and its components.

models. Various open-source python libraries like OpenCV and TensorFlow have also been implemented in the same to facilitate top-notch image processing and deep learning model implementations for accurate classification and detection of weed in farms and agriculture fields.

The software used in this study is adequate to support latest hardware components like camera modules and other hardware peripherals that are essential for the proper working of the system and its overall performance. A personalized dataset of varied images was curated that ensured the model was trained based on images of crops and weeds of various ethnicity portraying varied lighting, backgrounds and crop types are present. Augmentations such as flip, rotate, crop and brightness change were also used.

2.4 Implementation

2.4.1 Image Acquisition

The input data first is captured using a Xiaomi USB 2.0 HD webcam that supports capturing video datasets up to 720p and a frame rate of 30 frames per second (fps). This input data then undergoes image pre-processing, where the pixel values originally ranging from 0 to 255 are normalized to a scale of $\{0, 1\}$. Upon normalization, the performance of the CNN model improves ensuring better numerical stability and faster convergence. The input data also undergoes grayscale conversion as weed detection relies more upon shapes and textures than colour.^[11] Fig. 4 accurately helps us imagine how colour images are converted to grayscale for the model. The system is made more efficient by resizing the data to 64×64 pixels thus reducing the image size and lowering the computational cost.^[12,13]

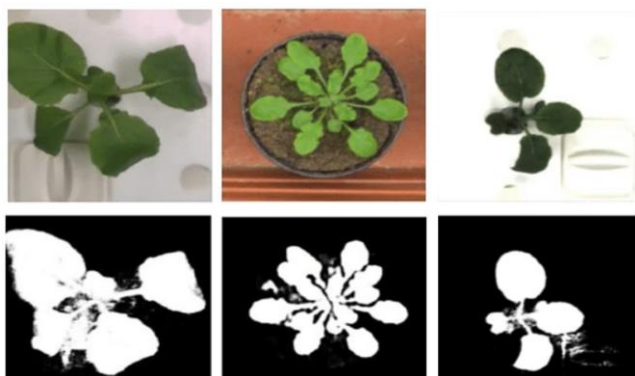


Fig. 4: Grayscale conversion of input dataset.

2.5 Feature extraction

Now features are being extracted from the pre-processed image using 2D Convolution that extracts out all important features and patterns like edges *etc.*^[14] The CNN model consists of 4 convolutional layers, each with 32 filters of size 3×3, followed by ReLU activation and 2×2 max-pooling. The input images are grayscale with a resolution of 64×64 pixels. The second convolutional layer again applies 64 filters of the same size. Rectified Linear Unit (ReLU) here acts as the activation function which converts the negative values to zero thus introducing non-linearity.

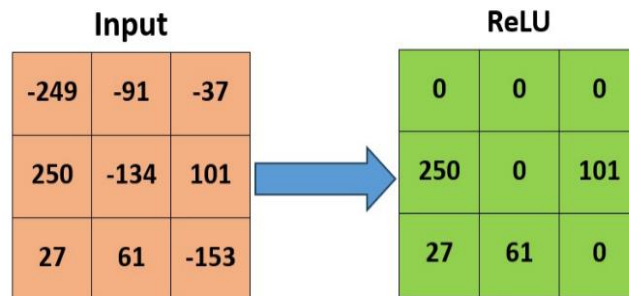


Fig. 5: A demonstration of the rectified linear unit.

The particular non-linearity introduced by the ReLU activation function allows the CNN network to learn more complex patterns and functions that are beyond the linear relationships. This makes the network computationally more efficient as fewer neurons activate at once, improving generalization, acting as simple threshold functionality. When its compared to other functions such as sigmoid/tanh *etc.*, it avoids expensive exponentials thus facilitating faster convergence rates during training of the network and helping the gradients remain significant during backpropagations. Fig. 5 shows how the negative inputs are converted into zero's thus introducing sparsity in the activations.^[15]

Equation (1) shows the mathematical representation of ReLU as an activation function.^[16]

$$f(x) = \max(0, x) \quad (1)$$

where,

if $x > 0$, then $f(x) = x$

or else $x \leq 0$, then $f(x) = 0$

The above equation is observed to be common in most of the studies as it being a very generalised equation here particularly showing how the function actively converts negative value inputs into zero's and keeps the positive ones unaffected.

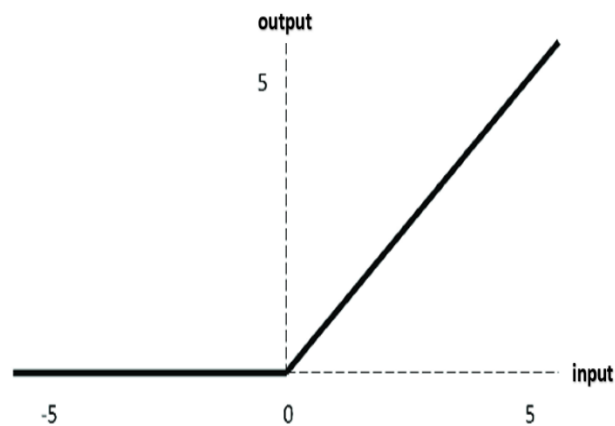


Fig. 6: Graphical representation of ReLU.

Fig. 6 demonstrates the how the activation function looks like when plotted between two axis. However, when its limitations are taken into account, some neurons might give output as zero and never get activated. But never the less, the

function has proven its efficiency and reliability even after considerations of its drawbacks.

Max Pooling and Dropout are also being used as Max Pooling reduces the image spatial dimensions while preserving the essential features and the Dropout reduces the overfitting by randomly setting 25% of the neurons to zero during training procedure. A window of 2×2 size moves all over the feature map, thus keeping only maximum value from each window. This particularly contributes in reducing computational complexity.

Max Pooling in CNN is basically a down sampling technique which proves extremely beneficial in reducing spatial features and dimensions of an input volume dataset. It is non-linear in nature that serves for better efficiency and reduced computational power. It operates independently on each and every depth slice of the input image and resizes it spatially. It involves sliding a window called kernel of size 2×2 across the input data and performing matrix multiplication taking only the maximum values from each frame. Fig. 7 shows accurately the same using a set of sample values.

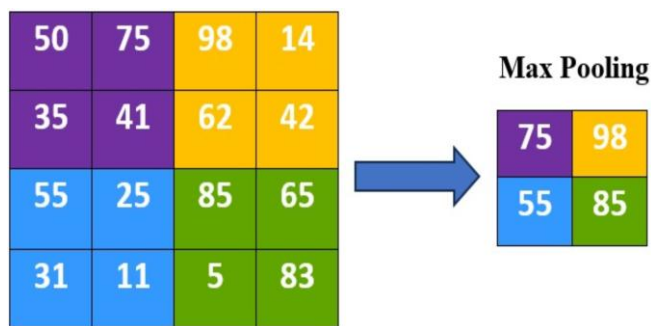


Fig. 7: Max pooling in CNN.

These particular maximum values then constitute a single pixel in the newly pooled output. The 2×2 window that moves all over the input image follows a particular stride of a certain number of pixels. This particular process when repeated until the final output produces an output image of size almost half the original and effectively reduction in pixels by 75%.^[15]

Now while training a neural network, it might not only learn the general pattern but also the noise and specific ungeneralised details of unseen data. This overfitting might

give higher accuracy while training the model with data set but will produce low accuracy in the testing procedures, thus leaving a large gap between the training and testing accuracy. The Dropout technique effective in such cases as during the training process it randomly removes a small fraction of neurons in the network, in our case 25%, 50% and 80% for different layers, so the dropout rates were set at 0.25, 0.5, and 0.8 respectively.

In mathematical terms,^[17] a mask is being applied to a set of neurons according to the percentage of dropout applied during the training period. At each step a mask matrix is generated where each entry is in form of a binary variable i.e., 0 or 1 indicating which neuron to be dropped or not.

$$y = W \cdot (M \odot x) \quad (2)$$

where,

x = input to a layer

W = weight matrix for particular layer

M = mask matrix

\odot = element - wise product

With dropout, the mask matrix particularly applied, where each element of M is '0' having probability of p and '1' with probability $1 - p$. During testing the dropout is called off but the weights are scaled by $1 - p$ to take account of the neurons that were dropped off during training process.

The layers are being employed where each layer detects more and more complex patterns in the input images. These higher-level features include shapes, edges, textures. Pooling of such layers helps the model to recognize the objects regardless of their position in an image thus making the model translation-invariant. The First Dropout layer introduces early regularization in the dataset, preventing co-adaptation of the neurons and encouraging increased robust feature learning. The Flatten layer now converts these multi-dimensional feature maps into 1-D vector for better transition into the dense layers (FCLs). The Second Dropout layer again randomly drops units from the flattened layer before its transition into the dense layers giving more regularization which were prone to overfitting due to their large number of parameters.

2.6 Classification

After the successful extraction of features from the input

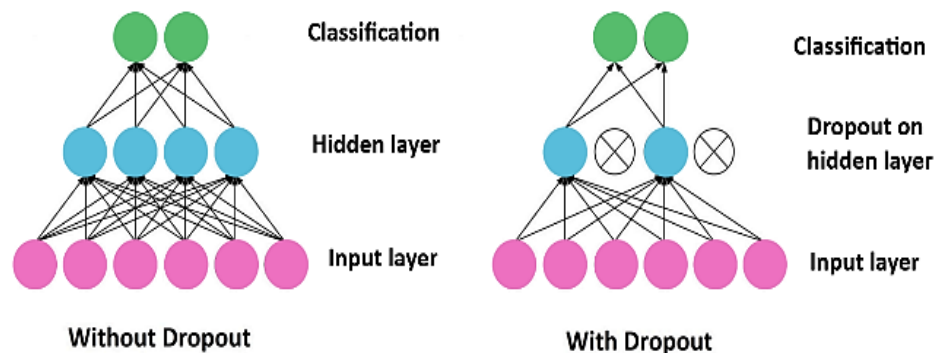


Fig. 8: Dropout in CNN.

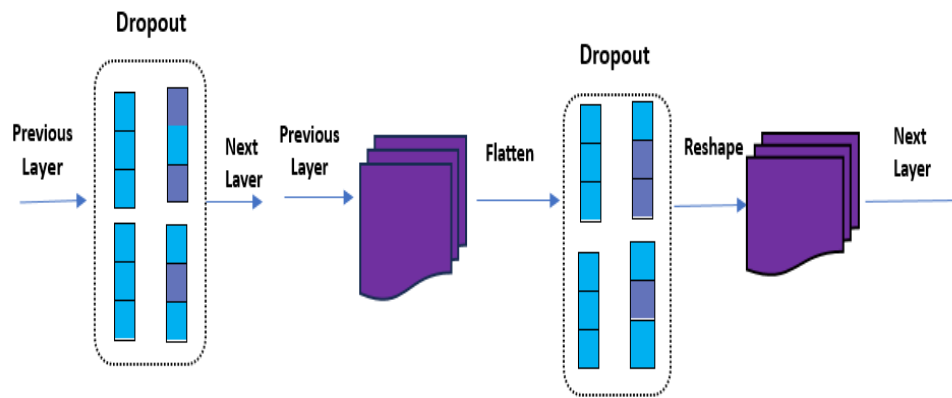


Fig. 9: Various dropout layers in CNN.

images, the model network now flattens the image dataset into a 1D vector and feeds to the Fully Convolutional Layer (FCL) as it accepts only one-dimensional input.

E.g.

MaxPooling2D output = (7, 7, 64)

Equivalent 1D vector output = $(7 \times 7 \times 64) = (3136)$

The dense layer comprises of 1024 neurons that acts as a hidden layer processing extracted features from previous CNN layers. In the output layer of the FCL, three neurons are taken that denote three possible classes. Here the SoftMax activation function is used that converts the output into probabilities whose sum results to 1. It is basically a mathematical function which is majorly used in cases involving multiple classes, where vector of real numbers (logits) is converted into probability distribution, where the values are in the range of 0 and 1. In [18] Brahim Jabir *et al.* accurately depicted and visualized how the hidden layers in a fully connected dense layer interact with one another and work accordingly. The CNN consisted of 3 convolutional layers with filter sizes (3×3) , (3×3) , and (5×5) respectively, followed by ReLU activations and MaxPooling.

Mathematically, Eq. 3. Accurately shows the working of SoftMax activation function for precise model prediction and detection.^[19]

$$P_i = \frac{e^{Z_i}}{\sum_{j=1}^n e^{Z_j}} \quad (3)$$

where,

e^{Z_i} = Exponential of input Z_i (Raw Score)

$\sum_{j=1}^n e^{Z_j}$ = Sum of exponentials of all inputs

Here, P_0 indicates probability of crop, P_1 indicates probability of weed and P_2 indicates probability of none. The class with highest probability is the model's prediction.

Whereas in YOLOv4, the classification techniques are directly including into the object detection process. Originally this method is ideal for real-time object detection but in this paper, we have proposed a different approach to utilize CNN for real-time object detection and training. YOLOv4 algorithm performs localization by detecting the position of an object and classification by object type identification in a single forward pass using the neural

network. Out of the three major components of the YOLOv4 network (i.e., backbone, neck, head), the head network is responsible for classification and final detection. It basically applies anchor boxes on the feature maps and generates the output with particular probabilities of the classes.^[20,21]

The process initiates with an input image of size 416×416 , where multiple detection heads of different scales are being used. The feature maps are of sizes 13×13 , 26×26 and 52×52 . If S = Grid Size, B = Number of anchor boxes per grid cell and C = Number of classes, then the tensor output for each scale shape would be:

$$S \times S \times B \times (5 + C) \quad (4)$$

where,

$5 = 4$ bounding box coordinates $(t_x, t_y, t_w, t_h) + 1$

C = Class probabilities

In Equation (4), the output tensor of YOLOv4 has been calculated. Now the bounding box offsets relative to the anchor boxes are being predicted. If t_x, t_y are predicted offsets for box center and t_w, t_h are the predicted offsets for width and height. So, to calculate the actual box predictions, the equations would look like:

$$b_x = \sigma(t_x) + c_x \quad (5)$$

$$b_y = \sigma(t_y) + c_y \quad (6)$$

$$b_x = p_w \cdot e^{t_w} \quad (7)$$

$$b_h = p_h \cdot e^{t_h} \quad (8)$$

where,

σ = Sigmoid function

(c_x, c_y) = Top-left coordinate of grid cell

(p_w, p_h) = Width and height of the anchor box

Now, when we actually step into the probability distribution analysis over all the classes, we use the SoftMax activation function here as well for independent multi-labelled classification. Equation (3) shows the SoftMax implementation of CNN as well as YOLOv4. But when we go with sigmoid for binary per class classification, the equation looks like:

$$P(c_i | object) = \sigma(t_{c_i}) \quad (9)$$

$$\text{Confidence}(c_i) = P_{obj} \cdot P(c_i | \text{object}) \quad (10)$$

Equation (10) denotes the final confidence probability for the class c_i .^[22]

2.7 Training the model

A large dataset of photos from agricultural fields are gathered and pre-processed in order to train the proposed prototype. These photos usually show different kinds of weeds and crops in a variety of backgrounds, lighting, and environmental settings. In order to create labelled data for supervised learning, the photos are tagged to differentiate between weed and non-weed areas. To enhance model generalization, the dataset is then enhanced using methods including flipping, rotation, scaling, and colour changes. To guarantee balanced learning and assess performance at various phases, the pre-processed data is separated into training, validation, and test sets.^[23] Training was performed with batch size of 32, 50 epochs, Adam optimizer (lr = 0.001), and categorical cross-entropy loss function.

After the dataset is ready, a deep learning model based on convolutional neural networks (CNNs) is trained to identify and categorize weeds. Using an optimizer like Adam or SGD, the model minimizes a loss function, usually cross-entropy, during training to identify patterns and characteristics that differentiate weeds from crops. The output layer predicts class probabilities using SoftMax activation. Metrics such as F1-score, recall, accuracy, and precision are used to track the model's performance. Using edge devices or mobile applications, the top-performing model is chosen after multiple epochs based on validation performance and then used for real-time weed detection and control in the field.

2.8 Testing of model

After the model's training and validation, the testing phase commences. A different test dataset with previously unseen photos is used to assess the trained model. This aids in evaluating how well the model generalizes to fresh, actual data. To determine performance metrics like accuracy, precision, recall, and F1-score, the model's predictions are contrasted with the actual labels. These measures reveal the model's ability to discriminate between weeds and crops, particularly under difficult circumstances like changing lighting, occlusions, or background noise. Any incorrect classifications are examined to find trends or particular instances where the model might be having trouble.^[24]

The model is tested offline as well as in real time in the field using Raspberry Pi 4 Model B. In this stage, the model is fed live video input, and the accuracy of the weed detection and localization is monitored. To make that the system functions well in real-world situations, its response speed, effectiveness, and dependability are tracked. The model is connected to an automated weed-removal sprayer, that performs reliably and accurately. Additionally, field testing offers insightful input for retraining or additional model refinement to increase resilience.

3. Results and analysis

3.1 Performance evaluation metrics

The proposed prototype in this paper is being evaluated and judged on the basis of the following performance evaluation parameters. These parameters are found out after conducting multiple number of experiments and epochs upon considerations in regard to various factors and scenarios to ensure overall accurate analysis of the performance of the system.

3.1.1 Accuracy

Accuracy of a system is basically the ratio of positively predicted results to the total number of observations done. Equation (11) shows how accuracy is being calculated using following parameters where, the numerator accounts for all the predictions that the model got correct and the denominator denotes all predictions that were made.^[25]

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

where,

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

Here True Positive is referred to the case when the object to be detected is actually weed and the system positively classifies it as a weed whereas, True Negative is the case when the class was not a weed, and the prototype accurately classifies it as not a weed.

Now when it comes to the false detections and scenario's we have parameters like False Positive and Negative respectively. False Positive is the case when the model positively classified it to be a weed but in actual it was not a weed class and False Negative is the case when the model classifies the object as not of a weed class, in practical it belonged to the weed class.

3.1.2 Precision

Precision in deep learning is a performance evaluation metric the basically evaluates the quality and correctness of the accuracy parameter i.e., positive classifications by the model.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

Equation (12) shows how precision of a model is being calculated on the basis of True Positives and False Positives. As here we seek to determine the actual correctness of a model, hence this only considers positive classification scenario's where the prediction is always right. But it also shows a major drawback by not the negatives at all, that might cause the model to miss certain correct predictions (i.e., low recall).^[26]

3.1.3 Recall

Within this performance evaluation parameter, we check in

actual how many cases did the model actually classified positively out of all the positive ones. It ranges from 0 to 1. This basically measures the model's real ability to capture all the relevant instances of the positive class.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (13)$$

Equation (13) answers to our question, "Out of all the actual weeds, how many did our model find?" If a model has higher recall, then we can safely say that the model is classifying most of the positive classes, hence maximum weeds in the field of crops are being successfully detected.

But if the recall alone is too high, that would mean that the model is classifying every object as weed, thus making the recall of the model 100% but reducing precision in its classification which accounts to be a failure in the model's classification.^[27]

3.1.4 F1 score

This parameter is solely based on the values of precision and recall of the particular model as it is a harmonic mean of the precision and recall of the model. It ranges in between 0 (worst) and 1 (best). This metric is the one that gives us a trade-off between the precision and recall of a particular model. As the harmonic mean is observed to punish the extreme high resulting values more, thus this is preferred over arithmetic mean process. As a result, both precision and recall values have to be above the mark in order to achieve a reasonably higher F1 score.

$$F1 \text{ score} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

Equation (14) shows how mathematically F1 score is being calculated using the precision and recall metric values. It is especially used in cases where a particular model has an imbalanced dataset or cases where the model needs to have a proper balance between precision and recall.^[28]

3.2 Experimental analysis

3.2.1 Metric values

The proposed prototype in this paper is trained and developed using a standard self-developed dataset. The

prototype was being implemented using CNN as well as YOLOv4 deep learning algorithms and after successful testing phase, the results have been concluded and compiled according to the above defined performance evaluation metrics.

The results of each technique have been thoroughly evaluated ensuring untampered standards and accurate real-world simulation. Table 1 summarizes the result metrics of YOLOv4 technique that was implemented on the very same setup for a through comparison.

Using Equation (11) we can calculate the value of accuracy as follows:

$$\text{Accuracy} = \frac{103 + 79}{103 + 79 + 6 + 12} = 91.0\%$$

Similarly, using Equation (12) and (13) the precision and recall are calculated:

$$\text{Precision} = \frac{103}{103 + 6} = 94.49\%$$

$$\text{Recall} = \frac{103}{103 + 12} = 89.56\%$$

Now, Equation (14) is being used to calculate the F1 Score for the particular technique:

$$F1 \text{ Score} = \frac{2 \times 0.9449 \times 0.8956}{0.9449 + 0.8956} = 91.96\%$$

Table 1: Results during field testing using YOLOv4.

Field Trial	True Cases		False Cases		%Error	%Success
	TN	TP	FN	FP		
1	8	9	3	0	15	85
2	6	11	3	0	15	85
3	3	16	0	1	5	95
4	10	10	0	0	0	100
5	7	11	2	0	10	90
6	11	9	0	0	0	100
7	8	11	1	0	5	95
8	6	9	3	2	25	75
9	14	6	0	0	0	100
10	6	11	0	3	15	85
Total	79	103	12	6	-	-
Average	-	-	-	-	9.0	91.0

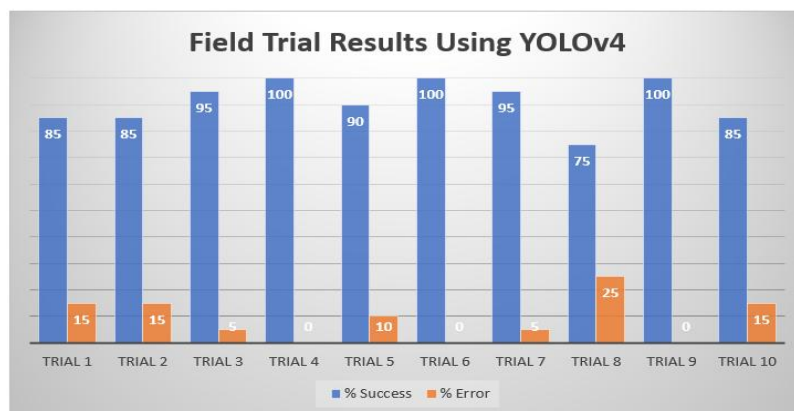


Fig. 10: Graphical representation of YOLOv4 results.

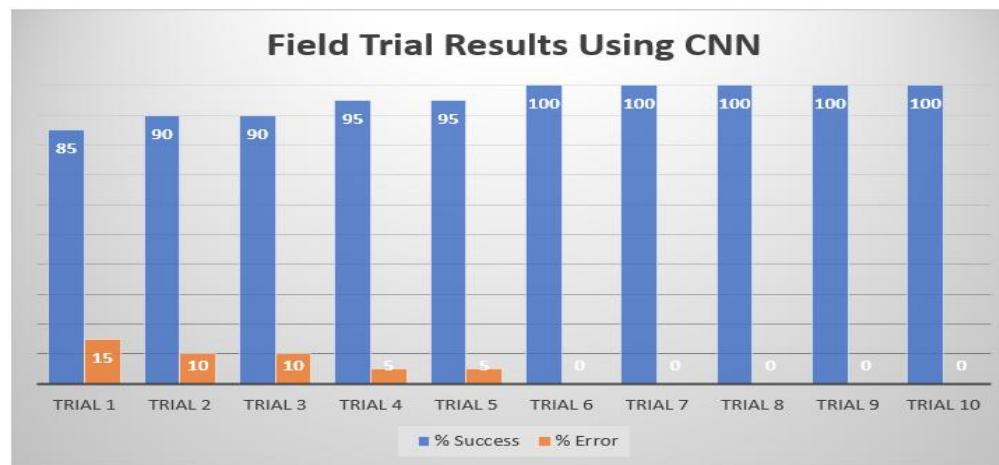


Fig. 11: Graphical representation of CNN results.

Table 2 summarizes the result metrics of CNN technique that was implemented on the very same setup for a through comparison.

Table 2: Results during field testing using CNN.

Field Trial	True Cases		False Cases		%Error	%Success
	TN	TP	FN	FP		
1	6	11	2	1	15	85
2	6	12	2	0	10	90
3	6	12	0	2	10	90
4	3	16	0	1	5	95
5	7	12	1	0	5	95
6	11	9	0	0	0	100
7	8	12	0	0	0	100
8	4	16	0	0	0	100
9	14	6	0	0	0	100
10	10	10	0	0	0	100
Total	75	116	5	4	-	-
Average	-	-	-	-	4.5	95.5

$$Accuracy = \frac{116 + 75}{116 + 75 + 4 + 5} = 95.50\%$$

Similarly, using Equation (12) and (13) the precision and

recall are calculated:

$$Precision = \frac{116}{116 + 4} = 96.66\%$$

$$Recall = \frac{116}{116 + 5} = 95.86\%$$

Now, Equation (14) is being used to calculate the F1 Score for the particular technique:

$$F1\ Score = \frac{2 \times 0.9666 \times 0.9586}{0.9666 + 0.9586} = 96.25\%$$

3.2.2 Confusion matrix

From the shown confusion matrix, it can be clearly observed that the CNN technique has completely outperformed the YOLOv4 algorithm and proven its proficiency in accurate object detection and recognition. The YOLOv4 in field testing lacks true positive cases (TP = 103), whereas its greater true negative (TN = 79) and false negative (FN = 12) values result in lower precision as compared to CNN.^[29]

3.3 Discussion

The prototype in this paper is being developed and implemented using CNN classification and YOLOv4 supervised algorithms for a comparison-based study and

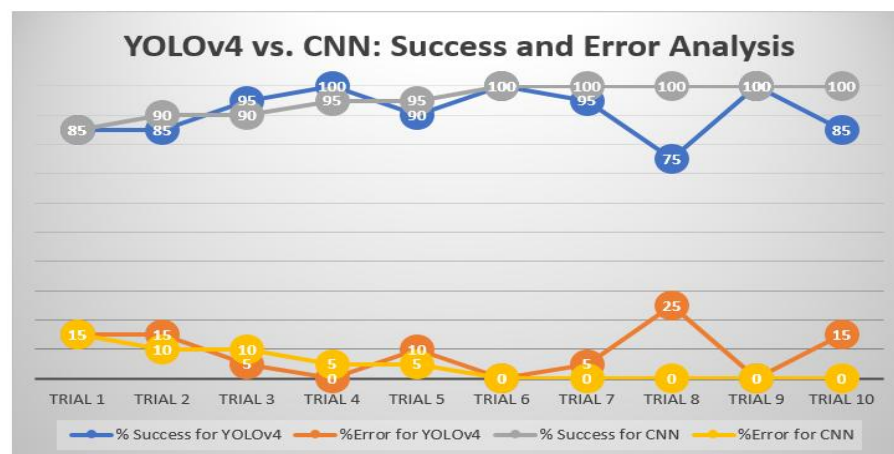


Fig. 12: Graphical representation of CNN results.

Actual: Positive	103	12
Actual: Negative	6	79
	Predicted: Positive	Predicted: Negative

Fig. 13: Confusion matrix for YOLOv4 technique.

Actual: Positive	116	5
Actual: Negative	4	75
	Predicted: Positive	Predicted: Negative

Fig. 14: Confusion matrix for CNN technique.

detailed analysis in the search for the best algorithm to be implemented. This step is particularly necessary for accurate classification of weeds and crops based on different geographical locations and regions. The main objective of this study was to determine the optimal performance of various deep learning (DL) algorithms in classification and precise elimination of weeds amongst the crop field.

In this study, the F1 Score that the model achieved for YOLOv4 was 91.96% while for the CNN technique it achieved a score of 96.25%. This was observed as the YOLOv4 technique is faster but cannot catch on to complex scenarios and smaller details of the particular object to be detected. Thus, it misses certain aspects of the weeds and doesn't provide detection in certain case scenarios, providing faster speeds for sure but compromises accuracy in detecting smaller or overlapping features of the object. Whereas, CNN is less likely to miss object detection as it focusses more on specific details of an object to be detected.

While the custom CNN-based classifier demonstrated higher accuracy in identifying weed presence, it does not localize the exact position of the weeds. This limits its practical application for precision spraying. In contrast, YOLOv4 is an object detector that not only identifies weeds but also provides spatial coordinates, enabling site-specific weed management. Therefore, the comparison is not entirely

direct, as the two models serve complementary rather than identical purposes.

Now as we observe in Table 3, a through comparison has been stated amongst accuracies of four other methods being generally used in effective object detection and classifications with the two root methods mentioned in this study. Jun Zhang *et al.*^[12] mentioned in his study about the higher accuracy of the original ViT model due to its stronger sequence modelling abilities and unique capabilities to capture long-range dependencies. But when we carefully consider both CNN and ViT in a comprehensive way, then the CNN model due to its better balance for local and global features, results in a overall better performance and improved classification.

Table 3: Comparison with other detection techniques.

Model Name	Accuracy (%)
VGG16	86.21
GoogleNet	79.23
AlexNet	80.09
ViT	89.09
YOLOv4	91.00%
CNN	95.50%

As weed detection has proved to be the most challenging task for development of autonomous robotic weed detection and elimination systems, robust and precise computer-vision based detection and sprayer systems that implement deep learning algorithms can overcome this particular challenge by accurately identifying the weeds among the crop fields and effectively eliminating the particularly targeted weed.

When we are to talk about the future scope and research possibilities in this particular ground then, more focus can be asserted upon developing and curating bigger and much more detailed dataset that provides much deeper and rich classification opportunities for the algorithm and its hidden layers. Also, focus can be asserted more on using hardware with better computational capabilities and processing power like powerful GPUs and high-performance CPUs as results will drastically improve due to efficient processing of millions of parameters and simplified matrix operations.

4. Conclusion

A robust Weed detection and elimination system is the needed in-order to efficiently boost-up the agriculture sector for large scale production of healthy crops and utilization of limited agriculture resources in an efficient way. The system in this study proposes a unique way of developing a prototype using machine learning and deep learning algorithms that harnesses computer vision technology for accurate classification of weeds and crops without any involvement of human labour or assistance. The study suggests selection of appropriate deep learning technique for the task that can achieve high end and promising results in the particular field of application. The CNN algorithm

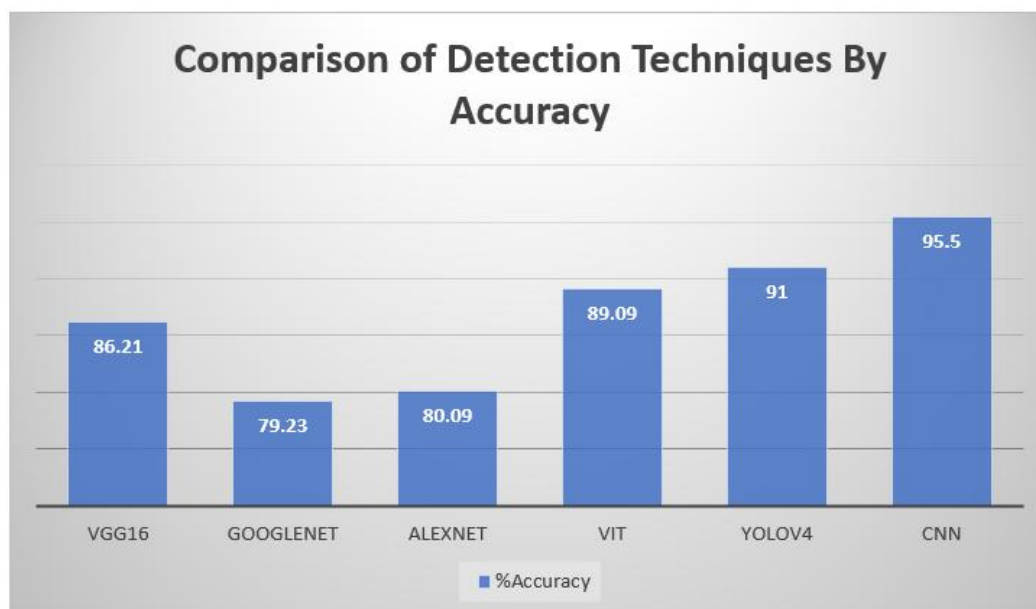


Fig. 15: Graphical comparison between various techniques.

proved to be more precise and accurate in doing so with an accuracy of 95.50%, precision and recall of 96.66% and 95.86% respectively. This surpasses the scores of the YOLOv4 technique for weed detection, although it cannot beat the speed and agility of YOLOv4 but when it comes to accurate classification and comprehensive detection CNN takes up the stakes and proves it worth by securing an F1 Score of 96.25%. The CNN classifier model is suitable for general field assessment, such as identifying whether weeds are present in an image. However, for the practical application of targeted and precision spraying, the YOLOv4 object detector is essential due to its ability to localize weeds within the image. YOLOv4 achieved an average inference speed of 30 FPS (frames per second), making it suitable for real-time applications, whereas the custom CNN model averaged around 5 FPS, making it more suitable for offline analysis. Although future research scope for this particular field of study is broad and insightful, yet this paper successfully highlights certain areas of aspect that can significantly improve the performance of a large scale weed detection and elimination system. Despite of certain limitations being encountered during the implementation of the study such as artificial lighting conditions, shadow overlaying *etc.*, the authors have achieved to prove the proficiency of the particularly suggested method of implementation for future implementations to come.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable

Use of artificial intelligence (AI)-assisted technology for manuscript preparation

The authors confirm that there was no use of artificial intelligence (AI)-assisted technology for assisting in the writing or editing of the manuscript and no images were manipulated using AI.

References

- [1] C. S. G. Sunil, Y. Zhang, C. Koparan, M. R. Ahmed, K. Howatt, X. Sun, Weed and crop species classification using computer vision and deep learning technologies in greenhouse conditions, *Journal of Agriculture and Food Research*, 2022, **9**, 100325, doi: 10.1016/j.jafr.2022.100325.
- [2] B. Turan, I. Kadioglu, A. Basturk, B. Sin, A. Sadeghpour, Deep learning for image-based detection of weeds from emergence to maturity in wheat fields, *Smart Agricultural Technology*, 2024, **09**, 100552, doi: 10.1016/j.atech.2024.100552.
- [3] A. Upadhyay, G. C. Sunil, Y. Zhang, C. Koparan, X. Sun, Development and evaluation of a machine vision and deep learning-based smart sprayer system for site-specific weed management in row crops: An edge computing approach, *Computers and Electronics in Agriculture*, 2024, **216**, 108495, doi: 10.1016/j.jafr.2024.101331.
- [4] S. Zahoor, S. A. Sof, Weed identification in crop field using CNN, *Journal of University of Shanghai for Science and Technology*, 2021, **23**, 15-21, doi: 10.3390/smartcities3030039.
- [5] P. K. Reddy, R. A. Reddy, M. A. Reddy, K. Sai Teja, K. Rohith, K. Rahul, Detection of weeds by using machine learning, *Proceedings of the International Conference on Emerging Trends in Engineering and Technology*, 2023, 882-892.
- [6] W. -H. Su, Advanced machine learning in point spectroscopy, RGB- and Hyperspectral-imaging for automatic discriminations of crops and weeds: a review, *Sensors*, 2021, **21**, 4707, doi: 0.3390/smartcities3030039.

- [7] U. S. Umanaheswari, A. R. Arjun, M. D. Meganathan, Weed detection in farm crops using parallel image processing, 2018 Conference on Information and Communication Technology (CICT), Jabalpur, India, 2018, 1-4, doi: 10.1109/INFOCOMTECH.2018.8722369.
- [8] O. M. Olaniyi, E. Daniya, J. G. Kolo, J. A. Bala, A. E. Olanrewaju, A computer vision-based weed control system for low-land rice precision farming, *International Journal of Advances in Applied Sciences*, 2020, **9**, 51-61, doi: 10.11591/ijaas.v9.i1.pp51-61.
- [9] M. D. Bah, A. Hafiane, R. Canals, Deep learning with unsupervised data labeling for weed detection in line crops in UAV images, *Remote Sensing*, 2018, **10**, 1690, doi: 10.3390/rs10111690.
- [10] V. Partel, S. C. Kakaria, Y. Ampatzidis, Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence, *Computers and Electronics in Agriculture*, 2019, **157**, 339-350, doi: 10.1016/j.compag.2018.12.048.
- [11] Y. Wang, H. Liu, D. Wang, D. Liu, Image processing in fault identification for power equipment based on improved super green algorithm, *Computers & Electrical Engineering*, 2020, **87**, 106753, doi: 10.1016/j.compeleceng.2020.106753.
- [12] J. Zhang, Weed recognition method based on hybrid CNN-transformer model, *Frontiers in Computing and Intelligent Systems*, 2023, **4**, 72-77, doi: 10.54097/fcis.v4i2.10209.
- [13] L. Moldvai, P. Ákos Mesterházi, G. Teschner, A. Nyéki, Weed detection and classification with computer vision using a limited image dataset, *Computers and Electronics in Agriculture*, 2024, **214**, 108301, doi: 10.3390/app14114839.
- [14] H. Jiang, C. Zhang, Y. Qiao, Z. Zhang, W. Zhang, C. Song, CNN feature-based graph convolutional network for weed and crop recognition in smart farming, *Computers and Electronics in Agriculture*, 2020, **174**, 105450, doi: 10.1016/j.compag.2020.105450.
- [15] M. A. Haq, CNN based automated weed detection system using UAV imagery, *Computer Systems Science and Engineering*, 2022, **42**, 837-849, doi: 10.1016/j.csse.2022.105450.
- [16] P. K. Reddy, R. A. Reddy, M. A. Reddy, K. S. Teja, K. Rohith, K. Rahul, Detection of weeds by using machine learning, Proceedings of International Conference on Emerging Trends in Engineering, B. Raj et al., Eds., Springer, 2023, 882-892, doi: 10.2991/978-94-6463-252-1_89.
- [17] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, R. Fergus, Regularization of neural networks using DropConnect, Proceedings of the 30th International Conference on Machine Learning, 2013, **28**, 1058-1066.
- [18] B. Jabir, L. Rabhi, N. Fali, RNN- and CNN-based weed detection for crop improvement: An overview, *Foods and Raw Materials*, 2021, **9**, 387-396, doi: 10.21603/2308-4057-2021-2-387-396.h.
- [19] Y. Tang, Deep learning using linear support vector machines, arXiv preprint arXiv:1306.0239, 2015.
- [20] A. Bochkovskiy, C. -Y. Wang, H. -Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934, 2020.
- [21] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, Proceedings CVPR'16, 2016, 779-788, doi: 10.48550/arXiv.1506.02640.
- [22] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, arXiv preprint arXiv:1804.02767, 2018.
- [23] O. L. Garcia-Navarrete, A. Correa-Guimaraes, Application of convolutional neural networks in weed detection and identification: A systematic review, *Computers and Electronics in Agriculture*, 2024, **216**, 108520.
- [24] M. Ofori, O. El-Gayar, An approach for weed detection using CNNs and transfer learning, Proceedings of the 54th Hawaii International Conference on System Sciences, 2021, 888-895.
- [25] R. Sapkota, J. Stenger, M. Ostlie, P. Flores, Towards reducing chemical usage for weed control in agriculture using UAS imagery analysis and computer vision techniques, *Scientific Reports*, 2020, **13**, 6548, doi: 10.1038/s41598-023-33042-0.
- [26] B. B. Sapkota, C. Hu, M. V. Bagavathiannan, Evaluating cross-applicability of weed detection models across different crops in similar production environments, *Frontiers in Plant Science*, 2022, **13**, doi: 10.3389/fpls.2022.837726.
- [27] O. E. Apolo-Apolo, M. Fontanelli, C. Frascioni, M. Raffaelli, A. Peruzzi, M. P. Ruiz, Evaluation of YOLO object detectors for weed detection in different turfgrass scenarios, *Applies Sciences*, 2023, **13**, 8502, doi: 10.3390/app13148502.
- [28] M. A. Saqib, M. Aqib, M. N. Tahir, Y. Hafeez, Towards deep learning based smart farming for intelligent weeds management in crops, *Frontiers in Plant Science*, 2023, **14**, doi: 10.3389/fpls.2023.1211235.
- [29] V. S. Babu, N. Venkatram, Weed detection and localization in soybean crops using YOLOv4 deep learning model, *Traitement du Signal*, 2023, **41**, 1019-1025, doi: 10.18280/ts.410242.

Publisher Note: The views, statements, and data in all publications solely belong to the authors and contributors. GR Scholastic is not responsible for any injury resulting from the ideas, methods, or products mentioned. GR Scholastic remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

Open Access

This article is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License, which permits the non-commercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as appropriate credit to the original author(s) and the source is given by providing a link to the Creative Commons License and changes need to be indicated if there are any. The images or other third-party material in this article are included in the article's Creative Commons License, unless

indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons License and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this License, visit: <https://creativecommons.org/licenses/by-nc/4.0/>

© The Author(s) 2025