Weed Science

www.cambridge.org/wsc

Deep learning–based automated weed coverage estimation for herbicide efficacy assessment and turfgrass management

Qiuyu Zu[1], Wenpeng Zhu[2], Jinxu Wang[1], Teng Liu[1], Yiran Li[2], Xiaojun Jin[3] and Jialin Yu[4]

[1]Research Assistant, Peking University Institute of Advanced Agricultural Sciences, Shandong Laboratory of Advanced Agricultural Sciences in Weifang, Shandong, China; [2]Intern, Peking University Institute of Advanced Agricultural Sciences, Shandong Laboratory of Advanced Agricultural Sciences in Weifang, Shandong, China; [3]Associate Professor, National Engineering Research Center of Biomaterials, Nanjing Forestry University, Nanjing, China and [4]Professor and Principal Investigator, Peking University Institute of Advanced Agricultural Sciences, Shandong Laboratory of Advanced Agricultural Sciences in Weifang, Shandong, China

Research Article

Cite this article: Zu Q, Zhu W, Wang J, Liu T, Li Y, Jin X, Yu J (2025). Deep learning–based automated weed coverage estimation for herbicide efficacy assessment and turfgrass management. Weed Sci. **73**(e60), 1–9. doi: 10.1017/wsc.2025.10029

Received: 19 November 2024
Revised: 25 April 2025
Accepted: 31 May 2025

Associate Editor:
Te-Ming Paul Tseng, Mississippi State University

Keywords:
Convolutional neural networks; deep learning; mobile application; weed coverage estimation

Corresponding authors: Xiaojun Jin; Email: xiaojunjin@njfu.edu.cn; Jialin Yu; Email: jialin.yu@pku-iaas.edu.cn

Abstract

In small-plot experiments, weed scientists have traditionally estimated herbicide efficacy through visual assessments or manual counts with wooden frames—methods that are time-consuming, labor-intensive, and error-prone. This study introduces a novel mobile application (app) powered by convolutional neural networks (CNNs) to automate the evaluation of weed coverage in turfgrass. The mobile app automatically segments input images into 10 by 10 grid cells. A comparative analysis of EfficientNet, MobileNetV3, MobileOne, ResNet, ResNeXt, ShuffleNetV1, and ShuffleNetV2 was conducted to identify weed-infested grid cells and calculate weed coverage in bahiagrass (*Paspalum notatum* Flueggé), dormant bermudagrass [*Cynodon dactylon* (L.) Pers.], and perennial ryegrass (*Lolium perenne* L.). Results showed that EfficientNet and MobileOne outperformed other models in detecting weeds growing in bahiagrass, achieving an F_1 score of 0.988. For dormant bermudagrass, ResNet performed best, with an F_1 score of 0.996. Additionally, app-based coverage estimates (11%) were highly consistent with manual assessments (11%), showing no significant difference ($P = 0.3560$). Similarly, ResNeXt achieved the highest F_1 score of 0.996 for detecting weeds growing in perennial ryegrass, with app-based and manual coverage estimates also closely aligned at 10% ($P = 0.1340$). High F_1 scores across all turfgrass types demonstrate the models' ability to accurately replicate manual assessments, which is essential for herbicide efficacy trials requiring precise weed coverage data. Moreover, the time for weed assessment was compared, revealing that manual counting with 10 by 10 wooden frames took an average of 39.25, 37.25, and 42.25 s per instance for bahiagrass, dormant bermudagrass, and perennial ryegrass, respectively, whereas the app-based approach reduced the assessment times to 8.23, 7.75, and 14.96 s, respectively. These results highlight the potential of deep learning–based mobile tools for fast, accurate, scalable weed coverage assessments, enabling efficient herbicide trials and offering labor and cost savings for researchers and turfgrass managers.

Introduction

The turfgrass industry plays a significant role in various sectors, including sports, landscaping, and recreation, contributing to the economy and enhancing environmental aesthetics (Breuninger et al. 2013). However, weed infestations pose a persistent challenge, compromising turfgrass quality (Brosnan et al. 2020; Yu and McCullough 2016). Weeds compete with turfgrass for essential resources such as nutrients, water, and light, leading to reduced vigor and aesthetic appeal (Brosnan et al. 2020). Effective weed management is crucial to maintaining healthy turfgrass. Among available strategies, herbicides have long been a cornerstone of weed management in turfgrass (Hahn et al. 2020; Yu and McCullough 2016). However, the industry faces persistent challenges due to the emergence of herbicide-resistant weed populations (Brosnan et al. 2020) and the lack of new herbicide modes of action (Duke and Dayan 2022). In response, chemical companies are actively developing and evaluating new herbicide products to strengthen weed management practices.

Accurate estimation of weed coverage and density is essential for evaluating herbicide efficacy and developing effective weed management tactics (Liu et al. 2021; Shorewala et al. 2021). By assessing weed density before and after herbicide application, researchers can quantify herbicide effectiveness and facilitate comparisons across treatments (Ozaslan et al. 2024; Patton et al. 2019). Such evaluations are essential for determining the optimal herbicide dosages that maximize control while minimizing environmental impact (Jin et al. 2025). Moreover, weed density data support the development of integrated weed management strategies (Harker and O'Donovan 2013). Understanding weed density and distribution within a field allows agronomists and researchers to tailor management practices, improving overall weed management efficacy (Norsworthy et al. 2018). In addition, in the context of herbicide development, analyzing the interaction between weed density and herbicide effectiveness is crucial (Harker and O'Donovan 2013). Such analyses contribute to designing targeted weed control solutions that address specific weed density pressures, ensuring precise and effective management (Dammer 2016; Jin et al. 2025; Zheng et al. 2025).

In herbicide experiments, the collection of weed control data requires regular monitoring of weed coverage following pesticide application. However, human visual observation often leads to errors, especially at low (e.g., 5%, 10%) and intermediate (e.g., 50%) levels of weed density (Andújar et al. 2010). These limitations hinder visual methods in threshold-based weed management programs (Andújar et al. 2010), which are common in turfgrass weed evaluations (Bertin et al. 2009). Moreover, data collected solely through visual observation are often insufficient for publication in many scientific journals, making it challenging for researchers to share their findings. To address this issue, researchers have used wooden frames with a grid of lines to create multiple squares (Fermanian et al. 1980). In this method, the frame is randomly placed in the field plot, and the number of squares containing weeds is manually counted. The percentage of squares with weeds, relative to the total number of squares, provides an estimate of weed coverage (Fermanian et al. 1980). These frames typically contain a 10 by 10 grid or more. While increasing the number of squares improves estimation accuracy, it also increases the time required for manual counting.

Convolutional neural networks (CNNs) have revolutionized agricultural practices, enhancing precision in crop management (Jin et al. 2024; Tao and Wei 2025; Zhuang et al. 2023). These algorithms are particularly useful in precision weed management, enabling the identification of specific weed species, counting tillers, and pinpointing weed locations for targeted herbicide application (Deng et al. 2025; Jin et al. 2023). Despite advancements, there remains a gap in using CNNs for accurate weed density and coverage estimation, particularly in turfgrass scenarios. Previous research has explored weed localization using CNNs by generating grid cells on input images and using image classification networks to detect weeds within those cells (Jin et al. 2022). This approach, which employs a grid-based segmentation method, has demonstrated high performance (F_1 score ≥ 0.972) (Jin et al. 2022). However, its applicability for estimating turfgrass weed coverage and density has not yet been investigated. We hypothesize that this grid-based segmentation method can be adapted for weed coverage estimation in turfgrass, a concept that, to our knowledge, remains unexplored in the literature.

Developing deep learning models capable of identifying grid cells on input images offers the potential to automate weed coverage and density estimation, significantly reducing time and effort while improving the accuracy of herbicide efficacy assessments. This highlights the need for research to improve both the accuracy and efficiency of herbicide evaluations. Therefore, the objectives of this research were (1) to develop a mobile application (app) that automates the process of weed detection and density estimation using deep learning and (2) to compare the performance of various deep learning models in estimating weed coverage in turfgrass.

Materials and Methods

Overview

Seven CNN architectures—ResNet (He et al. 2016), ResNeXt (Xie et al. 2017), ShuffleNetV1 (Zhang et al. 2018), ShuffleNetV2 (Ma et al. 2018), EfficientNet (Tan and Le 2019), MobileNetV3 (Howard et al. 2019), and MobileOne (Vasu et al. 2023)—were investigated to detect weeds in bahiagrass (*Paspalum notatum* Flueggé), dormant bermudagrass [*Cynodon dactylon* (L.) Pers.], and perennial ryegrass (*Lolium perenne* L.).

ResNet is a groundbreaking CNN architecture that addresses the issue of vanishing gradients in very deep networks (He et al. 2016). It introduces residual learning by using shortcut connections, allowing for the training of extremely deep networks with improved accuracy. ResNet's architecture significantly advanced the field of deep learning by enabling the successful training of networks with hundreds or even thousands of layers (He et al. 2016).

ResNeXt builds upon the ResNet architecture, aiming to improve accuracy and efficiency in deep learning tasks (Xie et al. 2017). It employs a strategy called "cardinality," grouping multiple parallel paths (similar to ResNet blocks) to increase model capacity without significantly raising computational complexity. ResNeXt demonstrates that improving network performance is more effectively accomplished by widening the network using grouped convolutions, as opposed to merely increasing its depth (Xie et al. 2017).

ShuffleNetV1 is a lightweight CNN architecture designed for efficient image classification (Zhang et al. 2018). It uses group convolution and channel shuffling to reduce computational cost while maintaining accuracy. Its low complexity and resource requirements make it particularly suitable for mobile and embedded devices.

ShuffleNetV2 further optimizes the original ShuffleNet architecture to enhance both efficiency and accuracy (Ma et al. 2018). It introduces the concept of equal channel width and simplified network design, addressing the limitations of the first version, particularly in terms of memory access cost. The architecture also features a more effective channel split and a new feature map shuffle operation, enhancing both speed and performance (Ma et al. 2018).

EfficientNet employs a state-of-the-art CNN architecture with a compound scaling method to uniformly scale network depth, width, and resolution (Tan and Le 2019). This approach creates a family of models that achieve superior accuracy with fewer parameters and lower computational resources than previous architectures, making it ideal for tasks requiring high efficiency, particularly in resource-constrained environments.

MobileNetV3 optimizes the earlier MobileNet architecture, focusing on enhancing both efficiency and accuracy for mobile and edge devices (Howard et al. 2019). It combines neural architecture search with platform-aware network optimization, resulting in a model that balances high performance with low computational costs.

MobileOne is a streamlined CNN architecture designed for real-time apps on mobile and edge devices (Vasu et al. 2023). It uses a pre-parameterization technique to enhance both speed and accuracy during inference, achieving high efficiency with minimal computational requirements.

Image Acquisition

Images of Florida pusley (*Richardia scabra* L.) in actively growing bahiagrass were captured over several time periods from May to

August 2018. The training images were taken at the Gulf Coast Research and Education Center in Balm, FL, USA (27.71°N, 82.29°W), while the test images were collected from multiple commercial and residential lawns in Riverview, FL, USA (27.81°N, 82.42°W), and the University of South Florida campus in Tampa, FL, USA (27.95°N, 82.45°W).

For dormant bermudagrass, training images were captured at the University of Georgia Griffin Campus in Griffin, GA, USA (33.26°N, 84.28°W), while the test images were primarily collected from various golf courses in Peachtree City, GA, USA (33.39°N, 84.59°W). The training and test images predominantly contained annual bluegrass (*Poa annua* L.) and various winter annual broadleaf weeds.

For perennial ryegrass, images of common dandelion (*Taraxacum officinale* F.H. Wigg.), ground ivy (*Glechoma hederacea* L.), and spotted spurge [*Chamaesyce maculata* (L). Small; syn.: *Euphorbia maculata* L.) growing within perennial ryegrass were collected from various golf courses and institutional lawns in Indianapolis, IN, USA (39.76°N, 86.15°W) to construct the training datasets. The testing dataset consisted of images of the same weed species collected from multiple institutional lawns and golf courses in Carmel, IN, USA (39.97°N, 86.11°W).

All training and testing images were acquired using a Sony Cyber-shot camera (Sony, Minato-ku, Tokyo, Japan) with a resolution of 1,920 × 1,080 pixels and were captured between 9:00 AM and 5:00 PM under diverse weather and outdoor lighting conditions, including clear, cloudy, and partly cloudy skies.

Training and Testing

During image classification training, images of bahiagrass, dormant bermudagrass, and perennial ryegrass were cropped into subimages with a resolution of 426 × 240 pixels using Irfanview (v. 5.50, Irfan Skiljan, Jajce, Bosnia). The training dataset included 20,000 positive images (containing weeds) and 20,000 negative images (without weeds). For validation, an additional set of 5,000 positive images and 5,000 negative images was utilized.

The image classification neural networks were trained and evaluated using the PyTorch (v. 1.12.0) open-source deep learning framework developed by Facebook (San Jose, CA, USA). All computations were performed on an NVIDIA GeForce RTX 3080 graphics processing unit (Santa Clara, CA, USA). To ensure consistency across models, the following hyperparameters were standardized for each experimental setup:

- Training epochs: 30
- Learning rate: 0.001
- Batch size: 32

In image classification, precision (Equation 1), recall (Equation 2), and F_1 score (Equation 3) are widely used metrics for evaluating model performance. These metrics are derived from the confusion matrix, which compares a model's predicted classifications with the actual labels. The confusion matrix categorizes predictions into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), specifically:

- TP: Number of instances correctly classified as positive.
- FN: Number of instances incorrectly classified as negative when actually positive.
- FP: Number of instances incorrectly classified as positive when actually negative.
- TN: Number of instances correctly classified as negative.

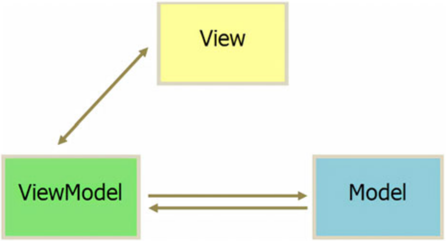


Figure 1. The Model-View-ViewModel (MVVM) architectural diagram.

Precision is the ratio of instances correctly predicted as positive to the total number of instances predicted as positive, focusing on minimizing FPs to ensure accurate positive predictions (Sokolova and Lapalme 2009).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

Recall is the proportion of TPs correctly classified by the model. A high recall indicates the model's effectiveness in classifying TPs, highlighting its ability to minimize FNs (Sokolova and Lapalme 2009).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

The F_1 score is a comprehensive metric that combines precision and recall into a single value. It is calculated as the harmonic mean of precision and recall, offering a balanced assessment of the model's performance, especially in scenarios where maintaining high precision and recall is crucial (Sokolova and Lapalme 2009).

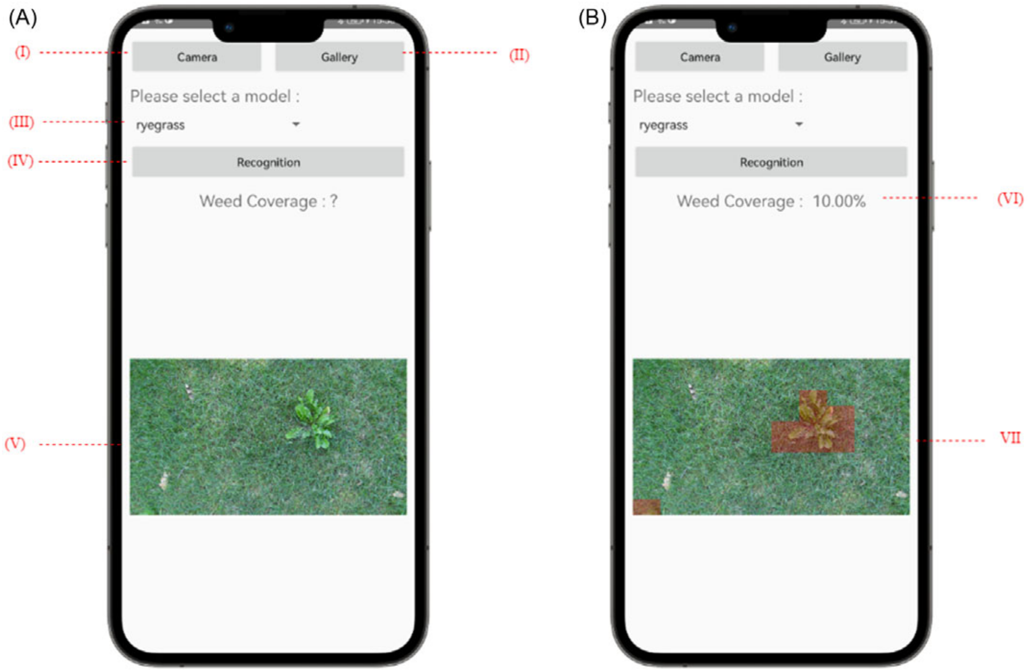
$$\text{F}_1\text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The rationale for selecting these metrics lies in their critical role in evaluating weed coverage and density using a classification neural network. Precision measures the proportion of grids predicted as containing weeds that are actually weed infested, helping to reduce FPs that could lead to overestimation of weed density. Recall quantifies the proportion of actual weed-infested grids that are correctly identified, ensuring that the model effectively captures weed presence. Because both FPs and FNs impact estimation, the F_1 score, which balances precision and recall, provides a meaningful overall assessment of the model's classification performance.

Development of the Mobile App

The mobile app for this study was developed using Android Studio, with Kotlin as the programming language. Kotlin offers several advantages, such as code conciseness and reduced NullPointerExceptions (Ardito et al. 2020). The Android Software Development Kit version employed was Application Programming Interface (API) level 33. The app was designed following the Model-View-ViewModel (MVVM) architectural pattern, utilizing Kotlin co-routines to create a robust and efficient app (Chauhan et al. 2021). In Android development, MVVM provides a clear separation of concerns among components, promoting clean code and enhancing testability (Lou 2016). Figure 1 illustrates the MVVM pattern's three fundamental components: the Model, the View, and the ViewModel.

https://doi.org/10.1017/wsc.2025.10029 Published online by Cambridge University Press



Please select a model : ryegrass

Weed Coverage : ?

Please select a model : ryegrass

Weed Coverage : 10.00%

Figure 2. User interface of the app. (A) Configuration page showing (I) button to select an image from the camera; (II) button to select an image from the gallery; (III) dropdown to select a model; (IV) button to start recognition; and (V) image view displaying the selected image. (B) Result page showing (VI) text view displaying weed coverage; (VII) image view showing the recognized image.

- Model: The Model component represents the data and core business logic of the app. It interacts with underlying data sources, such as databases or network APIs, to retrieve and manage data efficiently.
- View: The View component is responsible for rendering and handling the user interface (UI). It displays data provided by the ViewModel and triggers actions based on user input.
- ViewModel: The ViewModel serves as an intermediary between the Model and the View. It holds the data to be displayed in the UI and provides methods and properties that can be bound directly to UI elements. Designed to be independent of the specific UI framework, the ViewModel enhances testability and allows for better decoupling from the View layer.

To deploy the model on Android, this study utilized the Open Neural Network Exchange (ONNX) platform. Research has demonstrated that converting machine learning models to the ONNX format and deploying them with ONNX Runtime significantly improves inference speed while maintaining accuracy, making it particularly beneficial for real-time apps on mobile devices (Openja et al. 2022). ONNX serves as an open standard for representing machine learning models, enabling AI developers to use models across multiple frameworks, tools, runtimes, and compilers (Lin et al. 2019). It plays a crucial role in promoting interoperability and standardization in machine learning, allowing for more efficient model development, deployment, and optimization workflows (Kim et al. 2021).

Figure 2 illustrates the app’s UI. The process begins with the user importing an image from the camera or gallery and selecting a suitable recognition model based on the image content (Figure 2A). Once the model is prepared, the user clicks the recognition button to initiate weed coverage analysis with the chosen model, and the results are then displayed (Figure 2B).

To enhance segmentation accuracy and performance, a 10 by 10 grid was selected, as it strikes a balance between segmentation precision and computational efficiency, while also aligning with the actual wooden grid configurations commonly used by weed scientists. Given the hardware limitations inherent to mobile devices, such as limited memory and processing power, careful consideration was necessary to execute this segmentation efficiently. This constraint influenced the choice of in-memory segmentation, ensuring the process remains feasible within typical mobile device capabilities. For accurate recognition, images were preprocessed by segmenting them into 100 smaller sections using the 10 by 10 grid. Due to the time-intensive nature of this process, in-memory segmentation was used. In the Android system, the screen follows a two-dimensional coordinate system, with the origin at the top left corner, the positive x axis extending to the right, and the positive y axis extending downward (Figure 3). The blue area represents the selected image, while the red area indicates a subimage. Each subimage’s dimensions were determined by dividing the width and height of the selected image by 10. Starting from the top left corner, subimages were processed sequentially from left to right across each row. Once a row was completed, the process shifted to the next row, continuing until the bottom-right subimage was reached. Each subimage was stored in a list for subsequent recognition operations. The detailed segmentation process is illustrated in Figure 4.

Detailed Process Description

1. Start: The function begins, ready to process the source image.
2. Retrieve source image dimensions: The width and height of the source image are obtained for subsequent calculations.
3. Calculate segment dimensions: The width and height of each segment are calculated based on a 10 by 10 grid structure.

https://doi.org/10.1017/wsc.2025.10029 Published online by Cambridge University Press

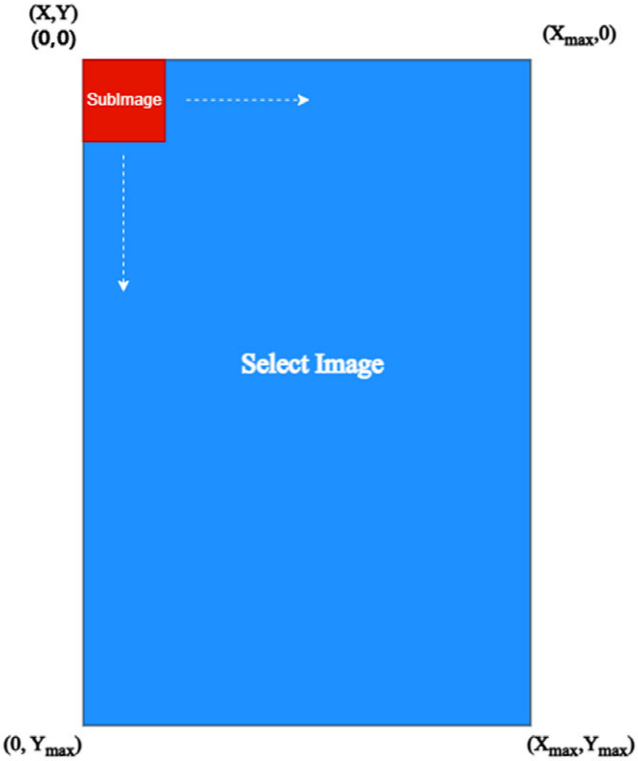


Figure 3. Grid-based coordinate system for image segmentation on Android. This figure shows how the input image is divided into a 10 by 10 grid for subimage processing. The origin (0, 0) starts at the top left, and subimages (highlighted in red) are extracted in row-major order from the original image (in blue).

4. Initialize SubImage list: An empty list is created to store the subimages.
5. Outer loop (columns): Iterates over columns, controlled by columnIndex (0 to columnTotal -1).
6. Inner loop (rows): Iterates over rows, controlled by rowIndex (0 to rowTotal -1).
7. Calculate starting coordinates: For each segment, the starting coordinates (subImageXValue and subImageYValue) are calculated based on rowIndex and columnIndex.
8. Create SubBitmap: A sub-bitmap is created from the source image at the calculated starting coordinates (subImageXValue, subImageYValue) with the determined width and height using Bitmap.createBitmap.
9. Add subbitmap to list: The new sub-bitmap is added to subImageList.
10. Inner loop condition check: If rowIndex < rowTotal -1, the inner loop continues.
11. Outer loop condition check: If columnIndex < column Total -1, the outer loop continues.
12. Return subimage list: After all loops complete, the function returns subImageList containing all subimages.
13. End: The function execution concludes.

Given the requirement to process 100 segmented images, the initial sequential recognition method was found to be inefficient. Dynamic batch processing using ONNX was introduced to improve efficiency and overall performance. Dynamic batching adjusts the batch size during inference based on real-time input data, enabling more flexible utilization of computational resources.

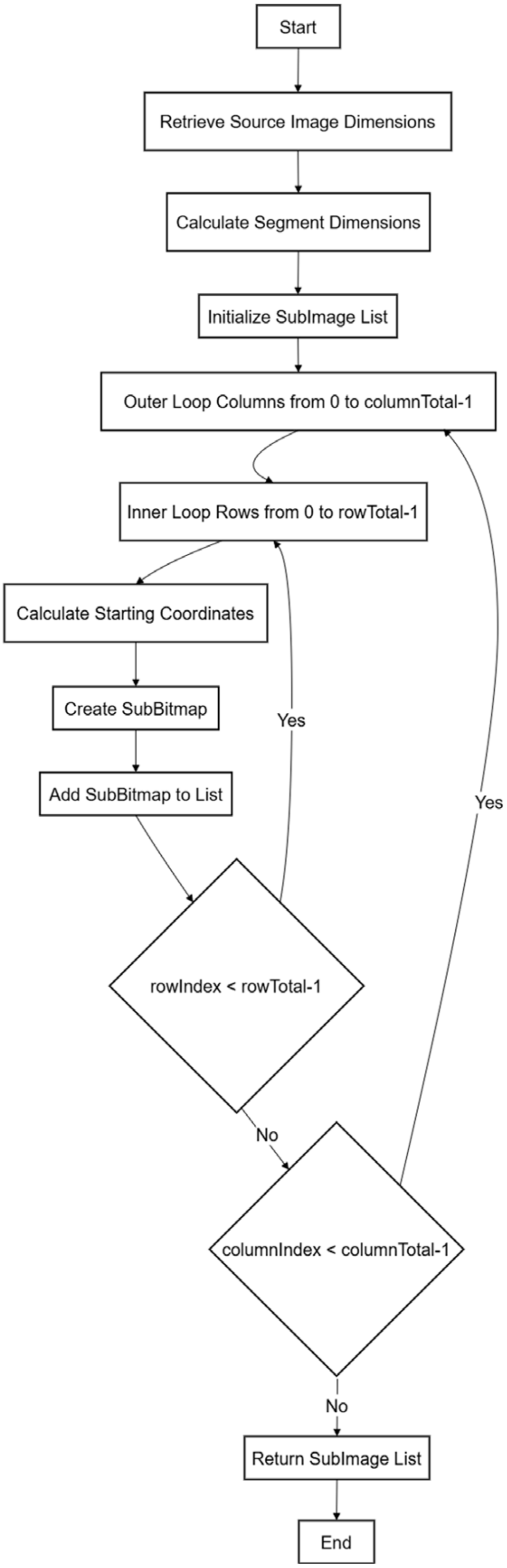

Start
Retrieve Source Image Dimensions
Calculate Segment Dimensions
Initialize SubImage List
Outer Loop Columns from 0 to columnTotal-1
Inner Loop Rows from 0 to rowTotal-1
Calculate Starting Coordinates
Create SubBitmap
Add SubBitmap to List
rowIndex < rowTotal-1
Yes
No
columnIndex < columnTotal-1
Yes
No
Return SubImage List
End

Figure 4. Workflow of mobile-based image segmentation. This figure outlines the process of segmenting the input image into a grid of subimages. After subimage dimensions are computed, a nested loop iterates through each cell to extract and store subimages for further classification.

Table 1. Neural network validation and testing result for detection of weeds in turfgrass.

		Validation[c]			Testing[d]		
Neural network[a]	Turfgrass species[b]	Precision	Recall	F_1 score	Precision	Recall	F_1 score
ResNet	Bahiagrass	0.984	0.984	0.984	0.957	0.942	0.949
ResNet	Dormant bermudagrass	0.986	0.985	0.985	0.996	0.997	0.996
ResNet	Perennial ryegrass	0.983	0.983	0.983	0.993	0.993	0.993
ResNeXt	Bahiagrass	0.986	0.986	0.986	0.962	0.924	0.939
ResNeXt	Dormant bermudagrass	0.987	0.986	0.986	0.989	0.989	0.989
ResNeXt	Perennial ryegrass	0.984	0.985	0.984	0.996	0.996	0.996
ShuffleNetV1	Bahiagrass	0.971	0.968	0.969	0.925	0.893	0.906
ShuffleNetV1	Dormant bermudagrass	0.986	0.984	0.985	0.993	0.992	0.993
ShuffleNetV1	Perennial ryegrass	0.976	0.977	0.977	0.991	0.991	0.991
ShuffleNetV2	Bahiagrass	0.987	0.987	0.987	0.953	0.926	0.938
ShuffleNetV2	Dormant bermudagrass	0.987	0.985	0.986	0.982	0.982	0.982
ShuffleNetV2	Perennial ryegrass	0.984	0.985	0.985	0.994	0.994	0.994
EfficientNet	Bahiagrass	0.990	0.989	0.989	0.987	0.989	0.988
EfficientNet	Dormant bermudagrass	0.986	0.984	0.985	0.989	0.989	0.989
EfficientNet	Perennial ryegrass	0.991	0.991	0.991	0.992	0.992	0.992
MobileNetV3	Bahiagrass	0.988	0.987	0.988	0.985	0.983	0.984
MobileNetV3	Dormant bermudagrass	0.979	0.982	0.980	0.990	0.988	0.989
MobileNetV3	Perennial ryegrass	0.987	0.987	0.987	0.992	0.992	0.992
MobileOne	Bahiagrass	0.989	0.988	0.989	0.988	0.988	0.988
MobileOne	Dormant bermudagrass	0.976	0.969	0.972	0.993	0.992	0.993
MobileOne	Perennial ryegrass	0.952	0.951	0.950	0.983	0.982	0.982

[a] Neural network: The specific neural network model used in the weed detection task.
[b] Turfgrass species: The turfgrass species used in the weed detection model.
[c] Validation: Precision, recall, and F_1 score results on the validation dataset, used for model tuning.
[d] Testing: Precision, recall, and F_1 score results on the independent testing dataset, used to evaluate model generalization.

This approach significantly improves system efficiency and response time, particularly when handling asynchronous or real-time requests.

App-Based Counting versus Manual Counting

In this study, seven models were trained for each turfgrass: bahiagrass, dormant bermudagrass, and perennial ryegrass. The optimal model for each turfgrass was selected through a comprehensive comparison of precision, recall, and F_1 score from both the validation and test datasets. The best model for each turfgrass species was integrated into the app for weed density estimation. Weed density was assessed manually and using the app, with each method repeated four times per image. A Student's *t*-test was conducted to compare the accuracy of the app-based method with manual counting, using a significance level of 0.05 to determine whether statistically significant differences existed between the two approaches.

Results and Discussion

Model Performance

For bahiagrass detection, EfficientNet and MobileOne demonstrated superior performance with identical F_1 scores (0.988), outperforming other architectures by notable margins. Although they maintained strong detection capabilities, ResNet and ResNeXt showed relatively lower generalization capabilities in this category (Table 1).

In the case of dormant bermudagrass, ResNet achieved the highest F_1 score (0.996), with ShuffleNetV1 and MobileOne forming a close second tier. The performance hierarchy revealed consistent patterns of difference between validation and testing phases across architectures.

Lolium perenne detection results exhibited tight clustering among top performers, with ResNeXt leading with an F_1 score of 0.996 on the testing dataset, followed by ShuffleNetV2 and ResNet. Notably, MobileOne showed a relatively reduced level of effectiveness in this category compared with its strong performance on other species.

App-based versus Manual Weed Coverage Estimation

The app integrates the top-performing models for each turfgrass species to estimate weed coverage and density: EfficientNet, ResNet, and ResNeXt for bahiagrass, dormant bermudagrass, and perennial ryegrass, respectively. For each turfgrass species, four individuals provided manual weed density estimates, and the app generated four corresponding estimates for comparison. As shown in Table 2, the app-based method produced a significantly higher average weed coverage estimate for bahiagrass compared with manual counting ($P < 0.0001$; 95% confidence interval (CI): [10.7600, 11.7400]; Cohen's $d = 31.8198$). In contrast, the app-based method provided estimates comparable to manual estimates for dormant bermudagrass ($P = 0.3560$; 95% CI: [−0.7400, 0.2400]; Cohen's $d = -0.7071$) and perennial ryegrass ($P = 0.1340$; 95% CI: [−1.0658, 0.0658]; Cohen's d = −1.2247), with no significant differences observed between the two methods.

Time Efficiency

The comparative analysis revealed substantially improved time efficiency using the app-based method compared with manual counting across all turfgrass species (Table 3). For bahiagrass, the app-based approach reduced processing time by approximately 79% relative to manual counting. Similar efficiency gains were observed for dormant bermudagrass (79% reduction) and *L. perenne* (65% reduction), with the app consistently completing tasks in under 15 s compared with manual counting, which

Table 2. Comparison of weed coverage percentage estimated by app-based and manual counting methods.

Methods	Bahiagrass	Dormant bermudagrass	Perennial ryegrass
	%		
App-based method[a]	15	11	10
Manual counting[b]	4	11	10
P-value (*t*-test)[c]	<0.0001	0.3560	0.1340
Specification[d]	Actively growing bahiagrass images were taken in Tampa, FL, USA; weed species included only *Richardia scabra*.	Dormant bermudagrass images were taken in Peachtree City, GA, USA; weed species included only *Poa annua*.	Actively growing perennial ryegrass images were taken in Carmel, IN, USA; weed species included *Taraxacum officinale*, *Glechoma hederacea*, and *Chamaesyce maculata*.

[a] App-based method: Percentage of weed coverage estimated using an automated app.
[b] Manual counting: Percentage of weed coverage estimated through manual counting.
[c] P-value (*t*-test): The statistical significance of the difference between the app-based and manual methods for estimating weed coverage. A P-value less than 0.05 indicates a significant difference.
[d] Specification: Details the specific turfgrass images used along with the associated weed species.

Table 3. Descriptive statistics of model and manual recognition time efficiency.

Methods	Bahiagrass	Dormant bermudagrass	Perennial ryegrass
	s		
App-based method[a]	8.23	7.75	14.96
Manual counting[b]	39.25	37.25	42.25
Specification[c]	Actively growing bahiagrass images were taken in Tampa, FL, USA; weed species included only *Richardia scabra*.	Dormant bermudagrass images were taken in Peachtree City, GA, USA; weed species included only *Poa annua*.	Actively growing perennial ryegrass images were taken in Carmel, IN, USA; weed species included *Taraxacum officinale*, *Glechoma hederacea*, and *Chamaesyce maculata*.

[a] App-based method: Time (in seconds) taken by the app to recognize weeds.
[b] Manual counting: Time (in seconds) taken for manual counting of weeds.
[c] Specification: Details the specific turfgrass images used along with the corresponding weed species.

required more than 37 seconds. These quantitative improvements demonstrate the app's ability to accelerate weed coverage assessment while maintaining accuracy across diverse growth conditions and geographic locations, as specified in Table 3.

The application of deep learning models such as ResNet, ResNeXt, and EfficientNet for detecting weeds in bahiagrass, dormant bermudagrass, and perennial ryegrass has shown promising results that can benefit weed management practices (He et al. 2016; Tan and Le 2019; Xie et al. 2017). These models effectively balance the need for high accuracy with computational efficiency, offering a tool that can save time and resources in large-scale agricultural operations. EfficientNet's compound scaling method enabled it to excel in detecting weeds in bahiagrass, capturing complex features with high precision and recall. The residual connections in ResNet and ResNeXt enhanced their capacity to learn deep patterns, contributing to their success with dormant bermudagrass and perennial ryegrass (Ebski et al. 2018; Zhou et al. 2021). In contrast, lightweight models, ShuffleNetV1 and MobileOne, although computationally efficient, lacked the depth needed for detailed feature extraction, resulting in lower accuracy and F_1 scores, especially in the testing dataset (Vasu et al. 2023; Zhang et al. 2018).

The app tended to overestimate weed coverage in bahiagrass, particularly when *R. scabra* was present. To effectively distinguish *R. scabra* growing in bahiagrass, augmenting the training dataset by incorporating more images of *R. scabra* in bahiagrass is essential. In contrast, the app performed well in determining weed coverage in dormant bermudagrass and actively growing perennial ryegrass, showing strong agreement with manual counting. However, further improvements are needed to enhance its accuracy in more complex, actively growing environments.

This study demonstrates that integrating deep learning models into a mobile app for weed coverage and density estimation provides a significant time-saving advantage over traditional manual methods. This efficiency not only streamlines the estimation process but also reduces labor costs, making it particularly well suited for large-scale agricultural experiments. By automating weed coverage assessments, the app-based method offers a faster and more accurate alternative, potentially facilitating herbicide efficacy experiments and improving overall field management practices.

Beyond its technical merits, the app offers substantial economic value for turfgrass practitioners and agricultural researchers. By automating weed coverage assessments, it mitigates human error, enhances consistency, and accelerates data collection. These capabilities facilitate more timely and informed decisions regarding herbicide applications and other weed management strategies. Ultimately, the app enhances both the precision and sustainability of weed control efforts, serving as a practical and cost-effective tool for improving field management and supporting high-throughput herbicide efficacy evaluations.

The app-based method demonstrated promise in estimating weed coverage but exhibited several limitations that warrant improvement. A primary issue was the misclassification of bahiagrass as a weed, particularly in areas where *R. scabra* was present. This misidentification significantly contributed to an overestimation of weed coverage during the active growing phase of bahiagrass and *R. scabra*, underscoring the app's difficulty in distinguishing morphologically similar species. An insufficient amount of *R. scabra* training data may have led to an overestimation of bahiagrass weed coverage, particularly during active growth stages. This finding suggests that further refinement of the app's deep learning models, particularly for actively growing turfgrass species mixed with morphologically similar weeds, is necessary to improve accuracy. In contrast, the app performed well in detecting weeds in dormant bermudagrass and perennial ryegrass, closely aligning with manual counting and indicating its effectiveness in scenarios where turfgrass and weeds are more morphologically distinct.

Due to the limitations of mobile device processors, achieving high-performance weed recognition remains challenging, highlighting the need for a more efficient and accurate algorithm. While our model is currently deployed locally on mobile devices, future deployments could leverage cloud computing. By utilizing the powerful computational capabilities of the cloud, we can overcome the limitations of mobile hardware. Cloud-based deployment would allow for batch processing, significantly improving recognition efficiency. Additionally, cloud-based models could be updated seamlessly, providing greater flexibility and ensuring users have access to the latest advancements in weed detection.

The app's effectiveness in estimating weed coverage may vary across geographic regions and management practices, influenced by factors such as weed species composition and turfgrass dormancy levels. Expanding the training dataset to include a broader range of weed and turfgrass species could enhance the app's generalizability and performance in diverse contexts. Factors such as weed species composition and turfgrass surface quality—shaped by diverse management practices—may influence the accuracy of weed coverage and density estimation. These variations present challenges to the model's generalization, potentially limiting its performance in unfamiliar environments. The training dataset used to develop the model is currently limited in geographic range and weed diversity, raising concerns about its generalizability to different regions or weed species. To address these limitations, users are encouraged to manually verify the app's weed identification accuracy under their specific local conditions. Future efforts will focus on expanding the training image database to include a broader range of turfgrass types, weed species, growth stages, weed densities, and biotypes. This expansion aims to enhance the model's adaptability and ensure consistent performance across diverse regions, weeds, and management practices. To address the misclassification of bahiagrass and *R. scabra*, several strategies can be employed. First, expanding the dataset to include more diverse images of *R. scabra*, particularly in mixed environments with bahiagrass, as well as across varying growth stages, would improve the model's ability to differentiate between the two species. Additionally, incorporating spectral or texture-based features could improve species differentiation by capturing unique characteristics. Finally, implementing a multi-class classification layer to explicitly distinguish turfgrass from weeds would help reduce overestimation and enhance detection accuracy. These approaches are crucial for improving overall classification performance.

This study evaluated various deep learning models for detecting weeds in actively growing bahiagrass, dormant bermudagrass, and actively growing perennial ryegrass and selected the best-performing model for integration into a mobile app. Subsequently, the app-based method was compared with traditional manual counting. For bahiagrass, the app-based method tended to overestimate weed coverage, likely due to challenges in distinguishing bahiagrass from morphologically similar weeds, such as *R. scabra*, during active growth stages. This highlights the need for improved species differentiation. This gap in detection accuracy reflects a broader challenge in the field of weed management, where distinguishing between species with similar morphology is crucial for precision agriculture and effective herbicide application. Encouragingly, the app demonstrated strong alignment with manual counting for dormant bermudagrass and perennial ryegrass. These results suggest that the app is effective for weed detection in cases where turfgrass is dormant or when weed species are morphologically distinct from turfgrasses. While the app performs well for dormant bermudagrass and perennial ryegrass, further refinement is needed for more complex cases like bahiagrass during active growth. Addressing these challenges will be a crucial step toward improving the utility of mobile app–based weed detection in diverse agricultural settings. Future research should expand to cover a broader range of turfgrass species and other crops, such as wheat (*Triticum aestivum* L.). Additionally, collaborations with agricultural researchers and field practitioners would facilitate the development of region-specific models and datasets, improving the app's adaptability to different environments and weed species. Such expansion would allow users to rapidly assess weed coverage and density through simple image capture, supporting enhanced weed management and herbicide efficacy evaluation.

Funding statement. This work was supported by the National Natural Science Foundation of China (grant no. 32072498), the Weifang Science and Technology Development Plan Project (grant no. 2024ZJ1097), the Key R&D Program of Shandong Province, China (grant no. 202211070163), the Taishan Scholar Program of Shandong Province, and the Yuandu Scholar Program of Weifang, Shandong China.

Competing interests. The authors declare no conflicts of interest.

References

Andújar D, Ribeiro A, Carmona R, Fernández-Quintanilla C, Dorado J (2010) An assessment of the accuracy and consistency of human perception of weed cover. Weed Res 50:638–647

Ardito L, Coppola R, Malnati G, Torchiano M (2020) Effectiveness of Kotlin vs. Java in Android app development tasks. Inf Software Technol 127:106374

Bertin C, Senesac AF, Rossi FS, DiTommaso A, Weston LA (2009) Evaluation of selected fine-leaf fescue cultivars for their turfgrass quality and weed suppressive ability in field settings. HortTechnology 19:660–668

Breuninger JM, Welterlen MS, Augustin BJ, Cline V, Morris K (2013) The turfgrass industry. Pages 37–103 *in* Stier JC, Horgan BP, Bonos SA, eds. Turfgrass: Biology, Use, and Management. Wiley Online Library

Brosnan JT, Elmore MT, Bagavathiannan M (2020) Herbicide-resistant weeds in turfgrass: current status and emerging threats. Weed Technol 34:424–430

Chauhan K, Kumar S, Sethia D, Alam MN (2021) Performance analysis of Kotlin coroutines on Android in a model-view-intent architecture pattern. Pages 1–6 *in* 2021 2nd International Conference for Emerging Technology (INCET). Belagavi, India: IEEE

Dammer KH (2016) Real-time variable-rate herbicide application for weed control in carrots. Weed Res 56:237–246

Deng L, Miao Z, Zhao X, Yang S, Gao Y, Zhai C, Zhao C (2025) HAD-YOLO: an accurate and effective weed detection model based on improved YOLOv5 network. Agronomy 15:57

Duke SO, Dayan FE (2022) The search for new herbicide mechanisms of action: is there a "holy grail"? Pest Manag Sci 78:1303–1313

https://doi.org/10.1017/wsc.2025.10029 Published online by Cambridge University Press

Ebski SJ, Arpit D, Ballas N, Verma V, Che T, Bengio Y (2018) Residual connections encourage iterative inference. Pages 1–14 *in* International Conference on Learning Representations. Vancouver, Canada: ICLR.

Fermanian TW, Huffine WW, Morrison RD (1980) Preemergence weed control in seeded bermudagrass stands. Agron J 72:803–805

Hahn D, Sallenave R, Pornaro C, Leinauer B (2020) Managing cool-season turfgrass without herbicides: optimizing maintenance practices to control weeds. Crop Sci 60:2204–2220

Harker KN, O'Donovan JT (2013) Recent weed control, weed management, and integrated weed management. Weed Technol 27:1–11

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Pages 770–778 *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV: IEEE Computer Society

Howard A, Sandler M, Chu G, Chen L, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V (2019) Searching for mobilenetv3. Pages 1314–1324 *in* Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, South Korea: IEEE/CVF

Jin X, Bagavathiannan M, McCullough PE, Chen Y, Yu J (2022) A deep learning-based method for classification, detection, and localization of weeds in turfgrass. Pest Manag Sci 78:4809–4821

Jin X, Han K, Zhao H, Wang Y, Chen Y, Yu J (2024) Detection and coverage estimation of purple nutsedge in turf with image classification neural networks. Pest Manag Sci 80:3504–3515

Jin X, Liu T, McCullough PE, Chen Y, Yu J (2023) Evaluation of convolutional neural networks for herbicide susceptibility-based weed detection in turf. Front Plant Sci 14:1096802

Jin X, Zhao H, Kong X, Han K, Lei J, Zu Q, Chen Y, Yu J (2025) Deep learning-based weed detection for precision herbicide application in turf. Pest Manag Sci 81:3597–3609

Kim S, Han B, Heo J (2021) Model transformation and inference of machine learning using open neural network format. J Inst Internet Broadcast Commun 21:107–114

Lin W, Tsai D, Tang L, Hsieh C, Chou C, Chang P, Hsu L (2019) Onnc: a compilation framework connecting onnx to proprietary deep learning accelerators. Pages 214–218 *in* 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS). Hsinchu, Taiwan: IEEE

Liu J, Abbas I, Noor RS (2021) Development of deep learning-based variable rate agrochemical spraying system for targeted weeds control in strawberry crop. Agronomy 11:1480

Lou T (2016) A Comparison of Android Native App Architecture MVC, MVP and MVVM. Master's thesis. Espoo, Finland: Aalto University. 49 p

Ma N, Zhang X, Zheng H, Sun J (2018) Shufflenet v2: practical guidelines for efficient cnn architecture design. Pages 116–131 *in* Proceedings of the European Conference on Computer Vision (ECCV). Munich, Germany: Springer

Norsworthy JK, Korres NE, Bagavathiannan M (2018) Weed seedbank management: revisiting how herbicides are evaluated. Weed Sci 66:415–417

Openja M, Nikanjam A, Yahmed AH, Khomh F, Jiang ZMJ (2022) An empirical study of challenges in converting deep learning models. Pages 13–23 *in* 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). Limassol, Cyprus: IEEE

Ozaslan C, Gürsoy S, DiTommaso A (2024) Band herbicide application combined with inter-row cultivation as a sustainable weed management strategy for reducing herbicide use: a meta-analysis. Crop Prot 175: 106474

Patton AJ, Braun RC, Weisenberger DVE (2019) Single applications of natural postemergence weed control options do not provide effective ground ivy control. Crop Forage Turfgrass Manage 5:1–7

Shorewala S, Ashfaque A, Sidharth R, Verma U (2021) Weed density and distribution estimation for precision agriculture using semi-supervised learning. IEEE Access 9:27971–27986

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45:427–437

Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. Pages 6105–6114 *in* Proceedings of the 36th International Conference on Machine Learning. Long Beach, CA: PMLR

Tao T, Wei X (2025) STBNA-YOLOv5: an improved YOLOv5 network for weed detection in rapeseed field. Agriculture 15:22

Vasu PKA, Gabriel J, Zhu J, Tuzel O, Ranjan A (2023) Mobileone: an improved one millisecond mobile backbone. Pages 7907–7917 *in* 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, Canada: IEEE/CVF

Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. Pages 1492–1500 *in* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE

Yu J, McCullough PE (2016) Efficacy and fate of atrazine and simazine in doveweed (*Murdannia nudiflora*). Weed Sci 64:379–388

Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. Pages 6848–6856 *in* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT: IEEE

Zheng S, Zhao X, Fu H, Tan H, Zhai C, Chen L (2025) Design and experimental evaluation of a smart intra-row weed control system for open-field cabbage. Agronomy 15:112

Zhou T, Zhao Y, Wu J (2021) Resnext and res2net structures for speaker verification. Pages 301–307 *in* 2021 IEEE Spoken Language Technology Workshop (SLT). Shenzhen, China: IEEE

Zhuang J, Jin X, Chen Y, Meng W, Wang Y, Yu J, Bagavathiannan M (2023) Drought stress impact on the performance of deep convolutional neural networks for weed detection in bahiagrass. Grass Forage Sci 78:214–223

https://doi.org/10.1017/wsc.2025.10029 Published online by Cambridge University Press