International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-12

http://dx.doi.org/10.12785/ijcds/1571054829

Exploring Novel CNN Architectures for Weed Seedling Recognition in Precision Agriculture

Monisha R[1], Tamilselvan KS[1], Vaishnavi T[2] and Sharmila A[3]

[1]*Department of ECE, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India – 641 407*
[2]*Department of IT, Kongu Engineering College, Erode, Tamil Nadu, India – 638 060*
[3]*Department of ECE, Bannari Amman Institute of Technology, Coimbatore, Tamil Nadu, India – 638 401*

Received 24 July 2024, Revised 27 January 2025, Accepted 29 January 2025

Abstract: Precision agriculture (PA) aims to maximize crop yields while minimizing inputs such as water, fertilizer and pesticides. To achieve this, PA relies on advanced technologies such as sensors, drones, and satellite imagery to monitor crops and optimize input. However, weeds pose a significant challenge, competing with crops for vital resources and thereby reducing production output. For weeds to be managed and controlled effectively, they must be accurately categorized. Effective weed management requires understanding each weed's characteristics, which can be challenging with traditional methods. In this research, a comparative analysis of a mixture of different lightweight and dense convolutional neural network models was conducted to classify multiclass weed seedlings. The models included AlexNet, Inception v3, ResNet (18, 34, 50, and 101), SqueezeNet (1.0, 1.1), VGG 16, VGG 19, XResNet (18, 34, 50, and 101), and XSEResNet (18, 34, 50, and 101). The results demonstrated that the Inception v3 model achieved better performance with 95% accuracy and 95% F1-score, surpassing other architectures. However, due to Inception v3's dense structure and large number of trainable parameters, efforts were taken to reduce its complexity without compromising its performance. As a result, a new, simplified version of Inception v3 was proposed, with improved metrics of 96% accuracy and 95.8% F1-score.

Keywords: Convolutional neural networks, Inception v3, Fast.ai, Precision agriculture, Weeds

1. Introduction

By 2050, it is estimated that there will be 9.7 billion people on the earth, according to the United Nations [1], leading to the continuous increase of food consumption, driven by the rapidly expanding human populace. This population growth, coupled with increasing urbanization and changing dietary habits, puts significant pressure on the agricultural sector to produce more food while minimizing environmental impact. There is an urgent need for the application of non-invasive modern technologies to better meet the food demand in the future. An approach that combines various technologies for acquiring and examining field data, processing it, and utilizing it appropriately for the task at hand is known as precision agriculture (PA) [2]. It is often defined as the highest degree of exactness, considering multiple aspects of crop cultivation [3]. Weed management, which removes the hindrances produced by risky crops, is one of the main aspects covered by PA. Weeds contend with crops for water, nutrition, and sunlight and tend to overgrow them [4]. A three-year research study finds that the majority of the weed populations result in significant output deficits in unweeded zones, spanning 19% to 56%. [5]. Moreover, the consumption of certain species of weeds by animals results in an unfavorable odor in their milk and also poses serious threats to their health. They also impact the ecology and surrounding environment by changing the quality of the soil, disrupting native plant populations, and causing erosion. The farmers also incur additional expenses, mostly in the form of labor, time, and financial losses due to poor harvests. These situations emphasize the necessity for proper weed management. So far the methods used to curb weed growth can be categorized into mechanical, chemical, and cultural methods. Mechanical methods, historically the first approach to handle weeds, include hand weeding, mulching, and tillage. But they pose certain challenges in real-world applications, such as their reliance on skilled operators and dependence on weather and soil conditions [6]. Chemical treatments, such as herbicide application, are used to control a wide range of weeds or to narrowly target specific weed species. However, they might contaminate soil and water, harming unintentional plants and animals and hastening the degradation of the ecosystem. There are also concerns about the potential health risks that agricultural workers and farmers may face from prolonged exposure to the chemicals. In addition, herbicide-resistant weeds might arise due to misuse or improper application,

E-mail address: 21ftec004@kpriet.ac.in, tamilselvan@kpriet.ac.in, vaishnavi.it@kongu.edu, sharmilaa@bitsathy.ac.in

making long-term suppression of weeds more difficult [7]. Cultural strategies, such as cover cropping and crop rotation, provide suppression of weeds rather than destruction. However, these controls require extensive preparation and precise timing. Advancements in machine learning (ML) and deep learning (DL) algorithms in recent years have made it easier to classify weeds based on images [8], [9]. Numerous fields, including detecting facial expressions [10], recognizing speeches [10], predicting crop yield [11], and classifying plant disease [12], have seen impressive results from these algorithms. ML algorithms frequently depend on feature engineering and extraction to get high performance [13]. However, the requirement for particular extraction circumstances limits the usefulness of shape-based characteristics. Later, because of their improved performance and capacity to automatically extract intricate features from images, DL algorithms—in particular, Convolutional Neural Networks (CNNs)—have become a potent tool for weed detection in computer vision applications [14]. With architectures ranging from AlexNet [15] for image classification to YOLO [16] for object recognition, CNNs have been increasingly popular for classification, object detection, and segmentation applications. DL techniques have been useful in a number of earlier research studies that investigated the classification of crops and weeds. Even with these developments, CNN-based weed categorization still has several shortcomings. Whether dense, original CNN architectures or lightweight variations offer higher accuracy for weed detection is a crucial subject that requires attention. The purpose of this research is to assess the applicability of cutting-edge CNN architectures for attaining high accuracy and examine how well they function in the classification of weeds. Furthermore, the study aims to create and suggest a new, lightweight Inception v3 model that is more accurate while yet being more efficient. The study on the trade-off between complexity and performance also needs to be addressed. This research makes several important contributions:

- The study uses six different types of weeds and performs experiments with the Fast.ai library.
- A comparative analysis is performed on 18 pre-trained CNN architectures, including AlexNet, Inception v3, ResNet (18, 34, 50, and 101), SqueezeNet (1.0, 1.1), VGG (16, 19), XResNet (18, 34, 50, and 101), and XSEResNet (18, 34, 50, and 101).
- The Inception v3 network is modified to produce a simpler model, which is compared against pre-trained CNN networks.
- The results show that the simplified Inception v3 performs better in weed classification, with 96% accuracy, 96.5% precision, 95.6% recall, and 95.8% F1-score. Furthermore, the simplified model offers 37.5% less complexity than the original Inception v3.

This paper offers useful insights into the implementation of contemporary CNN architectures in precision agriculture settings, in addition to highlighting their potential for weed categorization.

2. Background

PA aims to maximize crop productivity while minimizing resources by relying on advanced technologies like sensors, drones, and satellite imagery. Among the challenges faced in PA, weed detection and control remain critical, as weeds compete with crops for essential resources. Traditional weed management approaches—manual, chemical, biological, and mechanical [17], [18], [19], [20]—have limitations in efficiency, scalability, and environmental impact, prompting advanced solutions [21]. Modern weed identification techniques employ AI and fall into three categories: ML, DL, and hybrid approaches [21], [22]. These paradigms overcome the limits of conventional methods by automating feature extraction and increasing accuracy under various scenarios. ML approaches have been beneficial in early research that relied primarily on feature-based strategies. These methods involve the extraction of distinctive features such as color, texture, shape, and spectral qualities, followed by the use of ML classifiers. For example, support vector machines have been widely employed in weed classification and segmentation due to their robustness in dealing with high-dimensional data when compared to other models [23], [24], [25], [26]. Random forest and k-means clustering, on the other hand, have shown excellent performance at distinguishing weeds from other crops [27], [28], [29]. However, despite their perceived simplicity and effectiveness in controlled environments, ML systems have severe limits. They rely on the accuracy and relevance of manually derived characteristics, which are frequently task-specific and fail to generalize across varied environmental situations [29]. These models rarely adapt to complicated, high-dimensional data, and they struggle with overlapping plant structures found in real-world agricultural areas [26]. Recent advances in DL have provided potential solutions in the field of weed detection. These models learn hierarchical feature representations straight from raw data, removing the requirement for human feature extraction. CNNs, in particular, have performed admirably in image-based weed classification tasks. For example, Subeesh et al. used four state-of-the-art CNN models to classify weeds in controlled conditions. Among which, Inception v3 outperformed the other models, with 97.7% accuracy. Despite the positive results, the model's scalability was limited to difficult field conditions with dense vegetation [30]. Peng et al. modified RetinaNet for real-time detection, attaining 24.3 frames per second on a GTX TITAN X GPU. Although speed was enhanced, accuracy diminished, highlighting the necessity for additional specialized datasets and fine-tuning for various field circumstances [31]. Later, transfer learning was widely adopted, as it used pre-trained models trained on large benchmark datasets and fine-tuned the model for specific weed detection tasks. Tannouche et al. utilized VGG 16, VGG 19, Inception v3, Inception v4, MobileNet v1, and

MobileNet v2 to distinguish weeds. The results showed that Inception v4 achieved the highest precision with a rate of 99.51% on the mixed image sets and for its pre-trained version. While these models showed high classification accuracy, they required extensive manual preprocessing, such as segmenting the region of interest (ROI) for classification [32]. Garibaldi-Márquez et al. performed ROI extraction using connected component analysis and classification using ML and DL models individually. An average accuracy of 97.5% was obtained for all DL models, which was reported to be higher than ML models. This approach worked effectively only when the plants were in their early growth phases and when there was no occlusion or overlap in the field [33]. Maram et al. identified ROI and applied image enhancement techniques to the dataset for weed detection, which led to improvement in accuracy. Still, the authors found the model overfitting, indicating difficulties in generalizing to variable environments [34]. Jin et al. explored GoogLeNet, MobileNet v3, ShuffleNet v2, and VGGNet for detecting and discriminating weeds growing in turfgrass based on their susceptibility to ACCase-inhibiting and synthetic auxin herbicides. The authors reported that ShuffleNet v2 was the most efficient and reliable model among the neural networks evaluated. While all of the studied models enabled quick detection, there were misclassifications, particularly in regions where there was a need to detect individual weed species rather than discriminating weeds based on their susceptibility to herbicides [35]. Jin et al. utilized DenseNet, EfficientNetV2, ResNet, RegNet, and VGGNet to improve specificity in weed detection within grid cells [36]. Hybrid methodologies integrate ML and DL models or various DL models to enhance performance. Hybrid models gained popularity, combining architectures to balance feature extraction and classification accuracy, resulting in more reliable weed and crop categorization. Al-Badri et al. created a hybrid model that combines VGG-16, ResNet-50, and Inception-V3 to categorize Rumex weed plants in different light and overlapping situations, but it was only evaluated on one weed species [37]. Nasiri et al. used U-Net with ResNet-50 to enhance segmentation accuracy in tiny regions, although overlap difficulties remained [38]. Occasionally, the classification of weeds has been enhanced through the application of optimization techniques utilizing metaheuristic algorithms. Veeragandham and Santhi proposed a deep quantum neural network framework optimized using the Coot political optimization algorithm, resulting in a classification accuracy of 93.6%. However, adaptability to different soil types was restricted, highlighting the need to incorporate soil properties [39]. Dadashzadeh et al. created a stereoscopic system that used hybrid NN-ICA and NN-PSO algorithms, resulting in good classification accuracy under regulated lighting; nonetheless, the heterogeneity in weed species and growth stages limited generalizability [40]. Recent efforts have concentrated on enhancing models for edge computing and real-time processing, which are crucial for practical implementations in the domain. Razfar et al. investigated lightweight CNN architectures deployed on resource-constrained hardware, such as Raspberry Pi, and achieved acceptable accuracy and latency [41]. Rai et al. combined YOLO v8, MobileNet v3, and EfficientNetB0 with the Nvidia Jetson AGX Orin, applying techniques like pruning, quantization, and knowledge distillation to reduce computational costs [42]. Hasan et al. utilized YOLO v7 and YOLO v8 for corn weed detection on the Nvidia Jetson AGX Orin, achieving real-time weed detection suitable for selective sprayer applications. However, environmental conditions like lighting and plant occlusion affected accuracy, emphasizing the need for robust datasets and validation under diverse field conditions [43]. In their work on CNNs using UAV images, Mesías-Ruiz et al. pointed out that high-resolution, annotated datasets are crucial for achieving the best results [44]. For real-time implementation, edge computing was advised; however, field unpredictability and data quality constraints presented difficulties. Even after optimizations, constraints like device costs and environmental unpredictability persisted, indicating that further work is required to improve the model's affordability and robustness. Despite their capabilities, DL models encounter several drawbacks. Deep network training necessitates a significant amount of computer power and huge annotated datasets, which may not be easily accessible in agricultural settings. While substantial progress has been achieved in weed detection using ML, DL, and hybrid techniques, major gaps still exist. There is a demand for lightweight, efficient CNN architectures that can operate in real-time and under resource constraints. Existing models have limited applicability in various and complicated agricultural situations. Challenges in accurately segmenting overlapping crops and weeds. These gaps highlight the significance of creating unique, efficient, and adaptive weed identification frameworks that capitalize on the capabilities of ML, DL, and hybrid techniques while resolving their limitations. This work takes a methodical and practical approach to weed detection, building on the benefits of prior methods while addressing their limitations. Unlike standard ML methods that rely on task-specific handcrafted features that lack generalizability, this work uses DL models capable of learning hierarchical representations directly from raw data. While previous studies have achieved great accuracy using pre-trained CNN architectures such as VGG, ResNet, and Inception, they frequently deal with computational complexity, overfitting, and environmental unpredictability. This study goes a step further, doing a thorough comparison of 18 pre-trained CNN models to determine the most efficient architecture. This reduction solves a fundamental barrier for implementing deep learning models in resource-constrained areas, such as agricultural fields, where edge computing devices are common.

3. Materials

A. Dataset

A comparative analysis was performed on a subset of the V2 Plant Seedlings dataset from Kaggle [45]. The subset encompasses six classes of weeds in their seedling stage as RGB images. The class contains some of the most common weeds in large arable lands. Fig. 1 gives the image

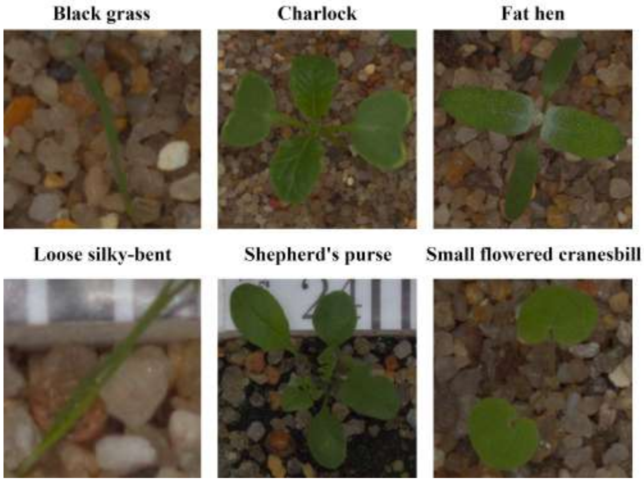

Figure 1. Samples of weed seedlings in the dataset

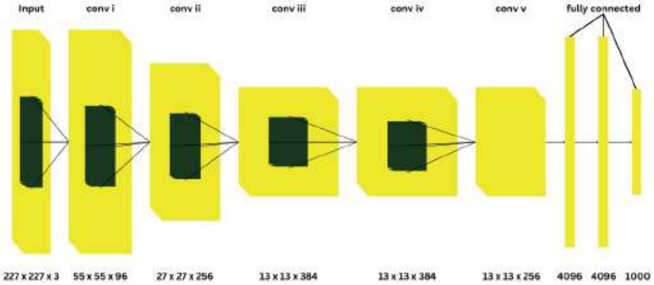

Figure 2. AlexNet

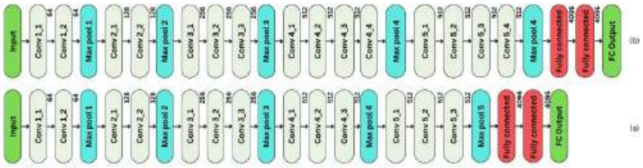

Figure 3. (a) VGG 16; (b) VGG 19

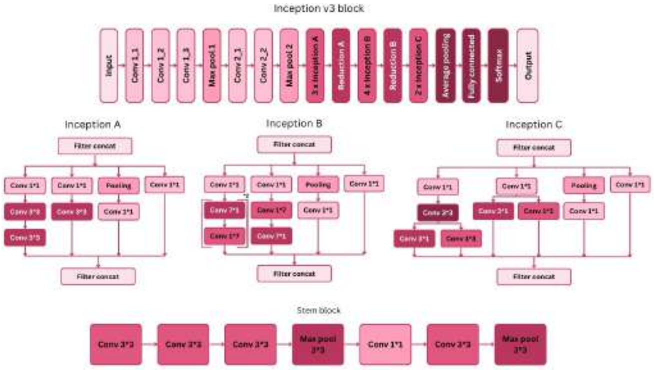

Figure 4. Inception v3

samples of each weed included in the dataset. The dataset comprised 2,911 images with an uneven number of samples in every class, thus making it an imbalanced dataset. The class distribution and leaf characteristic appearance of every weed species member are given in Table 1. Google Colab on a laptop with Microsoft Windows 10 Pro and an Intel Core i5 processor was used to implement all the architectures.

B. CNN Models

1) AlexNet

In 2012, Alex Krizhevsky and colleagues proposed AlexNet, as shown in Fig. 2, a deep CNN model that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), marking a breakthrough in computer vision [15]. AlexNet outperforms traditional ML methods and spurs rapid growth in DL. The architecture features five convolutional layers, two fully connected (FC) layers, and a softmax layer. The first layer uses 96 receptive filters (11×11) with local response normalization and max pooling. Layers starting from two to five use 5×5 and 3×3 filters, with 384, 384, and 256 feature maps, respectively. Dropout is applied in the FC layers to reduce overfitting.

2) VGGNet

The Visual Geometry Group (VGG) was runner-up in the 2014 ILSVRC [46]. The key contribution of VGG is to demonstrate that increasing network depth improves classification accuracy in CNNs. The VGG architecture includes two convolutional layers with ReLU activation, followed by a max pooling layer and FC layers, all using ReLU as depicted in Fig. 3. The final layer is a softmax layer. The VGG-E models—VGG-11, VGG-16, and VGG-19—differ in the number of convolutional layer: VGG-11 has 8, VGG-16 has 13, and VGG-19 has 16. VGG-19, the most computationally expensive, has 138 million parameters.

3) Inception v3

Inception v3, proposed by Google in 2015, is part of the GoogLeNet family, which uses the Inception network structure [47]. This structure reduces network parameters while increasing depth, making it ideal for image classification tasks. The Inception module typically combines three different sizes of convolutions and one max pooling operation which is shown in Fig. 4. This aggregation improves the network's ability to handle various scales and helps prevent overfitting. Inception v3 introduced several improvements over its predecessors, such as splitting large convolution kernels into smaller ones, reducing parameters, and accelerating training speed.

4) ResNet

Generally, in DL, it is said that the deeper the network, the higher the accuracy. As such, DL networks designed focused on stacking layers and complicating the model. However, it was seen that the network performance diminished when a certain threshold for layering was reached. The problem is attributed to the gradients becoming zero after complex computations and thus having nothing to

Class	Weed Name	Scientific Name	Leaf Characteristics	Train Images	Valid Images
BG	Black grass	*Alopecurus myosuroides*	Short, flat, bluish-green, and hairless. Typically rolled, rough, and having a clearly discernible keel	242	67
CK	Charlock	*Sinapis arvensis*	A wide, rounded tip with shallow ridges running around its edges	381	71
FH	Fat hen	*Chenopodium album*	Toothed borders, which are comparatively wide	445	93
LSB	Loose silky-bent	*Apera spica-venti*	Sharp, slender, coarse, and hairless. Rolled.	597	165
SP	Shepherd's-Purse	*Capsella bursa-pastoris*	The earliest true leaves are whole; later leaves are severely lobed or sliced, lance-shaped, and silvery in appearance.	207	67
SFC	Small-flowered cranesbill	*Geranium pusillum*	The leaves are opposite and have deeply and rather thinly cut hairs on the stalks; each solitary leaf lobe frequently has three smaller lobes.	457	119
			Total	**2,329**	**582**

TABLE I. Weed characteristics and class distribution

learn more from the data [48]. Thus, to avoid the vanishing gradient problem, the authors of ResNet introduce residual blocks with skip connections [49]. The main idea behind ResNet is to use residual connections, which allow information to stream from one layer to another, without passing through all the intermediate layers, as shown in Fig. 5. Thus, a leftover block is produced. To build ResNets, these leftover blocks are piled. ResNet's 34-layer simple network architecture is inspired by VGG-19, and the shortcut connection is added after that. ResNet makes it possible to train neural networks with hundreds of layers that are incredibly deep while still maintaining good accuracy. In this architecture, a convolutional layer processes the image input first, followed by several blocks with residuals. A shortcut link that adds the original input to the convolutional layers' output follows each residual block's two or more convolutional layers. This allows the network to learn the residual mapping—the difference between the block's input and output. To create the final output for classification, ResNet also employs an FC layer at the end and an overall pooling layer. In general, as the number of layers increases, the network becomes deeper and more complex, allowing it to learn complex features and achieve good performance on tasks such as image classification.

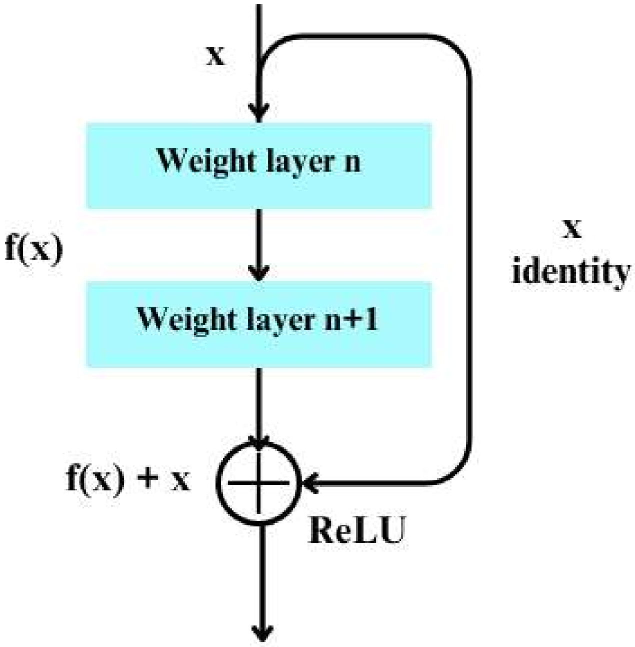

Figure 5. Residual block

5) XResNets

The ResNet model was tweaked as XResNets, which incorporated slight modifications, each variant focusing on different layers of the base architecture [50]. These modifications, often referred to as ResNet-B, ResNet-C, and ResNet-D, address different aspects of the architecture as shown in Fig. 6. ResNet-B and ResNet-D concentrate more on retaining more data, whereas ResNet-C concentrates on reducing the computational complexity. The former has alterations in the downsampling block by changing the strides for the convolution operation. The latter replaced the 7×7 convolution with three 3×3 convolutions. Though the variants underperform in accuracy when compared to the base model, they highlight the importance of random points in layer selections that could affect the overall efficacy.

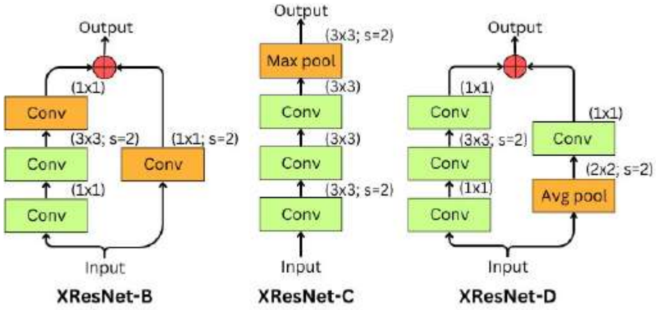

Figure 6. XResNets

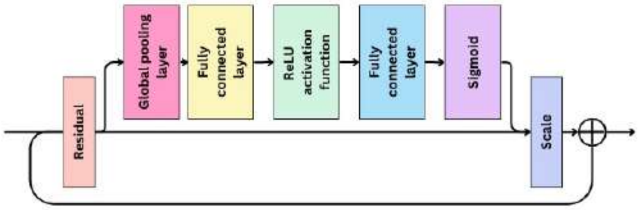

Figure 7. SE-Net

6) SE-ResNets

SE-Nets, squeeze and excitation networks, introduced a method to weigh each channel instead of assigning equal weights to all input channels [51]. Basically, in CNN's multi-channel architecture, the top layers are responsible for high-level feature extraction, whereas the bottom layers extract simple features like edges. To avoid sharing the same weights across all input channels, SE-Nets perform the weighing by parameterizing the weights at the end of the block. They take in a residual convolutional feature map as input and then apply average pooling, which results in reduced dimensions. Later, two FC layers are used for non-linear representation using bottleneck parametrization. The first FC layer is followed by ReLU and the second by the sigmoid activation function. The output of these layers is used to calculate the weights of each channel in a neural network. The model is illustrated in Fig. 7.

7) SqueezeNet

SqueezeNets were developed to design a network that can be deployed in any edge device or computer network [52]. They have fewer trainable parameters, thus rendering a small network with minimal processing time and memory. The two methodologies of the SqueezeNet model are built upon the AlexNet: one with heavy compression and the other with a hybrid compression technique. AlexNet has five convolutional layers in combination with a pair of max pooling and ReLU layers and three final dense layers. The entire network consisted of 61 million parameters. SqueezeNet, with its base as AlexNet, is made up of two convolutional layers, eight fire modules, three max-pooling layers, and one global average pooling layer, as shown in Fig. 8. In the fire block, an expand layer with a combination of 1×1 and 3×3 convolution filters receives input from a squeeze convolution layer, which only has 1×1 filters. The fire block consists of three hyperparameters: The number of filters (all 1×1) in the squeeze layer is denoted by $S_{1\times1}$. The expand layer's number of 1×1 filters is denoted by $e_{1(1}$ and the number of 3×3 filters is denoted by $e_{3(3}$. The sum of these filters is $S_{1(1} < (e_{1(1} + e_{3(3})$. To reduce the amount of input channels for the 3×3 filters, the squeeze layer is used. Here, concat has been used to link many layers to improve expressiveness (expressiveness in this context refers to the earlier portions' extraction of features and spatial information from the images). Furthermore, no FC layer exists. This yields a vector that has been flattened and whose dimension is equal to the number of classes. This vector is then supplied to the softmax layer. The number of parameters is significantly reduced when FC layers are absent.

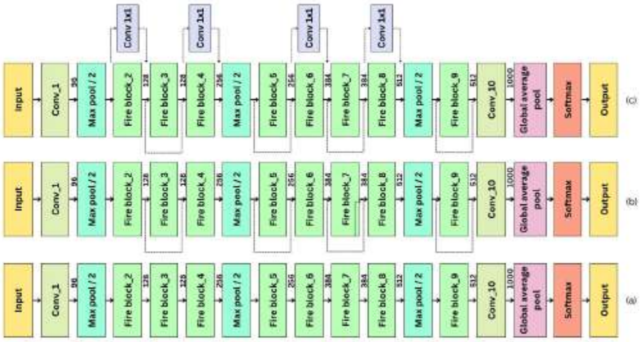

Figure 8. (a) SqueezeNet module; (b) with bypass; (c) with complicated bypass

8) Proposed Method

The proposed model had the following modifications:

- Filter Reduction: Lowering the number of filters in the inception blocks reduces the model's overall complexity. Less Overfitting: A simpler model is less likely to memorize training data specifics and generalize better to unseen images. Faster Training: Fewer parameters translate to faster training times for your specific dataset.

- Auxiliary Classifier Removal: Removing these classifiers simplifies the training process and might lead to the model focusing more on the main classification task.

- 3x3 Split Convolution: Splitting large convolutions into smaller ones can improve computational efficiency without sacrificing the receptive field (area considered by the filter). This can be particularly beneficial for resource-constrained environments.

- Block Removal: Removing $Mixed_7c$: Replacing the final block with an identity layer effectively stops the network from further processing features at that stage. This can be beneficial if the additional processing capacity doesn't contribute significantly to weed classification in your dataset.

4. Methods

First, the training data was pre-processed using various augmentation techniques, such as random flipping and resizing. These pre-processing steps increased the diversity of the training data, making the model more robust and resistant to overfitting. Next, the 18 CNN models' architecture was used for feature extraction and classification. During training, the weights of the new layer were updated using backpropagation and gradient descent, optimizing the model for the classification task. Finally, their results have been compared in terms of classification reports and confusion matrices. Throughout the training process, the model was evaluated on a validation set to ensure that it is not overfitting to the training data. The hyperparameters used for all models are discussed in Table 2.

Parameters	Values
Loss function	Categorical cross-entropy
Learning rate	0.001
Epochs	50
Batch size	30
Library	Fast.ai

TABLE II. Hyperparameters

5. Performance Metrics

Several performance indicators for DL classification models have been applied to evaluate how well CNN-based algorithms perform in a specific scenario. The performance metrics specified in the equations (1)-(4) are considered for this study:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (4)$$

where True Positive (TP): The model accurately estimates a positive class. False Positive (FP): The model misidentifies a positive class. True Negative (TN): The model accurately predicts a negative class. False Negative (FN): The model mistakenly forecasts a negative class. In addition to the four metrics mentioned above, confusion matrices are also used. A confusion matrix is a table that compares a dataset's predicted labels against its true labels and summarizes the performance of a classification model. It comprises totals for TP, FP, TN, and FN. This tool assists with model evaluation, threshold selection, and understanding performance trade-offs by providing a short summary of prediction accuracy in classification tasks.

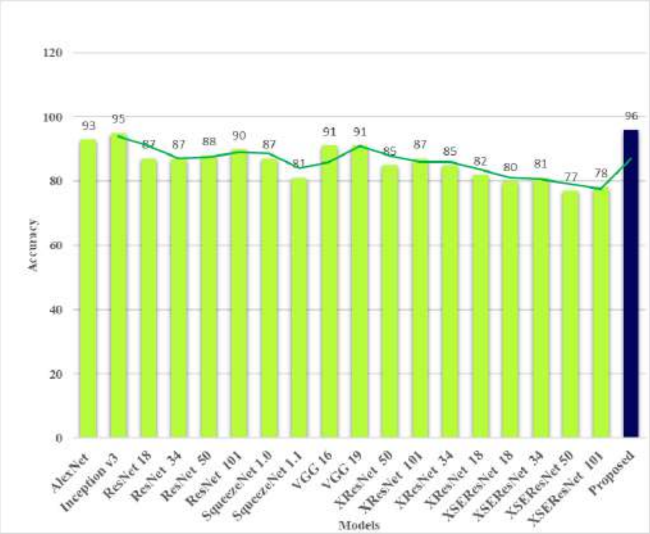

Figure 9. Accuracy comparison among pre-trained CNN and proposed model

6. Experimental Results and Discussion

In this study, a multiclass dataset is used to evaluate the performance of 18 cutting-edge CNN models for weed classification. The models evaluated are AlexNet, Inception v3, ResNet (18, 34, 50, and 101), SqueezeNet (1.0, 1.1), VGG 16, VGG 19, XResNet (18, 34, 50, and 101), and XSEResNet (18, 34, 50, and 101).

The experimental results are reported in three sections, based on pre-trained networks and the proposed technique. The findings should respond to the following questions:
1. Which pre-trained CNN network performs better at recognizing weeds?
2. Does reducing model complexity affect its performance?
3. What is the relationship between classification accuracy and model complexity in weed identification tasks?

A. *Which pre-trained CNN network performs better at recognizing weeds?*

In this section, the performances of all 18 pre-trained individual CNN networks are presented. Fig. 9 displays the accuracy of all the models. The Inception v3 model achieved the highest accuracy at 95%, and XSEResNet 50 gave the lowest accuracy at 77%. The SqueezeNet models had juxtaposing results. SqueezeNet 1.0 did reasonably well, with an accuracy of 87%; however, SqueezeNet 1.1, a more compact version, achieved just 81%. This performance gap can be explained by SqueezeNet 1.1's considerable reduction in parameters and processing time, which may limit its capacity to generalize effectively. The performance gain from other models implies that deeper architectures are better suited for this kind of classification since they can use more layers to understand complicated patterns in the data. Table 3 shows the precision, recall, and F1 scores generated by the models for each class. Among all designs, Inception v3 performed the best, with an average precision of 96%. Notably, Inception v3, VGG 16, and VGG 19 identified the SP and SFC classes with 100% accuracy. Furthermore, ResNet 101, XResNet (18, 34, 50, 101), and AlexNet obtained 100% precision in the SP class alone. In contrast, XSEResNet 50 had the lowest overall precision (77.16%), while XSEResNet 18 had the lowest precision in the black grass (BG) class (48%). BG consistently emerged as the most error-prone class across all models, while SP attained 100% precision across nine models. Inception v3 continued to outperform in terms of recall and F1-scores, with average recall and F1-scores of 94.16% and 95%, respectively. ResNet 50, ResNet 101, XResNet (18, 34, 50, 101), Inception v3, VGG 16, and VGG 19 models achieved 100% accuracy in the SFC class. VGG 16 and VGG 19 both obtained 100% recall for the SFC, SP, and CK classes. At the other end of the spectrum, SqueezeNet 1.1 and XSEResNet 34 had the lowest recall values (37%), with BG being the most difficult class to recall, followed by LSB. In terms of F1-scores, VGG 16 generated 100% F1-scores for the SP and SFC classes, while VGG 19 achieved 100% F1 for the SFC class. For the BG class, which was the most misclassified of all classes, Inception v3, XResNet 101, and Inception v3 achieved 91% precision, 82% recall, and 82% F1-scores. The XSEResNet variations fared badly overall, most likely because they relied on huge training datasets to effectively learn representations. Despite having limited training data, many models struggled to obtain high classification accuracy. Furthermore, architectural constraints and incorrectly calibrated hyperparameters—such as learning rate, batch size, or regularization settings—could have harmed their performance. The results show that certain classes, such as SP and SFC, were consistently categorized with high precision and recall, whereas BG and LSB presented major hurdles to all models. This emphasizes the importance of class-specific data augmentation and improved model designs in addressing such imbalances successfully. Fig 10 provides the confusion matrices for all the models.

(in %)	BG	CK	FH	LSB	SP	SFC
AlexNet						
Precision	77	99	99	86	100	99
Recall	61	99	98	93	99	100
F1-score	65	99	98	89	99	100
Inception v3						
Precision	91	100	99	86	100	100
Recall	74	99	98	96	98	100
F1-score	82	99	99	91	99	100
ResNet 18						
Precision	60	97	91	81	95	95
Recall	52	96	96	84	87	99
F1-score	56	96	93	82	91	97
ResNet 34						
Precision	58	92	90	83	98	97
Recall	54	97	96	83	85	99
F1-score	56	95	93	83	91	98

(in %)	BG	CK	FH	LSB	SP	SFC
ResNet 50						
Precision	53	99	98	82	100	97
Recall	58	99	98	79	96	100
F1-score	56	99	98	80	98	98
ResNet 101						
Precision	66	97	96	83	100	98
Recall	58	99	98	88	90	100
F1-score	62	98	97	86	94	99
SqueezeNet 1.0						
Precision	58	92	90	83	98	97
Recall	54	97	96	83	85	99
F1-score	56	95	93	83	91	98
SqueezeNet 1.1						
Precision	54	90	73	78	94	91
Recall	37	89	88	84	72	97
F1-score	44	89	80	81	81	93
VGG 16						
Precision	62	99	99	85	100	100
Recall	61	100	98	85	100	100
F1-score	62	99	98	85	100	100
VGG 19						
Precision	69	95	98	85	100	100
Recall	63	100	97	88	97	100
F1-score	66	97	97	87	98	100
XResNet 18						
Precision	49	93	84	86	100	89
Recall	70	94	92	67	73	100
F1-score	58	94	88	76	84	94
XResNet 34						
Precision	52	92	88	85	100	92
Recall	66	94	95	74	78	100
F1-score	58	93	91	79	87	96
XResNet 50						
Precision	52	92	88	85	100	92
Recall	66	94	95	74	78	100
F1-score	58	93	91	79	87	96
XResNet 101						
Precision	59	92	92	90	100	92
Recall	82	96	94	75	82	100
F1-score	69	94	93	82	90	96
XSEResNet 18						
Precision	48	90	83	79	94	83
Recall	48	93	82	75	75	98
F1-score	48	92	82	77	83	90
XSEResNet 34						
Precision	54	90	73	78	94	91
Recall	37	89	88	84	72	97
F1-score	44	89	80	81	81	93
XSEResNet 50						
Precision	53	87	80	74	90	79
Recall	39	86	80	81	66	93
F1-score	45	87	80	77	76	86
XSEResNet 101						
Precision	60	85	81	75	89	79

(in %)	BG	CK	FH	LSB	SP	SFC
Recall	39	93	77	84	60	94
F1-score	47	89	79	79	71	86

TABLE III. Classification metrics for 18 models

B. Does reducing model complexity affect its performance?

Table 4 summarizes the metrics obtained using the simplified Inception v3 model. The accuracy, as indicated in equation (1), was calculated as the proportion of correctly classified samples to the total number of samples. The simplified Inception v3 model was created based upon the classification results from 18 previously trained models. Among these models, Inception v3 displayed the highest accuracy, prompting its selection as the foundation for future investigation. Inception v3 beat other models thanks to its sophisticated design and optimisation methodologies. Its inception modules used multi-scale convolutional filters to successfully collect features at various levels, while factorized convolutions decreased computing costs without sacrificing performance. Batch normalization improved training stability and minimized overfitting. Furthermore, methods such as dimensionality reduction via 1x1 convolutions and robust regularization enhanced generalization. The goal was to investigate the effects of lowering the model's complexity while retaining classification accuracy. The proposed model achieved the highest accuracy of 98%. The accuracy improvement from the original network to the simplified network is notable due to the following actions:

- Filter Reduction
- Auxiliary Classifier Removal
- 3x3 Split Convolution
- Block Removal

Model	Accuracy	Precision	Recall	F1-score
Proposed	96	96.5	95.6	95.8

TABLE IV. Classification report

C. What is the relationship between classification accuracy and model complexity in weed identification tasks?

The proposed model has the highest accuracy (96%) of all models. It reduces complexity by about 37.5% over Inception v3, which previously had the best performance-to-complexity ratio. ResNet 101 (90%) and XResNet 101 (82%) exhibit much lower accuracy than the proposed model. Despite its reduced complexity (∼ 15M), the proposed model outperforms other models in all metrics. VGG 16 and VGG 19 achieve 91% accuracy, but need approximately 10× the complexity ∼ 138M and ∼ 143M) compared to the proposed model. This highlights the inefficiency of VGG architectures in this task. SqueezeNet remains the lightest (1.2M-1.3M parameters); however, it loses 17% accuracy compared to the proposed model. Although the proposed model is highly efficient, it may face deployment issues on extremely resource-constrained devices (e.g., microcontrollers) when compared to ultra-lightweight designs such as SqueezeNet. In comparison to Inception v3, it improves accuracy by 3.16%, making it better than the original Inception v3, suited for real-world deployment. Models such as SqueezeNet could still be explored for ultra-lightweight applications, but they would result in considerable performance losses.

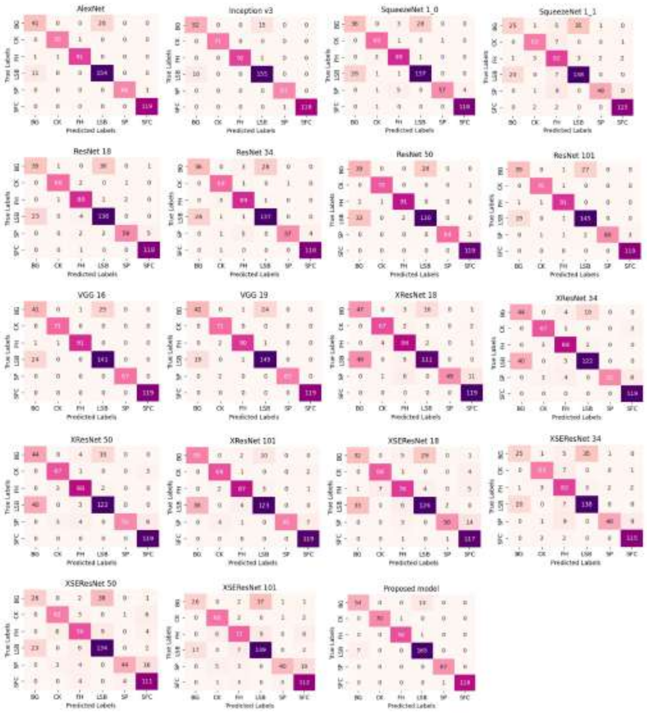

Figure 10. Confusion matrices for all the pre-trained and proposed models

7. Conclusion

Deep learning models are proficient in weed management owing to their capacity to discern features and patterns in images. This study examines the performance of 18 CNN models (lightweight and dense) for weed classification. The findings of this study suggest that Inception v3, a dense model, produced the best results, with 95% accuracy, 96% precision, and 94.16% recall. However, it is computationally expensive. To address this, a simplified version of Inception v3 that reduces the trainable parameters by 37.5% to 15 million is proposed. The modified version was constructed by reducing filters, removing auxiliary classifiers, and splitting large kernels into smaller ones. The simplified model outperformed all the other 18 models, with 96% accuracy, 96.5% precision, and 95.6% recall. In comparison, ResNet50 in [53] scored 95.23%. However, there are several limitations to the study: employing the free resources available in Google Colab limits the depth of research; using publicly available datasets instead of data taken from the fields; and lacking information on the contributions made by fine-tuning the hyperparameters. Future research will focus on further decreasing model size, enhancing the performance of lightweight CNN models, and investigating deployment on edge devices.

Data Availability

The data used in this work is publically available at Kaggle website: https://www.kaggle.com/datasets/vbookshelf/v2-plant-seedlings-dataset

References

[1] United Nations, “World population projected to reach 9.7 billion by 2050,” https://www.un.org/development/desa/en/news/population/world-population-prospects-2019.html, 2024, accessed: 2024-05-01.

[2] H. S. Pathak, P. Brown, and T. Best, “A systematic literature review of the factors affecting the precision agriculture adoption process,” *Precision Agriculture*, vol. 20, pp. 1292–1316, 2019.

[3] F. J. Pierce and P. Nowak, “Aspects of precision agriculture,” *Advances in agronomy*, vol. 67, pp. 1–85, 1999.

[4] S. Tshewang, B. M. Sindel, M. Ghimiray, and B. S. Chauhan, “Weed management challenges in rice (oryza sativa l.) for food security in bhutan: A review,” *Crop Protection*, vol. 90, pp. 117–124, 2016.

[5] G. Adeux, E. Vieren, S. Carlesi, P. Bàrberi, N. Munier-Jolain, and S. Cordeau, “Mitigating crop yield losses through weed diversity,” *Nature Sustainability*, vol. 2, no. 11, pp. 1018–1026, 2019.

[6] Liebman, “Integration of soil, crop and weed management in low-external-input farming systems,” *Weed research*, vol. 40, no. 1, pp. 27–47, 2000.

[7] C. Délye, M. Jasieniuk, and V. Le Corre, “Deciphering the evolution of herbicide resistance in weeds,” *Trends in Genetics*, vol. 29, no. 11, pp. 649–658, 2013.

[8] L. N. Smith, A. Byrne, M. F. Hansen, W. Zhang, and M. L. Smith, “Weed classification in grasslands using convolutional neural networks,” in *Applications of Machine Learning*, vol. 11139. SPIE, 2019, pp. 334–344.

[9] A. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, and M. G. Jones, “A survey of deep learning techniques for weed detection from images,” *Computers and electronics in agriculture*, vol. 184, p. 106067, 2021.

[10] K. Raju, B. Chinna Rao, K. Saikumar, and N. Lakshman Pratap, “An optimal hybrid solution to local and global facial recognition through machine learning,” *A fusion of artificial intelligence and internet of things for emerging cyber systems*, pp. 203–226, 2022.

[11] D. S. Dhaliwal and M. M. Williams, “Sweet corn yield prediction using machine learning models and field-level data,” *Precision Agriculture*, vol. 25, no. 1, pp. 51–64, 2024.

[12] V. Gautam, R. K. Ranjan, P. Dahiya, and A. Kumar, “Esdnn: A novel ensembled stack deep neural network for mango leaf disease classification and detection,” *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 10 989–11 015, 2024.

[13] A. Bakhshipour, “Cascading feature filtering and boosting algorithm for plant type classification based on image features,” *IEEE Access*, vol. 9, pp. 82 021–82 030, 2021.

[14] A. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, and M. G. Jones, “A survey of deep learning techniques for weed detection from images,” *Computers and electronics in agriculture*, vol. 184, p. 106067, 2021.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[16] J. Redmon, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[17] S. Andert, “The method and timing of weed control affect the productivity of intercropped maize (zea mays l.) and bean (phaseolus vulgaris l.),” *Agriculture*, vol. 11, no. 5, p. 380, 2021.

[18] L. Morin, “Progress in biological control of weeds with plant pathogens,” *Annual review of phytopathology*, vol. 58, no. 1, pp. 201–223, 2020.

[19] A. H. Van Bruggen, M. Finckh, M. He, C. Ritsema, P. Harkes, D. Knuth, and V. Geissen, “Indirect effects of the herbicide glyphosate on plant, animal and human health through its effects on microbial communities,” *Frontiers in Environmental Science*, vol. 9, p. 763917, 2021.

[20] C. Stuligross and N. M. Williams, “Past insecticide exposure reduces bee reproduction and population growth rate,” *Proceedings of the National Academy of Sciences*, vol. 118, no. 48, p. e2109909118, 2021.

[21] C. Shackleton, R. H. Ali, and T. A. Khan, “Enhancing rangeland weed detection through convolutional neural networks and transfer learning,” *Crop Design*, p. 100060, 2024.

[22] A. Subeesh and C. Mehta, “Automation and digitization of agriculture using artificial intelligence and internet of things,” *Artificial Intelligence in Agriculture*, vol. 5, pp. 278–291, 2021.

[23] J. Su, D. Yi, M. Coombes, C. Liu, X. Zhai, K. McDonald-Maier, and W.-H. Chen, “Spectral analysis and mapping of blackgrass weed by leveraging machine learning and uav multispectral imagery,” *Computers and Electronics in Agriculture*, vol. 192, p. 106621, 2022.

[24] X. Zhao, X. Wang, C. Li, H. Fu, S. Yang, and C. Zhai, “Cabbage and weed identification based on machine learning and target spraying system design,” *Frontiers in Plant Science*, vol. 13, p. 924973, 2022.

[25] K. Joshitha, S. Nishanthini, B. Harshini, and T. Sadhana, “Raspberry pi based crop and weed classification using machine learning algorithm,” in *IOP Conference Series: Earth and Environmental Science*, vol. 980, no. 1. IOP Publishing, 2022, p. 012029.

[26] B. M. Moreno and P. E. Cruvinel, “Computer vision system for identifying on farming weed species,” in *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. IEEE, 2022, pp. 287–292.

[27] K. Ota, J. Y. L. Kasahara, A. Yamashita, and H. Asama, “Weed and crop detection by combining crop row detection and k-means clustering in weed infested agricultural fields,” in *2022 IEEE/SICE*

International Symposium on System Integration (SII). IEEE, 2022, pp. 985–990.

[28] H. Pathak, C. Igathinathane, K. Howatt, and Z. Zhang, “Machine learning and handcrafted image processing methods for classifying common weeds in corn field,” *Smart Agricultural Technology*, vol. 5, p. 100249, 2023.

[29] H. S. El Imanni, A. El Harti, E. M. Bachaoui, H. Mouncif, F. Eddassouqui, M. A. Hasnai, and M. I. Zinelabidine, “Multispectral uav data for detection of weeds in a citrus farm using machine learning and google earth engine: Case study of morocco,” *Remote Sensing Applications: Society and Environment*, vol. 30, p. 100941, 2023.

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

[31] H. Peng, Z. Li, Z. Zhou, and Y. Shao, “Weed detection in paddy field using an improved retinanet network,” *Computers and Electronics in Agriculture*, vol. 199, p. 107179, 2022.

[32] A. Tannouche, A. Gaga, M. Boutalline, and S. Belhouideg, “Weeds detection efficiency through different convolutional neural networks technology,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, p. 1048, 2022.

[33] F. Garibaldi-Márquez, G. Flores, D. A. Mercado-Ravell, A. Ramírez-Pedraza, and L. M. Valentín-Coronado, “Weed classification from natural corn field-multi-plant images based on shallow and deep learning,” *Sensors*, vol. 22, no. 8, p. 3021, 2022.

[34] B. Maram, S. Das, T. Daniya, and R. Cristin, “A framework for weed detection in agricultural fields using image processing and machine learning algorithms,” in *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP).* IEEE, 2022, pp. 1–6.

[35] X. Jin, M. Bagavathiannan, A. Maity, Y. Chen, and J. Yu, “Deep learning for detecting herbicide weed control spectrum in turfgrass,” *Plant Methods*, vol. 18, no. 1, p. 94, 2022.

[36] X. Jin, M. Bagavathiannan, P. E. McCullough, Y. Chen, and J. Yu, “A deep learning-based method for classification, detection, and localization of weeds in turfgrass,” *Pest Management Science*, vol. 78, no. 11, pp. 4809–4821, 2022.

[37] A. H. Al-Badri, N. A. Ismail, K. Al-Dulaimi, A. Rehman, I. Abunadi, and S. A. Bahaj, “Hybrid cnn model for classification of rumex obtusifolius in grassland,” *IEEE Access*, vol. 10, pp. 90 940–90 957, 2022.

[38] A. Nasiri, M. Omid, A. Taheri-Garavand, and A. Jafari, “Deep learning-based precision agriculture through weed recognition in sugar beet fields,” *Sustainable computing: Informatics and systems*, vol. 35, p. 100759, 2022.

[39] S. Veeragandham and H. Santhi, “Optimization enabled deep quantum neural network for weed classification and density estimation,” *Expert Systems with Applications*, vol. 243, p. 122679, 2024.

[40] M. Dadashzadeh, Y. Abbaspour-Gilandeh, T. Mesri-Gundoshmian, S. Sabzi, and J. I. Arribas, “A stereoscopic video computer vision system for weed discrimination in rice field under both natural and controlled light conditions by machine learning,” *Measurement*, p. 115072, 2024.

[41] N. Razfar, J. True, R. Bassiouny, V. Venkatesh, and R. Kashef, “Weed detection in soybean crops using custom lightweight deep learning models,” *Journal of Agriculture and Food Research*, vol. 8, p. 100308, 2022.

[42] N. Rai, Y. Zhang, M. Villamil, K. Howatt, M. Ostlie, and X. Sun, “Agricultural weed identification in images and videos by integrating optimized deep learning architecture on an edge computing technology,” *Computers and Electronics in Agriculture*, vol. 216, p. 108442, 2024.

[43] A. M. Hasan, D. Diepeveen, H. Laga, M. G. Jones, and F. Sohel, “Object-level benchmark for deep learning-based detection and classification of weed species,” *Crop Protection*, vol. 177, p. 106561, 2024.

[44] G. A. Mesías-Ruiz, I. Borra-Serrano, J. Peña, A. I. de Castro, C. Fernández-Quintanilla, and J. Dorado, “Weed species classification with uav imagery and standard cnn models: Assessing the frontiers of training and inference phases,” *Crop Protection*, vol. 182, p. 106721, 2024.

[45] VBOOKSHELF, “V2 plant seedlings dataset, version 1,” https://www.kaggle.com/datasets/vbookshelf/v2-plant-seedlings-dataset, 2018, accessed: 2024-05-05.

[46] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[48] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[49] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[50] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 558–567.

[51] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[52] F. N. Iandola, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.

[53] K. Gupta, R. Rani, and N. K. Bahia, “Plant-seedling classification using transfer learning-based deep convolutional neural networks,” *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, vol. 11, no. 4, pp. 25–40, 2020.