

Article

From Plants to Pixels: The Role of Artificial Intelligence in Identifying *Sericea Lespedeza* in Field-Based Studies

Aftab Siddique ^{1,*}, Kyla Cook ¹, Yasmin Holt ¹, Sudhanshu S. Panda ², Ajit K. Mahapatra ¹, Eric R. Morgan ³, Jan A. van Wyk ⁴ and Thomas H. Terrill ¹

¹ Department of Agricultural Sciences, Fort Valley State University, 1005 State University Drive, Fort Valley, GA 31030, USA; kcook12@wildcat.fvsu.edu (K.C.); yholt@wildcat.fvsu.edu (Y.H.); mahapatraa@fvsu.edu (A.K.M.); terrillt@fvsu.edu (T.H.T.)

² Institute for Environmental Spatial Analysis, University of North Georgia, 3820 Mundy Mill Road, Oakwood, GA 30566, USA; sudhanshu.panda@ung.edu

³ Institute for Global Food Security, Queen's University, University Road, Belfast BT7 1NN, UK; eric.morgan@qub.ac.uk

⁴ Department of Veterinary Tropical Diseases, Faculty of Veterinary Science, University of Pretoria, Private Bag x04, Onderstepoort 0110, South Africa; jan.vanwyk@up.ac.za

* Correspondence: aftab.siddique@fvsu.edu; Tel.: +1-478-287-9001

Abstract: The increasing use of convolutional neural networks (CNNs) has brought about a significant transformation in numerous fields, such as image categorization and identification. In the development of a CNN model to classify images of sericea lespedeza [SL; *Lespedeza cuneata* (Dum-Cours) G. Don] from weed images, four architectures were explored: CNN model variant 1, CNN model variant 2, the Visual Geometry Group (VGG16) model, and ResNet50. CNN model variant 1 (batch normalization with adjusted dropout method) demonstrated 100% validation accuracy, while variant 2 (RMSprop optimization with adjusted learning rate) achieved 90.78% validation accuracy. Pre-trained models, like VGG16 and ResNet50, were also analyzed. In contrast, ResNet50's steady learning pattern indicated the potential for better generalization. A detailed evaluation of these models revealed that variant 1 achieved a perfect score in precision, recall, and F1 score, indicating superior optimization and feature utilization. Variant 2 presented a balanced performance, with metrics between 86% and 93%. VGG16 mirrored the behavior of variant 2, both maintaining around 90% accuracy. In contrast, ResNet50's results revealed a conservative approach for class 0 predictions. Overall, variant 1 stood out in performance, while both variant 2 and VGG16 showed balanced results. The reliability of CNN model variant 1 was highlighted by the significant accuracy percentages, suggesting potential for practical implementation in agriculture. In addition to the above, a smartphone application for the identification of SL in a field-based trial showed promising results with an accuracy of 98–99%. The conclusion from the above is that a CNN model with batch normalization has the potential to play a crucial role in the future in redefining and optimizing the management of undesirable vegetation.

Keywords: convolutional neural network; weight decay; learning rate; sericea lespedeza



Citation: Siddique, A.; Cook, K.; Holt, Y.; Panda, S.S.; Mahapatra, A.K.; Morgan, E.R.; van Wyk, J.A.; Terrill, T.H. From Plants to Pixels: The Role of Artificial Intelligence in Identifying *Sericea Lespedeza* in Field-Based Studies. *Agronomy* **2024**, *14*, 992. <https://doi.org/10.3390/agronomy14050992>

Academic Editor: Yang Zhu

Received: 13 April 2024

Revised: 6 May 2024

Accepted: 7 May 2024

Published: 8 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sericea lespedeza (SL; *Lespedeza cuneata*) is a warm-season perennial legume that has gained attention in small ruminant production for its nutritive value and anthelmintic properties. This plant, native to eastern Asia, was introduced to the U.S. in the early 20th century for erosion control and has since established itself not only to this end, but also as a beneficial forage resource [1], particularly with small ruminants (sheep and goats). The agronomic benefits of SL of drought tolerance and the ability to grow on acidic, infertile soils where other legumes cannot thrive have been known for over a century [2], but recent research has revealed that it also has value not only as a nutraceutical forage, but

particularly also as biosecurity for its bioactivity against internal parasite infection in sheep and goats [3]. This bioactivity, which has been attributed to the high level of prodelphinidin-type condensed tannins in SL [4,5], also includes antibacterial properties [6], reduction in ruminal methane production [7,8] and stress reduction [9–11]. However, the value of SL as a natural (non-chemical) dewormer has been the primary driver for the recent renewed interest in the plant by producers and scientists.

For proprietors keen on amplifying biodiversity, incorporating SL is an informed step towards encouraging a dynamic and diverse biotic milieu. While the merits of SL in agricultural and ecological contexts are clear, innovative tools are needed to identify and manage it more effectively. This is particularly true during SL establishment, which can be challenging due to its small seed size and relatively slow initial growth [2]. Competition with weeds can be an issue during SL establishment, which may compete with SL during growth stages, and this inability to differentiate SL plants from weeds has led to the erroneous assumption by producers that they have a planting failure when this may not be the case [11].

According to Buchanan and Burns [12], due to having small seeds, producing weak seedlings, and usually being established during early spring, SL finds strong competition from natural weeds in the field when conditions favor their rapid growth. As a summer perennial, SL becomes dormant after a killing frost in the fall, and then regrows from the roots in the spring, so weeds can also be a challenge at this time in well-established stands. Because of SL's poor initial competitive ability with weeds, weed control becomes a major problem during establishment (first year) and thereafter (subsequent years in spring), so identifying weeds in SL fields and eradicating them early is a major necessity.

Despite the importance of the above, there has been a lack of studies specifically aimed at differentiating between SL and weeds. Previous investigations to this end have employed many remote sensing methods [13–15] such as satellite remote sensing, aerial images [16], and microsatellite imaging to observe SL. This research group has also made use of a combination of Landsat satellite imagery to identify SL [16]. However, the identification of weeds in SL pastures has not been studied and requires research on image classification techniques specifically tailored to identifying weeds in SL fields during the crucial establishment phase and during spring re-emergence after winter dormancy in established stands. This approach would provide more accurate and timely interventions, enabling site-specific crop management (SSCM) that is both efficient and cost-effective. By implementing this customized approach, farmers can utilize precise weed control methods, hence improving the successful development of sustainable agriculture and enhancing overall crop management.

With the advent of machine learning (ML) and the broader field of artificial intelligence (AI), innovative solutions are emerging to detect and manage certain invasive plant species that are considered weeds. Early studies, such as by Turner et al. [17], highlighted the potential of remote sensing data in combination with ML to identify specific plant species. These methods use spectral data to provide high accuracy rates in diverse landscapes [17]. In addition, convolutional neural networks (CNNs), a class of deep learning algorithms, have frequently been applied in recent years for plant species identification. For instance, Zhang et al. [18] used CNNs to detect invasive plant species from aerial imagery, demonstrating their potential applicability to remote-sensing-based image classification techniques. Several domain-specific studies have been conducted to identify plants and plant-related diseases using AI, including CNN models, with the use of a variety of AI-assisted ML techniques for the identification of different diseases and adverse plant conditions [19–22]. Concerning SL, specific Arc-GIS-based models have been developed to identify production sites to cultivate SL with low to minimum resource input methodologies [16,23]. However, based on a thorough literature search, there has been no previous research conducted on developing an AI-based ML model to differentiate weeds from SL in field-based studies.

2. Materials and Methods

2.1. Field Image Acquisition, Preparation, and Mobile App Development

During the process of image collection of SL and weeds, natural light conditions were used for field collection during daytime, regardless of light conditions, whether sunny or cloudy. No resolution feature was used during the data curation process, with distances from the plant varying from 10 cm to 20 cm. In order to avoid blurredness, careful consideration was given to capturing specific features of the targeted SL.

As a first step in the mobile application development process, a total of 5000 smartphone images, taken with a Samsung Galaxy A54 (Samsung Electronics, Ridgefield Park, NJ, USA), were selected, comprising 1000 of SL and 4000 of different weeds and plants, and a combination of weeds and SL was selected in developing the smartphone application for SL identification. To execute all the detection steps, starting from image acquisition through results dissemination, the smartphone application (SL app) was developed using the saved trained android studio software 2022.2.1 (Version: Flamingo) model. By training the detection algorithm with a large dataset of SL images, a functional relationship was established and subsequently classified for the screening of the SL plants from weeds and other plants under field conditions. Additionally, the expectation is that the use of cell-phones other than the Samsung Galaxy A54 will yield similar results due to the procedure of normalizing the image data.

2.2. Image Classification Model Selection

The present study used convolutional neural networks (CNNs), a regularized type of feed-forward neural network similar to the widely used backpropagation neural network (BPNN), which is specifically applied for image-classification-based studies. Randomized weights are provided to input and output neurons in the BPNN, making network learning take much longer to achieve the best-fit correlation, and vanishing gradient and exploding gradient situations may occur [24]. Therefore, in CNNs, these occurrences are prevented by using regularized weights over fewer connections [24]. Therefore, CNNs are known as shift-invariant or space-invariant artificial neural networks (SIANNs) [25], since in the network a designated group of neurons shares fixed or regularized weights provided by users, and it works better in image classification, similar to the self-organizing map (SOM) ANN image classification approach involving a fixed window of pixels 3×3 , 5×5 , or $n \times n$ sizes, but through a feed-forward learning process. Given the complex nature of image data (Figure 1), the choice of methodology greatly influences the effectiveness of such a model in our study. The CNN (SIANN) is based on a stepwise process of delineating the model intricacies, emphasizing those involved at each stage of model development, and showing the pipelines for classification (Figure 2).

2.3. Dataset Stratification

For any image-classification-based study, data segmentation is the most fundamental principle. Recognizing that improperly segmented data can lead to overfitting or generalization issues, a two-way split was adopted, comprising training and validation datasets, with an 80:20 ratio [21,26]. While the training dataset serves as the primary source from which the model learns, the validation set is a tool for fine-tuning the model to ensure it is neither over nor underfitting. The test set, kept separate, evaluates the model's performance in an "unseen" scenario. To further systemize the training process, images within these sets were grouped by their respective classes, ensuring balanced representation and easy accessibility during model training. For the analysis of data, 800 randomly selected images of SL and 3200 images of weeds were used for training; 200 images of SL and 800 images of weeds were used in the model validation. For the testing of the developed models, 500 images from each class were used in this study.



Figure 1. Examples of different sericea lespedeza images (first row in top), weed images (second row in middle), and SL images in between weeds (third row in bottom), a combination used in the development of CNN image-based classification. Different images used in the figure illustration show examples of different colors, lighting conditions, angles of images taken, and distances from plants.

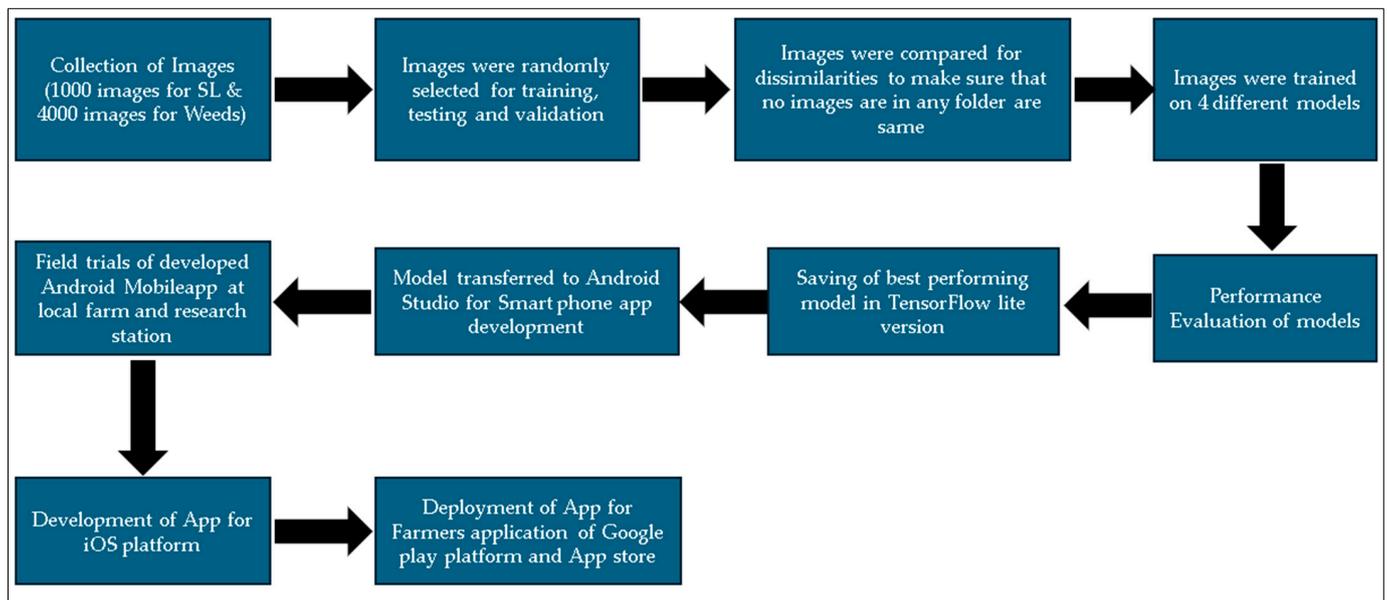


Figure 2. Flow diagram for the pipeline showing development of image classification model and weed identification app.

2.4. Preprocessing Dynamics

Preprocessing digital images, consisting of a grid of pixel values at their core, is essential to ensure uniformity and to enhance training efficacy. To this end, Keras' 'ImageDataGenerator' was used, which allowed for augmentation of the data by introducing minor variations in our images, such as rotations, zooms, and shifts. Such augmentation prevents overfitting and exposes the model to a broader set of data scenarios. Along with

augmentation, we undertook two standard preprocessing steps through, firstly, resizing all images to 150×150 pixels, eliminating the variability in image dimensions, and second, normalizing pixel values, which traditionally range from $[0, 255]$ to $[0, 1]$, to ensure a more tractable data range to facilitate gradient descent optimization during model training. The 'ImageDataGenerator' class from the 'keras.preprocessing.image' library was utilized to enhance the data in the training dataset, hence improving the model's capacity to generalize from our training data through the addition of variability.

This class implements multiple image-transforming operations, such as rotation, where each image can be rotated by a randomly selected angle of up to 20 degrees ($\text{rotation_range} = 20$); width shift, where images can be horizontally displaced by a maximum of 20% of their width ($\text{width_shift_range} = 0.2$); height shift, where the image's original height is vertically shifted by up to 20% ($\text{height_shift_range} = 0.2$); shearing, where the images undergo a shear modification with an intensity of 20% ($\text{shear_range} = 0.2$); zoom, where the images are subjected to a zoom effect with a range of up to 20% ($\text{zoom_range} = 0.2$); horizontal flip, where images have the possibility of being randomly flipped horizontally ($\text{flip_horizontal} = \text{true}$); fill mode, where the "nearest" method is utilized to populate freshly generated pixels that emerge during transformations such as rotations or shifts ($\text{fill_mode} = \text{"nearest"}$).

The ImageDataGenerator library used in python script may apply various augmentations to each image, such as rotations, shifts, zooming, flipping, and shearing, all in a single pass. The use of this dynamic method for data augmentation results in the inability to accurately determine the precise quantity of distinct images produced during training. This is due to the fact that the augmentation process takes place in real time as the model receives the training images. Conversely, the system produces a potentially limitless range of images by implementing random modifications to each image in a group, guided by the given parameters.

To approximate the quantity of augmented images produced during the training phase of the model, the following equation is used: $\text{generated images} = \text{number of images utilized in training} \times \text{number of epochs employed in training}$. For this study, we employed 800 images of SL and 3200 images of weed plants. By conducting 100 training epochs, this method yielded roughly 80,000 images of SL (synthetic labels) and 320,000 images of weeds, resulting in a total of around 400,000 generated images. This approach guarantees that every epoch has the ability to exhibit distinct variations of the input images, hence improving the resilience of the training process.

2.5. CNN Architecture

Unlike traditional neural networks, CNNs excel in detecting image patterns, ranging from simple edges to complex features. Our model's inception layer was designed to process tensors with dimensions of $150 \times 150 \times 3$, indicative of the image's height, width, and RGB channels. The initial convolutional layer, equipped with $32 \times 3 \times 3$ filters and the ReLU activation function, focused on elementary feature detection. Pooling layers, interspersed after convolutional layers, reduced spatial dimensions, condensed information, and aided in computational efficiency. As the network was delved into more deeply, the convolutional layers increased in complexity, with 64 and 128 filters, which focused on intricate patterns within images. The architecture then transitioned from convolutional to dense layers via a flattening layer, which reshaped the 3D output tensor from the last pooling layer into a 1D vector. This vectorized output fed into a dense layer with 512 neurons, culminating in an output layer that used SoftMax activation to categorize the images.

The initial iteration of our customized convolutional neural network (CNN), known as model custom variant 1, was designed using the regularization capabilities of batch normalization and dropout. The architecture consisted of a series of convolutional layers with 3×3 kernel sizes, first employing 32 filters and progressively increasing to 64 and 128 filters. After each convolutional layer, batch normalization was applied to normalize the output of the preceding activation layer. This was conducted by removing the batch

mean and dividing by the batch standard deviation. After each normalization, a 2×2 max pooling procedure was used to decrease the dimensionality of the feature maps, effectively compressing the gained knowledge features and minimizing computing demands. The model additionally used dropout at a rate of 0.4 after the final dense layer before the output layer. Dropout randomly excludes a portion of features during each training phase, thereby reducing overfitting. The constructed model utilized the Adam optimizer alongside a binary cross-entropy loss function, which is well-suited for tasks involving binary classification. The objective of this method was to achieve learning stability and minimize the number of epochs needed for efficient model training.

The second custom model, known as model custom variant 2, followed a similar structure to the previous versions but utilized the RMSprop optimizer with a learning rate of 0.0001. The learning rate was used to evaluate the impact of a slower and more precise modification of the weights on the model's learning and generalization capabilities. The model retained a similar structure to the initial version, but it omitted batch normalization. This was designed to specifically assess the effectiveness of the RMSprop optimizer on its own.

In addition to the new models, the pre-trained models VGG16 and ResNet50 were also modified for the purpose of the classification challenge. By utilizing transfer learning, the base models were imported together with pre-trained weights from the ImageNet dataset. The uppermost layers of these models were substituted with a flattening layer in order to transform the two-dimensional feature maps into a one-dimensional vector. This was succeeded by a dense layer consisting of 512 neurons and a concluding output layer with a sigmoid activation function, which was used for binary classification. During training, the layers imported from the previous models, both VGG16 and ResNet50, were frozen to preserve the learned features. Only the newly added layers were modified to alter their weights. Both the VGG16 and ResNet50 models were constructed using the Adam optimizer and the binary cross-entropy loss function.

2.6. Model Compilation and Training Dynamics

Once the CNN's architecture was solidified, the model's compilation phase commenced. The Adam optimizer was the optimizer of choice, since it dynamically adjusts the learning rate, a feature especially useful for deep learning tasks. Given the categorical nature of the present classification, the loss function employed was categorical cross-entropy. Also, accuracy, indicating the proportion of correctly classified images, was the primary metric throughout the training phase.

Iterative refinement was used for the training of the present model, which adjusted its internal parameters with each epoch, using a complete forward and backward pass of the training examples, honing its prediction capabilities. This iterative process can be likened to repeatedly studying a topic, refining understanding with each pass (iteration). However, since, as with any study regimen, there is a risk of over-studying or "memorizing", a validation set was used as a safeguard against the model "memorizing" training data, i.e., overfitting. This separate dataset, providing a check, was included to ensure that the model generalized well.

2.7. Model Variants

Beyond the baseline CNN model, the research ventured into several model variants. The first variant (CNN model variant 1) was accentuated with batch normalization and adjusted dropout. This technique, applied post-convolution, standardized the outputs, stabilizing and accelerating training, and coupled with a 0.4 dropout rate, also introduced a regularizing effect. The second variant (CNN model variant 2) delved deeper, introducing an extra step, while, similar to the first, it deviated architecturally in its optimization technique. Here, the RMSprop optimizer, known for its adaptive nature, was implemented with a learning rate of 0.0001. Beyond these custom variants, the exploration also incorporated renowned architectures, like the pre-trained models VGG16 and ResNet50, which were

equipped with weights trained on the extensive ImageNet dataset and were appended with custom dense layers to fit the classification task.

Each model variant, including the pre-trained architectures, underwent rigorous training, governed by predefined parameters. Crucial to this process were early stopping callback functions, which halted training if the model ceased to improve, as well as learning rate adjustment, which dynamically tweaked the learning rate based on the model's performance.

2.8. Batch Normalization and Adjusted Dropout Method

Batch normalization (batch norm, BN) is a transformative technique in deep learning, designed to accelerate the training of deep networks and optimize their performance [27]. It addresses the issue of internal covariate shift, where the distribution of network activations changes during training due to parameter adjustments. In practice, batch normalization (Figure 3) computes the mean and variance for each feature in a mini-batch, then normalizes the feature to have a mean of zero and unit variance. This normalized output is further scaled and shifted, using learnable parameters. When integrated into custom CNN models, batch normalization often results in faster convergence, allows for the use of higher learning rates, and makes the model less sensitive to initialization strategies [28–30]. It also imparts a minor regularization effect, adding slight noise that can counteract overfitting. The batch normalization first computes the mini-batch's means (μ) and variance (σ^2) scale and shifts the feature using two new parameters per feature, often called γ (scale) and β (shift) [28,30,31]. These parameters are learned during training alongside the original model parameters, and the mathematical expression is provided in Equation (1) for an input of m :

$$BN = \gamma \left(\frac{m - \mu}{\sqrt{\sigma^2 - \epsilon}} \right) + \beta \quad (1)$$

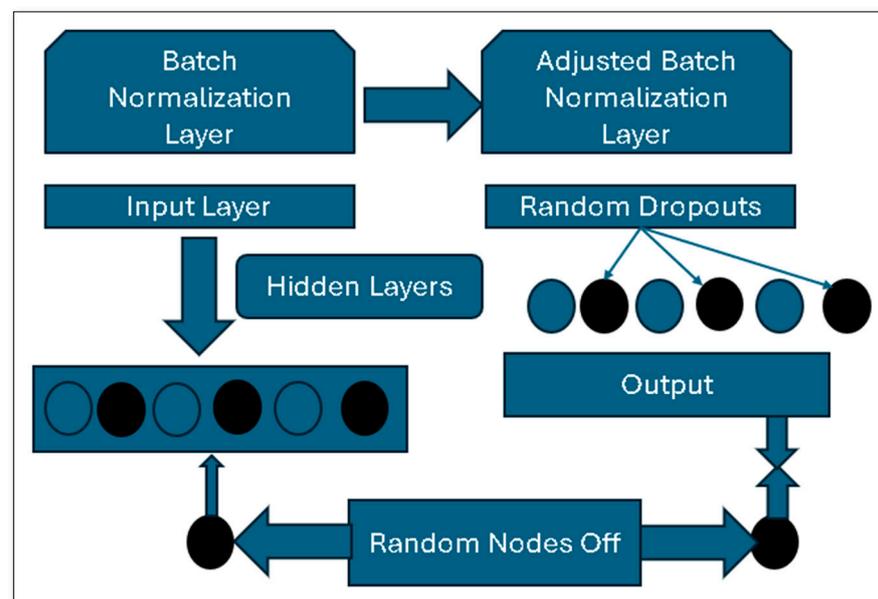


Figure 3. Graphical representation of batch normalization with adjusted dropout method. Black circle represents dropout non-functional neural network nodes.

On the other hand, dropout, another regularization technique designed to combat overfitting, randomly deactivates a fraction of input units during training in each update cycle. For instance, with a dropout rate of 0.5, roughly half of the input units are “turned off” at each step. However, by adjusting this behavior during the inference phase, dropout is disabled, and all units are active, with weights scaled down proportionally to account for the more extensive unit activity. In CNNs, the concept of dropout is often adjusted

for spatial structures. For instance, spatial dropout drops entire channels of feature maps, thereby ensuring consistent absences in the feature map.

In addition to the above, dropout after batch normalization is commonly advised in order to avoid unstable training dynamics. When used in image classification using CNNs, batch normalization and adjusted dropout techniques, like spatial dropout, often empower the network. This combination facilitates training deeper, more robust models that exhibit superior generalization of unseen data.

2.9. RMSprop and Adjusted Learning Rate Method

RMSprop, standing for root mean square propagation, is a sophisticated optimization technique tailored to facilitate and expedite the training process of neural networks, notably CNNs, which dominate the realm of image classification [32,33]. At its core, RMSprop aims to dynamically adjust each parameter's learning rate based on its gradients' historical magnitudes [34]. This method of RMSprop solves the common drawback of a static learning rate, which, if set too high, can cause unstable training with erratic oscillations, and, if too low, results in an agonizingly slow convergence [35,36].

In RMSprop, the moving average of the squared gradient, which is computed, subsequently informs the adjusted learning rate (Figure 4). A decay factor, β , plays a pivotal role in this latter calculation, determining the balance between the influence of the past average and the present gradient. This emphasis on the recent history of gradients aids in navigating complex optimization landscapes, often characterized by deep CNNs with high dimensionality stemming from copious weights and biases and intricate non-linearities, ushered in by activation functions [37–39].

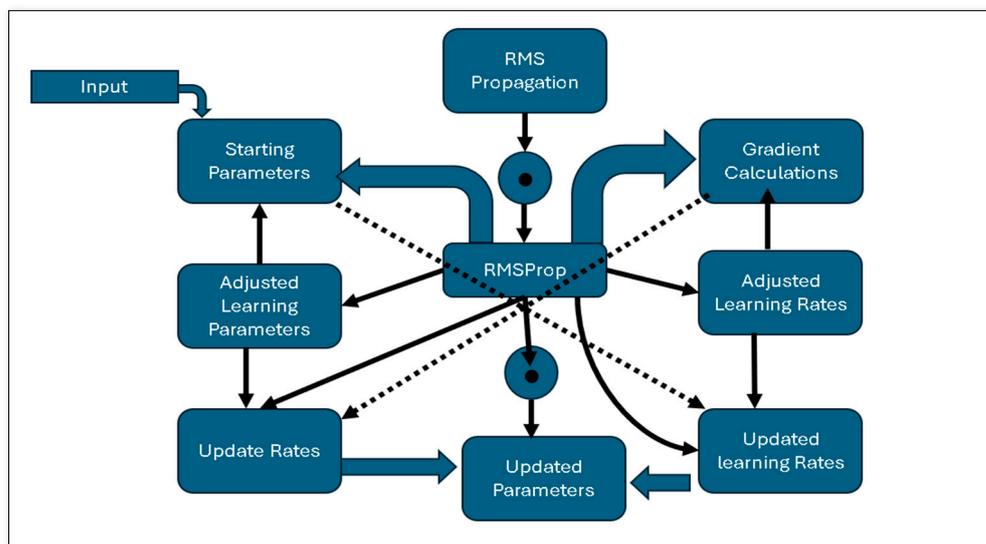


Figure 4. Diagrammatic representation of RMSprop optimization with adjusted dropout method.

In addition to the above, RMSprop proved to be more resilient against the drawback of the vanishing and exploding gradient issues that plague deep networks [40]. However, RMSprop's standard formulation can be further augmented by introducing additional strategies to adjust the global learning rate (η). Methods like step, exponential, and plateau-based decay offer nuanced control, thereby allowing the model to adapt to plateaus or steep regions in the loss landscape. Despite modern deep learning libraries like TensorFlow and PyTorch have simplified their application using RMSprop as a staple optimizer [41–43]. However, it is paramount to recognize the existence of other equally competent optimizers, such as Adam, which combines the strengths of RMSprop with the momentum principle [30]. Ultimately, selecting an optimizer is not a one-size-fits-all decision, but rather a careful choice, often necessitating multi-

ple iterations and experiments tailored to the unique nuances of the specific dataset and problem.

3. Results

In developing the CNN model for distinguishing SL images from weed images, a quartet of architectures, namely CNN model variant 1, CNN model variant 2, VGG16, and ResNet50, presented a spectrum of capabilities and nuances. CNN model variant 1, with the combination of batch normalization and adjusted dropout, showcased an exemplary trajectory, scaling from an initial 87.15% training accuracy to a 99.6% validation accuracy by the 13th epoch (Figure 5). Its resilience was evident in select epochs where validation accuracy peaked at 100%, indicating robustness and the potential for superior generalization.

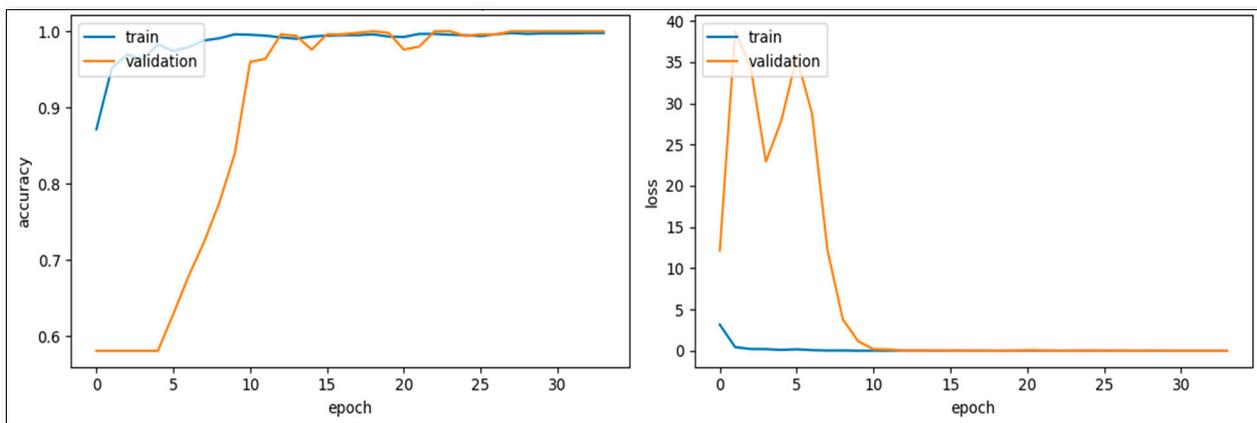


Figure 5. Graphical representation of training and validation accuracies and loss for CNN model variant 1 (batch normalization with adjusted dropout) for SL image datasets.

Contrastingly, CNN model variant 2, driven by the RMSprop optimizer and an adaptive learning rate, embarked on its learning pattern more steadily, but exhibited commendable growth dynamics. Notably, by the 11th epoch, this model logged a 90.78% validation accuracy, showing its rapid maturation towards development. However, the heightened validation loss hinted at potential overfitting nuances particularly intrinsic to RMSprop's behavior in the classification context (Figure 6). The loss graph of the batch normalization optimizer exhibited volatility, especially in the validation loss. The sudden increase in validation loss, followed by a rapid decrease, may indicate the use of a high learning rate or inadequate regularization during the initial stages of training. Nevertheless, the subsequent convergence of the training and validation loss suggests a process of recuperation and adjustment, resulting in a steady and transferable state of learning. This exemplified the optimizer's capacity to regain stability after an initial disruption, maybe by employing techniques such as learning rate scheduling or adaptive learning rate modifications. Finally, the graph of the RMSprop optimizer exhibited a similar pattern of high variance to the batch normalization graph at the beginning. However, it showed a propensity to gradually increase the validation loss over time, following an initial period of convergence with the training loss. The gradual divergence observed here was a warning sign of possible overfitting, as the model began to excessively conform to the training data, resulting in a decline in its performance on the validation set. This emphasizes the importance of closely monitoring the patterns of loss and considering the implementation of early halting to prevent overfitting from becoming more prominent.

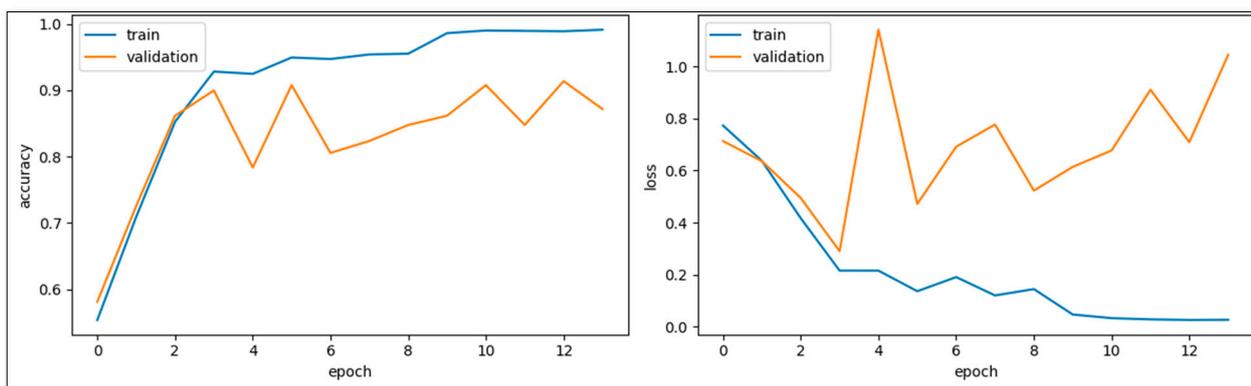


Figure 6. Graphical representation of training and validation accuracies and loss for CNN model variant 2 (RMSprop optimizer with adjusted learning rate) for SL image datasets.

Compared with pre-trained established architectures, such as ResNet50 and VGG16 using transfer learning, VGG16 stood out for its easy convergence, indicating a training accuracy of 77.17% in its initial epochs (Figure 7). However, its rapid assimilation was double-edged, particularly evident from the 11th epoch, suggesting potential overfitting issues. Overfitting arises when a model acquires knowledge of both the fundamental patterns and random variations present in the training data, hence hindering its capacity to make accurate predictions of unseen data [44]. This problem frequently occurs in extremely complicated models with an excessive number of parameters that rapidly adjust to the intricate details of the data. Nevertheless, the presence of a substantial dataset does not guarantee protection against overfitting; the variety and integrity of the data are equally crucial [45]. Even carefully developed models can be adversely affected if regularization techniques such as dropout or weight decay are not appropriately calibrated, or if the learning rate is excessively high, leading to overfitting of the training data and overlooking crucial overarching patterns necessary for validation accuracy [46].

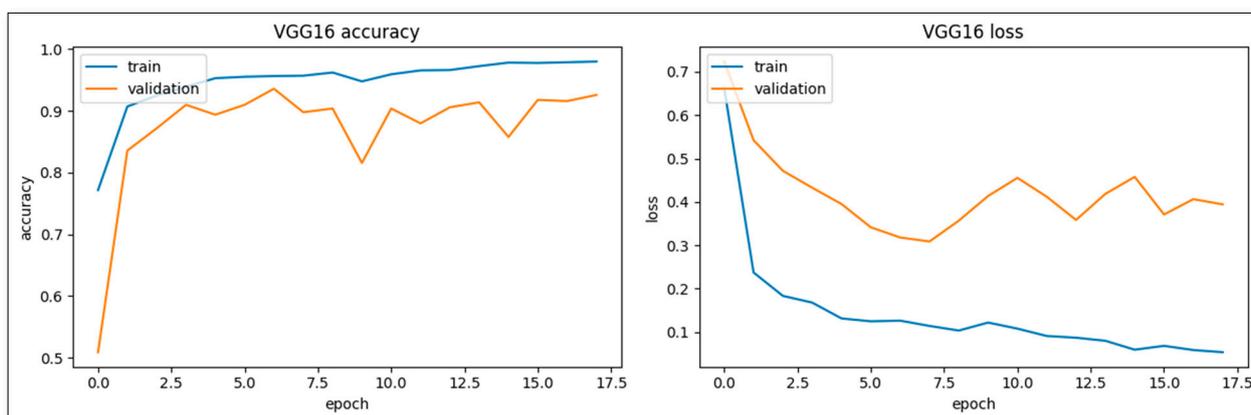


Figure 7. Graphical representation of training and validation accuracies and loss for pre-trained VGG16 model using SL images datasets.

Data leakage can also cause overfitting when information from the validation set influences the training process, while label noise refers to inaccurate data labels that result in poor model learning. Both of these factors may further decrease the performance of the model [47]. In addition, small batch sizes might provide gradients with high levels of noise, which can worsen the problem of overfitting. If the validation set does not accurately represent the wider distribution of data, then it might lead to a misleading indication of overfitting [48].

In order to address the issue of overfitting, it is essential to modify the complexity of the model, use or enhance regularization techniques, fine-tune the learning rate, ensure the quality and diversity of the data, and accurately partition the dataset. Utilizing an iterative methodology for model tuning and validation is crucial in order to successfully identify and solve the underlying factors contributing to overfitting. When assessing the training process of CNN models using loss graphs, the VGG16 and ResNet50 models offered important observations. The loss curves obtained from the VGG16 model exhibited a prototypical example of a training procedure that was well-trained. The training loss exhibited a consistent decline followed by stabilization, indicating that the model was efficiently incorporating the training data. Similarly, the reduction in validation loss was consistent with the decrease in training loss, indicating that the model was effectively applying its learned knowledge to a new, unknown dataset. The narrow discrepancy between the training and validation loss curves during the training epochs highlighted the model's ability to preserve equilibrium, hence avoiding the typical issues of underfitting or overfitting. The objective in training neural networks is frequently to achieve this stability, as it signifies a model that has acquired a resilient representation of the data.

ResNet50, on the other hand, emphasized steadiness. Despite a more deliberate initial pace, its learning pattern was marked by consistency, potentially leading to better generalization, as evidenced by the narrower disparity between its training and validation accuracies (Figure 8). On the other hand, the ResNet50 model had a distinct pattern. The training loss exhibited a continuous decrease, but there remained a persistent disparity between the training and validation losses following the initial severe decrease in the validation loss. This recurring disparity may be seen as the model's robust performance on the training data, but a less successful application to the validation data, suggesting a potential issue of overfitting. This scenario implies that the model may have learned certain unique characteristics of the training data that do not apply well to other situations. This necessitates additional research into approaches for regularization or training methods to improve the effectiveness of the model on validation data.

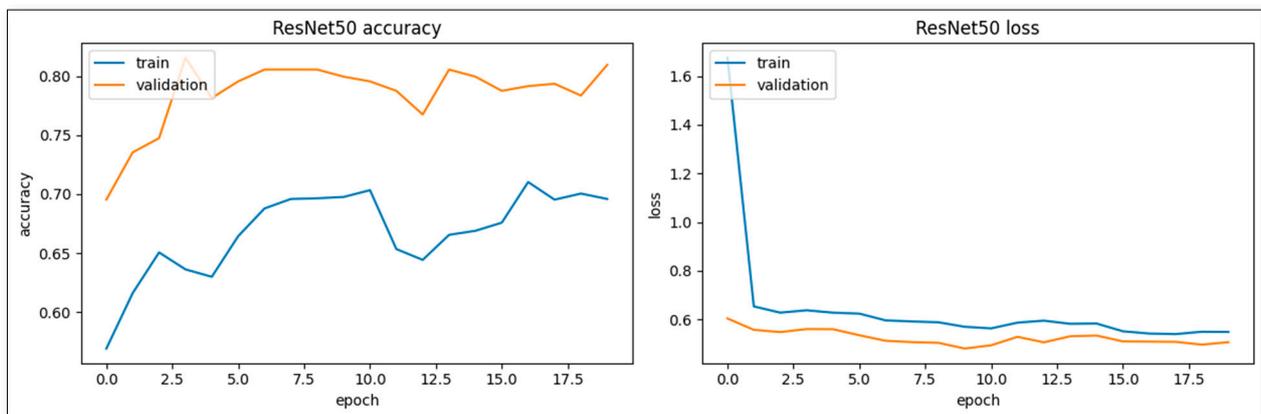


Figure 8. Graphical representation of training and validation accuracies and loss for pre-trained ResNet50 model using SL images datasets.

The presented comparative classification reports in Tables 1 and 2 provide a detailed performance evaluation of four distinct neural network configurations: custom CNN variant 1, custom CNN variant 2, VGG16, and ResNet50. Both tables elaborate on the precision, recall, and F1 score information, offering a multifaceted perspective on each model's capabilities and limitations with and without the use of any data augmentation process in the analysis.

Table 1. Comparative summary of performance matrix for the developed CNN classification model, compared with pre-trained model (VGG16 and ResNet50).

Type of Model	Precision		Recall		F1 Score	
	SL	Weed	SL	Weed	SL	Weed
CNN model variant 1 (batch normalization with adjusted dropout)	100	100	100	100	100	100
CNN model variant 2 (RMSprop with adjusted learning rate)	86	93	91	89	88	91
VGG16 pre-trained model	88	91	88	91	88	91
ResNet50 pre-trained model	87	77	61	93	72	84

Table 2. Comparative summary of performance matrix for the developed CNN classification model without data augmentation, compared with pre-trained model (VGG16 and ResNet50).

Type of Model	Precision		Recall		F1 Score	
	SL	Weed	SL	Weed	SL	Weed
CNN model variant 1 (batch normalization with adjusted dropout)	100	100	100	100	100	100
CNN model variant 2 (RMSprop with adjusted learning rate)	97	100	100	98	99	99
VGG16 pre-trained model	99	91	87	99	92	95
ResNet50 pre-trained model	85	81	69	91	76	86

Tables 1 and 2 show a comparison of the performance metrics for the developed model with and without using data augmentation techniques, namely precision, recall, and F1 score, for two CNN model variants in two categories: SL and weed. The models consisted of a customized convolutional neural network (CNN) with batch normalization and adjusted dropout, referred to as CNN model variant 1, and RMSprop with an adjusted learning rate, known as CNN model variant 2. Additionally, two pre-trained models, VGG16 and ResNet50, were also included in our analysis. CNN model variant 1 demonstrated exact performance in all measures listed in both tables, indicating a well-optimized model for this particular task.

On the other hand, CNN model variant 2 demonstrated a significant enhancement in its performance metrics between with and without data augmentation, suggesting potential improvements in its learning rate modifications or increased training iterations that have strengthened its effectiveness. The VGG16 model, although robust, showed only slight enhancements when no data augmentation was being used and usually performed less effectively compared to the customized models. This suggests that although pre-trained models like VGG16 possess considerable capabilities, they may need substantial adjustments in order to perform exceptionally well in more specific tasks.

However, ResNet50 had the highest level of variability in its performance, specifically facing challenges with the recall measure for SL, both with and without data augmentation. The fact that ResNet50 did not perform well in this particular classification task may indicate that it is not well-suited for this task without significant adjustments or a specialized training approach. The comprehensive analysis highlighted the significance of model selection and customization in attaining the best outcomes in image classification tasks, particularly when dealing with specific categories such as SL and weed. The data absolutely show that customized models or properly fine-tuned pre-trained models can greatly surpass generic models, contingent upon the specific complexity of the task and the peculiarities of the data.

CNN model variant 2 exhibited lower scores in the first table (Table 1), with a precision of 86% for SL and 93% for weed, a recall of 91% for SL and 89% for weed, and an F1 score of 88% for SL and 91% for weed. However, in the second table, there was a significant

enhancement, with precision improving to 97% for SL and 100% for weed, recall reaching 100% for both SL and weed, and the F1 score rising to 99% for SL and weed. The notable enhancement indicated that modifications to the learning rate or even the inclusion of more training data greatly improved the efficacy of the model.

In the first table, the pre-trained model VGG16 demonstrated reasonable performance with precision scores of 88% for SL and 91% for weed, recall scores of 88% for SL and 91% for weed, and F1 scores of 88% for SL and 91% for weed. The second table showed a modest enhancement, with precision rising to 99% for SL and 91% for weed, recall reaching 87% for SL and 99% for weed, and the F1 score reaching 92% for SL and 95% for weed. These modifications suggested a certain level of progress, most likely resulting from the optimization of some aspects related to this particular task. However, they still fell short of achieving the highest level of performance exhibited by the specialized models.

ResNet50 had the highest level of variability and typically performed worse than the other models, especially in the first table where it achieved a precision of 87% for SL and 77% for weed, a recall of 61% for SL and 93% for weed, and an F1 score of 72% for SL and 84% for weed. The second table displayed notable enhancements, with the precision for SL reaching 85% and 81% for weed. The recall for SL stood at 69% and 91% for weed. Additionally, the F1 score for SL was 76%, while for weed it was 86%. Although there were improvements, the model's performance suggests that it may not be suitable for this specific task without substantial adjustments or focused training. The rationale behind not choosing identical hyperparameters for each model comparison stemmed from our intention to investigate the impact of various configurations and regularization procedures on the effectiveness of the developed CNN model. Each variant was specifically tailored to assess a certain component of model training. For example, variant 1 examined the fundamental architecture with batch normalization and a modest dropout rate, whereas variant 2 investigated the impact of using a different optimizer and a lower learning rate on the same architecture as variant 1.

This unorthodox methodology enabled a thorough comprehension of how various modifications of hyperparameters can impact the learning dynamics and generalization abilities of the models. The wide range of configurations also aided in determining the most efficient arrangement for the particular dataset and task, hence facilitating the construction of an optimal model for practical applications.

After the training, testing, and validation of the model developed for the identification of SL in field-based studies, an in-house smart phone application was also developed for the field trials. The mobile application was developed using Android studio software (Version: 2022.2.1, Flamingo). The developed mobile application was successfully able to differentiate SL from weeds up to an accuracy of 98% (Figure 9).

The development of a mobile application not only represented the process of turning theoretical research into practical instruments that can be used in the field, but also demonstrated the flexibility and relevance of the research.

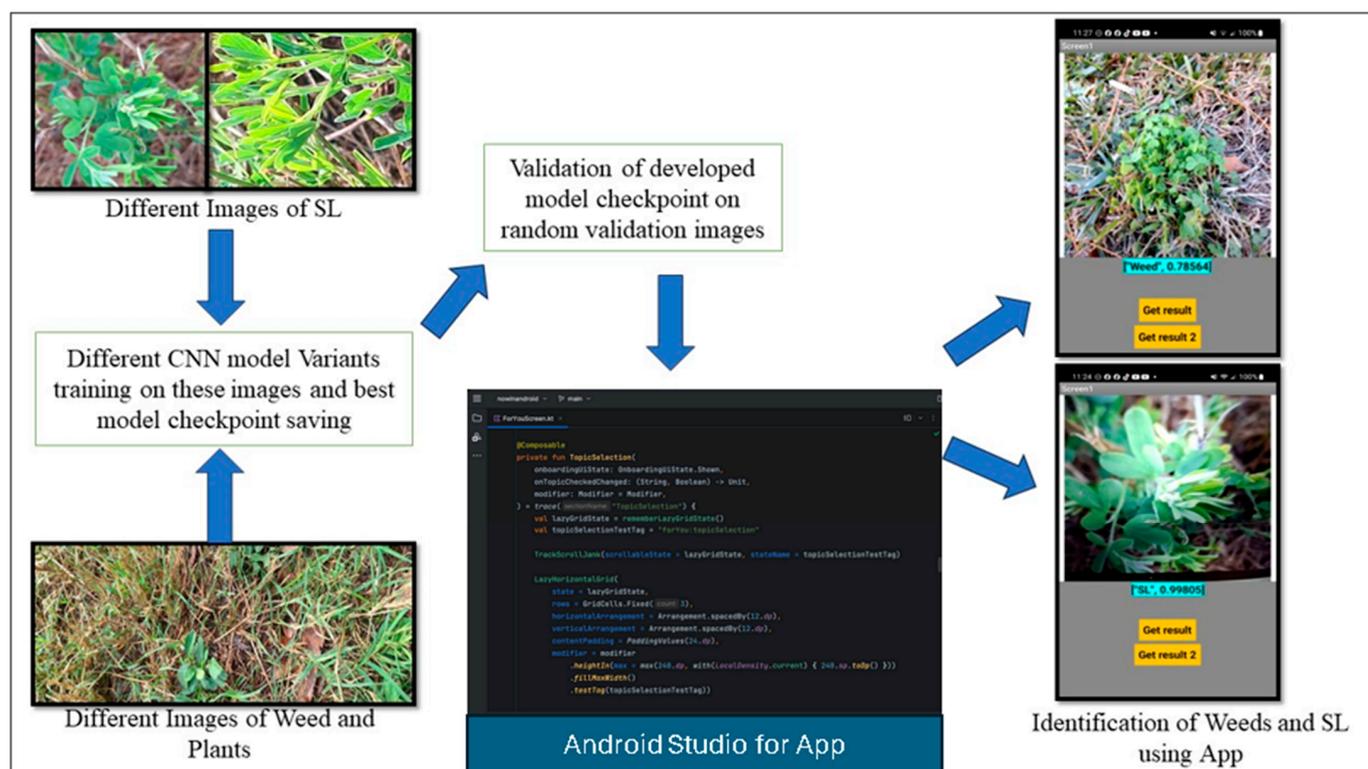


Figure 9. Flowchart and smart phone app results for differentiation of SL from plant weed species.

4. Discussion

According to the results presented in Tables 1 and 2, our custom CNN model variant 1 showed superior performance. It achieved an ideal outcome of 100% for both precision and recall in all scenarios, regardless of whether data augmentation techniques were used. This model was created to accurately detect SL and weeds, demonstrating better performance than the study conducted by Dyrmann et al. [49]. In their study, the recall and precision percentages for separating weeds from winter wheat fields were lower, at 46.3% and 86.6%, respectively. Our models exhibited a diverse range of accuracies in identifying weeds, ranging from 84% to 100%, when compared to earlier studies. The pre-trained models used in our research, with and without data augmentation, achieved average accuracy rates ranging from 84% to 91% and 86% to 95%, respectively. These findings align with the results obtained by Olsen et al. [50], who utilized the Inception-v3 and ResNet50 models on their specific database of weed images.

Gothai et al. [51] conducted a study to investigate the capabilities of different configurations of convolutional neural networks (CNNs). They found that the validation accuracies of several custom CNN models differed. Among the models tested, a non-stride CNN model achieved an accuracy of 93.12% after 40 epochs, while an AlexNet model reached an accuracy of 92.62% at the same epoch count. In addition, different CNN models, which included a growing number of convolutional layers and utilized a pre-trained VGG16 model, achieved accuracies ranging from 92.36% to 96.53% after 40 epochs.

Our advanced models, which combined batch normalization with dropout methods and RMSprop with adjustable learning rates, achieved higher validation accuracy levels of 99.51% and 98.54% after 26 and 15 epochs, respectively. In addition, our VGG16 model achieved a validation accuracy of 94.14% after 28 epochs.

This study highlights the exceptional accuracy, completeness, and overall performance of the generated models, with scores ranging from 85% to 100% under different circumstances. These models demonstrated their efficacy in field-based investigations for the detection of SL and weeds, matching or exceeding the findings of previous research.

Further studies have confirmed the effectiveness of using deep learning algorithms to classify images and accurately identify weeds in turfgrass. The authors Yu et al. [52–54] successfully utilized DetectNet, GoogLeNet, and VGGNet to accurately detect annual bluegrass and different types of broadleaf weeds. Their research emphasized the superior performance of DetectNet, which achieved the greatest F1 scores. Additionally, VGGNet demonstrated commendably good recall and F1 scores in different scenarios. This evidence further reinforces the capacity of sophisticated neural networks to excel in intricate agricultural tasks.

Nevertheless, there are still obstacles to overcome in the realm of deep learning for agriculture, namely regarding the influence of environmental factors like lighting and soil composition, as well as the constraints imposed by datasets lacking sufficiently labeled information [55,56]. Efforts to overcome these obstacles and improve the precision and practicality of weed and forage crop identification models remain a crucial focus of research.

5. Conclusions and Future Studies

Utilizing technology, especially cutting-edge neural network models like the ones examined in our study, will become more essential as the agricultural sector deals with rising global food demand, shrinking arable land, and the unpredictable effects of climate change. For instance, our developed smartphone-based approach of custom CNN variant 1's application with unprecedented accuracy highlights the potential of such models to revolutionize precision agriculture at a time when unmanned aerial vehicle (UAV) technologies are getting smarter day by day. In consequence, the use of these models in conjunction with UAV or satellite photography in real-time applications should be explored further in future studies to provide immediate, field-level insights regarding site-specific crop management (SSCM). This would support farmers in making prompt adjustments and maximizing resource use with real-time insights into weed growth, pest damage, and nutritional deficits, allowing them to optimize resource use and reduce water, fertilizer, and pesticide consumption, thus also helping improve the environment. The model would, for instance, be more broadly applicable if the dataset were expanded to cover a wider range of crops, different growth phases, and various biotic and abiotic stress variables. The discriminative capacity of the model may be further improved by multi-spectral and hyperspectral imaging, even with UAV systems, allowing it to recognize early-stage pest infestations or sub-surface anomalies that are not visible to the naked eye. In this way, a future can be foreseen where farms are more productive, as well as sustainably so, to support a helpful relationship between farmers, technology, and the environment. The transition from desktop-based analytics to a mobile platform represents a notable improvement in the ease of access and usefulness of the research. It enables field researchers or agricultural practitioners to analyze data on site in real time, significantly improving the speed and usefulness of the gathered information. Due to its exceptional precision, the program is a powerful tool in precision agriculture, offering users a dependable means of monitoring and regulating SL in many agricultural environments.

Finally, custom CNN variant 1's accuracy shows how artificial intelligence can improve precision agriculture and SSCM. Real-time applications using such models and UAVs or satellite data warrant additional exploration, and our research is expanding in that direction. We are in the process of developing a real-time UAV imaging and AI-based modeling approach to determine weed infestation in the SL field to spray weedicide on the spot during image acquisition. Expanding the dataset to include more crops, growth phases, and biotic and abiotic stress factors might improve model applicability. Such advances could lead to more productive and sustainably managed farms, benefiting farmers, technological advances, and the surroundings.

Author Contributions: Conceptualization, A.S.; Data curation, K.C. and Y.H.; Formal analysis, A.S.; Funding acquisition, T.H.T.; Investigation, A.S. and A.K.M.; Methodology, A.S. and S.S.P.; Validation, T.H.T.; Writing—original draft, A.S.; Writing—review & editing, S.S.P., A.K.M., E.R.M., J.A.v.W. and T.H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by USDA-National Institute of Food and Agriculture (Capacity Building Grant) award number 2022-38821-37299. The Fort Valley State University undergraduate cohort—contributed to collection of images and preparation of dataset.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Terrill, T.; Mosjidis, J.; Moore, D.; Shaik, S.; Miller, J.; Burke, J.; Muir, J.; Wolfe, R. Effect of pelleting on efficacy of sericea lespedeza hay as a natural dewormer in goats. *Vet Parasitol.* **2007**, *146*, 117–122. [CrossRef]
- Hoveland, C.S.; Windham, W.R.; Boggs, D.L.; Durham, R.G.; Calvert, G.V.; Newsome, J.F.; Dobson, J.W., Jr.; Owsley, M. Sericea lespedeza production in Georgia. *Res. Bull.-Ga. Agric. Exp. Stn.* **1990**. Available online: <https://www.cabidigitallibrary.org/doi/full/10.5555/19900738785> (accessed on 3 March 2024).
- Terrill, T.H.; Miller, J.E.; Burke, J.M.; Mosjidis, J.A.; Kaplan, R.M. Experiences with integrated concepts for the control of *Haemonchus contortus* in sheep and goats in the United States. *Vet Parasitol.* **2012**, *186*, 28–37. [CrossRef]
- Kommuru, D.; Barker, T.; Desai, S.; Burke, J.; Ramsay, A.; Mueller-Harvey, I.; Miller, J.; Mosjidis, J.; Kamiseti, N.; Terrill, T. Use of pelleted sericea lespedeza (*Lepedeza cuneata*) for natural control of coccidia and gastrointestinal nematodes in weaned goats. *Vet Parasitol.* **2014**, *204*, 191–198. [CrossRef]
- Mechineni, A.; Kommuru, D.; Gujja, S.; Mosjidis, J.; Miller, J.; Burke, J.; Ramsay, A.; Mueller-Harvey, I.; Kannan, G.; Lee, J.; et al. Effect of fall-grazed sericea lespedeza (*Lepedeza cuneata*) on gastrointestinal nematode infections of growing goats. *Vet Parasitol.* **2014**, *204*, 221–228. [CrossRef]
- Min, B.R.; Pinchak, W.E.; Merkel, R.; Walker, S.; Tomita, G.; Anderson, R.C. Comparative antimicrobial activity of tannin extracts from perennial plants on mastitis pathogens. *Sci. Res. Essay* **2008**, *3*, 66–73.
- Naumann, H.D.; Muir, J.P.; Lambert, B.D.; Tedeschi, L.O.; Kothmann, M.M. Condensed tannins in the ruminant environment: A perspective on biological activity. *J. Agric. Sci.* **2013**, *1*, 8–20.
- Puchala, R.; Anmut, G.; Patra, A.; Detweiler, G.; Wells, J.; Varel, V.; Sahlu, T.; Goetsch, A. Methane emissions by goats consuming Sericea lespedeza at different feeding frequencies. *Anim. Feed Sci. Technol.* **2012**, *175*, 76–84. [CrossRef]
- Batchu, P.; Hazard, T.; Lee, J.H.; Terrill, T.H.; Kouakou, B.; Kannan, G. High-condensed tannin diet and transportation stress in goats: Effects on physiological responses, gut microbial counts and meat quality. *Animals* **2021**, *11*, 2857. [CrossRef]
- Batchu, P.; Terrill, T.H.; Kouakou, B.; Estrada-Reyes, Z.M.; Kannan, G. Plasma metabolomic profiles as affected by diet and stress in Spanish goats. *Sci. Rep.* **2021**, *11*, 12607. [CrossRef]
- Botha, H. The use of sericea lespedeza (Smart Man’s Lucerne) in South Africa. In Proceedings of the What Works with Worms 2015 International Congress on Sustainable Parasite Management, Pretoria, South Africa, 25–26 May 2015.
- Buchanan, G.A.; Burns, E.R. *Weed Control in Sericea Lespedeza*; Circular # 165; Agricultural Experiment Station, Auburn University: Auburn, AL, USA, 1969.
- Wang, C.; Zhou, B.; Palm, H.L. Detecting invasive sericea lespedeza (*Lepedeza cuneata*) in Mid-Missouri pastureland using hyperspectral imagery. *Environ. Manag.* **2008**, *41*, 853–862. [CrossRef]
- Gholizadeh, H.; Friedman, M.S.; McMillan, N.A.; Hammond, W.M.; Hassani, K.; Sams, A.V.; Charles, M.D.; Garrett, D.R.; Joshi, O.; Hamilton, R.G.; et al. Mapping invasive alien species in grassland ecosystems using airborne imaging spectroscopy and remotely observable vegetation functional traits. *Remote Sens. Environ.* **2022**, *271*, 112887. [CrossRef]
- Rakotoarivony, M.N.A.; Gholizadeh, H.; Hammond, W.M.; Hassani, K.; Joshi, O.; Hamilton, R.G.; Fuhlendorf, S.D.; Trowbridge, A.M.; Adams, H.D. Detecting the invasive *Lepedeza cuneata* in grasslands using commercial small satellite imagery. *Int. J. Remote Sens.* **2023**, *44*, 6802–6824. [CrossRef]
- Panda, S.; Terrill, T.H.; Mahapatra, A.; Morgan, E.; Siddique, A.; Pech-Cervantes, A.A.; Van Wyk, J. Geospatial engineering and technology supported climate-sensitive Sericea lespedeza fodder production suitability analysis modeling in the southeastern United States. In Proceedings of the 2023 IST-Africa Conference (IST-Africa), Tshwane, South Africa, 31 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–12.
- Turner, W.; Rondinini, C.; Pettorelli, N.; Mora, B.; Leidner, A.; Szantoi, Z.; Buchanan, G.; Dech, S.; Dwyer, J.; Herold, M.; et al. Free and open-access satellite data are key to biodiversity conservation. *Biol. Conserv.* **2015**, *182*, 173–176. [CrossRef]
- Zhang, Y.; Sun, L.; Yan, C.; Ji, X.; Dai, Q. Adaptive residual networks for high-quality image restoration. *IEEE Trans. Image Process.* **2018**, *27*, 3150–3163. [CrossRef]
- Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
- Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 83. [CrossRef]
- Qin, H.; Li, X.; Liang, J.; Peng, Y.; Zhang, C. DeepFish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing* **2016**, *187*, 49–58. [CrossRef]

22. Nguyen, T.H.; Nguyen, T.N.; Ngo, B.V. A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease. *AgriEngineering* **2022**, *4*, 871–887. [[CrossRef](#)]
23. Panda, S.S.; Terrill, T.H.; Mahapatra, A.K.; Morgan, E.R.; Siddique, A.; Pech-Cervantes, A.A.; Van Wyk, J.A. Optimizing Sericea Lespedeza fodder production in the southeastern US: A climate-informed geospatial engineering approach. *Agriculture* **2023**, *13*, 1661. [[CrossRef](#)]
24. Balas, V.E.; Kumar, R.; Srivastava, R. (Eds.) *Recent Trends and Advances in Artificial Intelligence and Internet of Things*; Springer International Publishing: Cham, Switzerland, 2020.
25. Zhang, W.; Itoh, K.; Tanida, J.; Ichioka, Y. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Appl. Opt.* **1990**, *29*, 4790–4797. [[CrossRef](#)]
26. Piechaud, N.; Hunt, C.; Culverhouse, P.F.; Foster, N.L.; Howell, K.L. Automated identification of benthic epifauna with computer vision. *Mar. Ecol. Prog. Ser.* **2019**, *615*, 15–30. [[CrossRef](#)]
27. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
28. Obla, S.; Gong, X.; Aloufi, A.; Hu, P.; Takabi, D. Effective activation functions for homomorphic evaluation of deep neural networks. *IEEE Access* **2020**, *8*, 153098–153112. [[CrossRef](#)]
29. Wang, H.; Qu, Z.; Zhou, Q.; Zhang, H.; Luo, B.; Xu, W.; Guo, S.; Li, R. A comprehensive survey on training acceleration for large machine learning models in IoT. *IEEE Internet Things J.* **2021**, *9*, 939–963. [[CrossRef](#)]
30. Zhang, C.; Chang, C.C.; Jamshidi, M. Bridge damage detection using a single-stage detector and field inspection images. *arXiv* **2018**, arXiv:1812.10590.
31. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19. Available online: https://openaccess.thecvf.com/content_ECCV_2018/html/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.html (accessed on 3 March 2024).
32. Agarwal, S.; Terrail, J.O.D.; Jurie, F. Recent advances in object detection in the age of deep convolutional neural networks. *arXiv* **2018**, arXiv:1809.03193.
33. Goundar, N.N. Improved Deep Learning Model Based on Integrated Convolutional Neural Networks and Transfer Learning for Shoeprint Image Classification. Master’s Thesis, Auckland University of Technology, Auckland, New Zealand, 2023.
34. Yu, C.; Qi, X.; Ma, H.; He, X.; Wang, C.; Zhao, Y. LLR: Learning learning rates by LSTM for training neural networks. *Neurocomputing* **2020**, *394*, 41–50. [[CrossRef](#)]
35. Torres-Velázquez, M.; Chen, W.J.; Li, X.; McMillan, A.B. Application and construction of deep learning networks in medical imaging. *IEEE Trans. Radiat. Plasma Med. Sci.* **2020**, *5*, 137–159. [[CrossRef](#)]
36. Zhang, Y.; Wang, S.; Sun, G.; Mao, J. Aerodynamic surrogate model based on deep long short-term memory network: An application on high-lift device control. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2022**, *236*, 1081–1097. [[CrossRef](#)]
37. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Hassani, A.E.; Pandey, H.M. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Comput. Electron. Agric.* **2020**, *175*, 105456. [[CrossRef](#)]
38. Yaqub, M.; Feng, J.; Zia, M.S.; Arshid, K.; Jia, K.; Rehman, Z.U.; Mehmood, A. State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sci.* **2020**, *10*, 427. [[CrossRef](#)]
39. Tiwari, V.; Joshi, R.C.; Dutta, M.K. Deep neural network for multi-class classification of medicinal plant leaves. *Expert Syst.* **2022**, *39*, e13041. [[CrossRef](#)]
40. Krohn, J.; Beyleveld, G.; Bassens, A. *Deep Learning Illustrated*; Addison-Wesley Professional: Boston, MA, USA, 2019.
41. Ott, J.; Pritchard, M.; Best, N.; Linstead, E.; Curcic, M.; Baldi, P. A Fortran-Keras deep learning bridge for scientific computing. *Sci. Program.* **2020**, *2020*, 8888811. [[CrossRef](#)]
42. Liu, Y.; Li, D. AdaXod: A new adaptive and momental bound algorithm for training deep neural networks. *J. Supercomput.* **2023**, *79*, 17691–17715. [[CrossRef](#)]
43. Elshamy, R.; Abu-Elnasr, O.; Elhoseny, M.; Elmougy, S. Improving the efficiency of RMSProp optimizer by utilizing Nesterov in deep learning. *Sci. Rep.* **2023**, *13*, 8814. [[CrossRef](#)]
44. Xu, C.; Coen-Pirani, P.; Jiang, X. Empirical Study of Overfitting in Deep Learning for Predicting Breast Cancer Metastasis. *Cancers* **2023**, *15*, 1969. [[CrossRef](#)]
45. Hosseini, M.; Powell, M.; Collins, J.; Callahan-Flintoft, C.; Jones, W.; Bowman, H.; Wyble, B. I tried a bunch of things: The dangers of unexpected overfitting in classification of brain data. *Neurosci. Biobehav. Rev.* **2020**, *119*, 456–467. [[CrossRef](#)]
46. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
47. Algan, G.; Ulusoy, I. Image classification with deep learning in the presence of noisy labels: A survey. *Knowl.-Based Syst.* **2021**, *215*, 106771. [[CrossRef](#)]
48. Smith, S.L.; Kindermans, P.J.; Ying, C.; Le, Q.V. Don’t decay the learning rate, increase the batch size. *arXiv* **2017**, arXiv:1711.00489.
49. Dyrmann, M.; Jørgensen, R.N.; Midtby, H.S. RoboWeedSupport-Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Adv. Anim. Biosci.* **2017**, *8*, 842–847. [[CrossRef](#)]
50. Olsen, A.; Konovalov, D.A.; Philippa, B.; Ridd, P.; Wood, J.C.; Johns, J.; Banks, W.; Girgenti, B.; Kenny, O.; Whinney, J.; et al. DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* **2019**, *9*, 2058. [[CrossRef](#)]

51. Gothai, E.; Natesan, P.; Aishwariya, S.; Aarthy, T.B.; Singh, G.B. Weed identification using convolutional neural network and convolutional neural network architectures. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; pp. 958–965.
52. Yu, J.; Sharpe, S.M.; Schumann, A.W.; Boyd, N.S. Detection of broadleaf weeds growing in turfgrass with convolutional neural networks. *Pest Manag. Sci.* **2019**, *75*, 2211–2218. [[CrossRef](#)]
53. Yu, J.; Sharpe, S.M.; Schumann, A.W.; Boyd, N.S. Deep learning for image-based weed detection in turfgrass. *Eur. J. Agron.* **2019**, *104*, 78–84. [[CrossRef](#)]
54. Yu, J.; Schumann, A.W.; Cao, Z.; Sharpe, S.M.; Boyd, N.S. Weed detection in perennial ryegrass with deep learning convolutional neural network. *Front. Plant Sci.* **2019**, *10*, 1422. [[CrossRef](#)]
55. Boursianis, A.D.; Papadopoulou, M.S.; Diamantoulakis, P.; Liopa-Tsakalidi, A.; Barouchas, P.; Salahas, G.; Karagiannidis, G.; Wan, S.; Goudos, S.K. Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review. *Internet Things* **2022**, *18*, 100187. [[CrossRef](#)]
56. Arsenovic, M.; Karanovic, M.; Sladojevic, S.; Anderla, A.; Stefanovic, D. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry* **2019**, *11*, 939. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.