An Optimized Artificial Intelligence Powered iOS Mobile App for Weed Identification

BISWAJIT BISWAL and JACKSON EDWARDS, South Carolina State University, USA

Weeds are a major burden in small and local farming communities in the United States due to the lack of technology, awareness, and education. Weed control is one of the biggest factors that affects crop production. Manual weeding gives maximum unique control of the weeds in the field. However, manual weeding has high labor intensity and high labor costs. This makes weed management difficult for small and local farmers in the state of South Carolina, resulting in loss of crop yield and poor quality production. In this work, the AI-based iOS mobile app is used to identify weed plants. In our work, we have successfully implemented an iOS mobile app to capture a weed image and identifies weed-type using CNN, CreateML, Xcode and Swift programming language. Our model was tested with our database of weed plants with a precision of 96%. Our results show that the iOS mobile app developed successfully identifies the weed plant. Our future work will include testing the iOS mobile app with more weed plant data to provide precise weed identification.

Additional Key Words and Phrases: Artificial Intelligence, CNN, iOS app, Weed Plant, CreateML, Core ML, Swift programming

ACM Reference Format:
Biswajit Biswal and Jackson Edwards. 2026. An Optimized Artificial Intelligence Powered iOS Mobile App for Weed Identification. 1, 1 (March 2026), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Small and local farming communities in the United States use customized weed management for their crops. Weed management includes technology in agriculture, soil study, fertilizer, custom herbicides, weed control, etc. Recent climate changes and increasing environmental pollution make weed control difficult for small and local farmers, resulting in lower crop yield and lower crop quality [1]. According to a survey [2], the annual loss of crop production caused by weeds worldwide is equivalent to the annual food ration needed for more than a billion people. This indicates the severity of weeds on crop production and food security. Thus, weed control is essential in crop production with quality assurance.

Manual weeding is fundamental for the most precise control of weeds in the agricultural field, but it is a high labor cost and highly labor intensive [3]. This makes weed management challenging and difficult for small and local farming communities in the state of South Carolina, resulting in loss of crop production and poor quality crops. Thus, it is vital to use technologies such as AI, Computer Vision (CV) to identify and categorize weeds, and improve the identification accuracies of weeds in fields. High-definition imaging devices, such as smart mobile phones and drones, allow data collection, automatic feature extraction, and precise identification of weeds. Thus, these technologies help farming communities with customized weed management while reducing labor and increasing product yield and quality.

Authors' Contact Information: Biswajit Biswal, bbiswaji@scsu.edu; Jackson Edwards, jedwar15@scsu.edu, South Carolina State University, Orangeburg, SC, USA.

Smart phone apps (i.e., iOS mobile apps) are a growing technology that uses AI and CV techniques to identify and classify images of weeds from agricultural fields. These apps allow farmers to capture weed images and instantly provide details of the weed, allowing farmers to have a customized weed management.

This work proposed a mobile application powered by AI designed to automate the process of weed plant identification. This app harnesses the computational power of trained, validated, and tested Convolutional Neural Networks (CNN) [4], using an image dataset comprised of 500 weed plant images, facilitating the classification of the ten most common weed categories.

Our contributions in this work are to build a database of weed plants collected from our partner local farms, as well as our research and demonstration farm, and to develop a user-friendly iOS mobile app that allows farmers to capture images of suspected weeds and rapidly receive the classified weed category along with a confidence score. Upon evaluating the app, it shows an impressive classification accuracy of 96%, demonstrating the efficiency and reliability of the app.

The rest of this paper is prepared as follows. In Section II, we review related work, focusing on existing smart phone-based weed identification and their limitations. Section III provides an overview of our work AI based iOS app architecture. Section IV details the implementation of the CNN model. Section V presents results, analysis, and performance comparison. Finally, Section VI concludes the paper, summarizing our contributions and discussing future directions to improve the iOS app for the identification of weed plants.

2 Related Work

Limited work has been done on farmer-friendly iOS smart apps for weed identification. Thus, it remains a critical gap in the literature. In this section, we examined previous research efforts on weed identification using advanced image analysis techniques such as AI, data science, mobile apps.

Mishra and Gautam [5] presented a review article on weed classification and detection using AI and image processing techniques. In their survey, they represented challenges and solutions for weed classification and detection under varying lighting conditions and growth stages.

Peteinatos et al. [6] proposed convolutional neural networks for the identification of weeds in maize, sunflower and potatoes. Three different convolutional neural networks (CNN), namely Visual Geometry Group with 16 layers deep (VGG-16), Residual Network with 50 layers deep (ResNet−50) and Extreme Inception (Xception), were used and trained on a pool of 93,000 images. Their reported test accuracies were between 77% and 98% for plant detection and weed species discrimination, respectively.

Wang et al. [7] presented deep learning detection models such as You Only Look Once (YOLO), YOLOv3, YOLOv5, and Faster Region-Based Convolutional Neural Network (Faster R- CNN) to detect field weeds. These models were trained using their Weed25 data set, which contained 14,035 images of 25 different weed species. Their results showed that the average accuracies of weed detection under the same training parameters were 91.8% (YOLOv3), 92.4% (YOLOv5) and 92.15% (Faster R-CNN).

A new simple CNN architecture proposed and built by [8] to classify 12 classes (three crops and nine weeds) during their early stages of growth. The results showed that the more training classes,the harder it is to achieve a good classification. However, the 12-class CNN achieved an average test accuracy of 94.38%.

Olsen et al. [9] introduced a multiclass image data set of weed species, Deepweeds, which contains eight invasive weed species and was collected entirely under natural field conditions. They trained two CNNs such as Inception-v3

Table 1. Summary of Related Work

Related Work	CNN used	Smart App based	iOS App based
[7–10, 14, 17]	✓	✗	✗
[15]	✗	✓	✗
[16, 18]	✓	✓	✗
this work	✓	✓	✓

[10] based on GoogLeNet [11] and ResNet−50 [12] for 100 epochs and their reported average classification accuracies were 95.1% (Inception-v3) and 95.7% (ResNet−50), respectively.

Dos Santos Ferreira et al. [13] used a neural network, AlexNet, pretrained on the ImageNet dataset to classify four classes of weeds such as soil, Glycine max, grass and broadleaf weeds from drone data. Their model achieved an average accuracy of 99.5%.

Rahman et al. [14] presented hierarchical crowdsourcing based on smartphones for the identification of weeds. They used two types of crowdsourcing, a nonexpert crowd contributed by Amazon Mechanical Turk (AMT), and an expert based crowd such as county extension agents. Then they used a probabilistic decision to determine the suitability of the two levels of crowdsourcing to identify the weed from the image. Evaluation of their designed system identified weeds with an accuracy of 80% using the low-cost AMT crowd, while incurring a maximum latency of 3 hours.

Jayasundara et al. [15] used a customized ResNet model the U-shaped fully convolutional network architecture (U-net, FCN) for weed management in the rice field. This model provided a training accuracy of 94.88% and a validation accuracy of 64.39%. They developed a mobile application which uses their model to help farmers to analyze weeds and recommend suitable solutions for weed management.

Jin et al. [16] evaluated the effectiveness of using deep convolutional neural networks (DCNNs) to detect and discriminate grass weeds based on their susceptibility to Acetyl CoA Carboxylase (ACCase) Inhibitors and synthetic auxin herbicides. In their proposed work, they trained GoogLeNet, MobileNet-v3, ShuffleNet-v2, and VGGNet to discriminate vegetation into three categories based on the herbicide weed control spectrum. ShuffleNet-v2 and VGGNet showed a high overall accuracy of 99.9% and F1 scores 99.8% in the validation and testing datasets to detect and discriminate weeds susceptible to synthetic auxin herbicides and ACCase inhibitors. The inference time of ShuffleNet-v2 changed into just like MobileNet-v3, however, exceptionally quicker than GoogLeNet and VGGNet. ShuffleNet- v2 turned out to be the most coherent and reliable version of the neural networks evaluated.

Ahmed et al. [17] developed a weed classification engine to classify eight categories of weeds. In their system, a CNN model was used as the core mechanism of the engine. Intel OpenVINO toolkit was used to convert their trained CNN model into an optimized Intermediate Representation (IR) model before it was integrated into a mobile device for weed classification.

As shown in Table 1, most of the work used the CNN architecture for weed detection but did not implement a smart app for farmer-friendly use. This limitation remains a critical gap in the literature, especially in the AI-powered iOS app for weed identification.

3 System Overview

In our work, the approach for the identification of weed plants is based on AI-integrated smart phone application (iOS app). Fig. 1 shows the system overview of our approach. The AI/CreateML tool leverages the AI and machine learning infrastructures to create customized models that take less time to train. The weed plant database is created through farmers' crowdsourcing and the student assistant cohort. The iOS mobile app is designed to capture weed plant images and then it allows to classify the plant image as weed or not with a probability score.

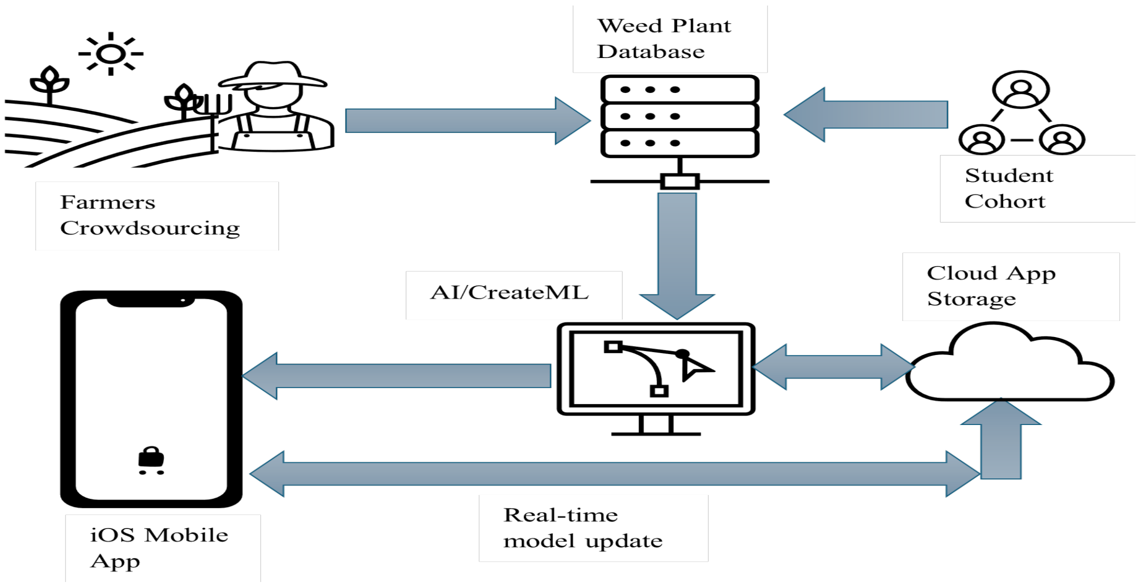

Fig. 1. System Overview

The Cloud App Storage keeps different version of models that can be downloaded to iOS app based on the requirement and usage. However, the models can be integrated directly to iOS mobile devices if connected to Xcode devices where models are trained and tested.

3.1 Weed Plant Database

Now a days, smart phone camera sensors are known to capture highly sensitive, high-quality images. In our work, the student assistant cohort collected images of weed plants using hand-held smart phones from the 1890 research and demonstration farm. Farmer crowdsourcing is from participating local farms who are compensated with smart phones for their participation in our project providing weed images from their farms. Five farmers participated and received an advantage from our project. Weed images were collected at regular intervals during different growth periods because all available weed plant databases are quite outdated with light and quality image issues and are creating challenges of occluded feature extraction. In addition, we have added few weed plant datasets that are identified from plants database [18].

3.2 CreateML based iOS App

CreateML is a powerful tool developed by Apple Inc. that helps to build machine learning models without any coding. CreateML provides a graphical user interface (GUI) and a set of tools for training and testing machine learning models using a variety of data types such as images, video, sound, text, tabular data, etc. CreateML used a deep learning image classification model to train on data and automatically train a neural network based on the image classification template;

Once trained, the model is evaluated and then exported as a CoreML model to integrate into our iOS app. The following steps are followed to create our model.

- Open Xcode and go to “File" > “Open Developer Tool" > “Create ML" to access CreateML.
- Select “New Project" and choose the deep learning model template “Image Classification".
- Drag and drop the labeled weed plant training image data into the designated “Training Data" area.
- Update training parameters such as number of epochs, learning rate, and network architecture.
- Select augmentations as per requirements. It is essential because there is less chance that the model will just remember the data instead of learning from them.
- Start the model training process by clicking on “Train".
- Analyze and evaluate the accuracy of the model in testing data.
- click “Get" to export the trained model as a Core ML file which is seamlessly integrated into our iOS app.

3.3 Artificial Intelligence (CNN Architecture)

For processing image structures and extracting features, an exceptional deep neural network called CNN is used as shown in fig. 2. CNN architecture is designed by connecting several layers, such as convolutional layers, pooling layers, and a fully connected layer (Dense). The image features are adaptively and automatically learned by applying kernel filters. The pooling layer is then applied to the output of the convolutional layer to reduce the dimension of the data. Convolution and pooling operations are repeated several times before being fed into a set of fully connected layers, which perform classification based on assigned weights and biases. Then a SoftMax activation function is used to determine the probability scores for class labels of the input image.

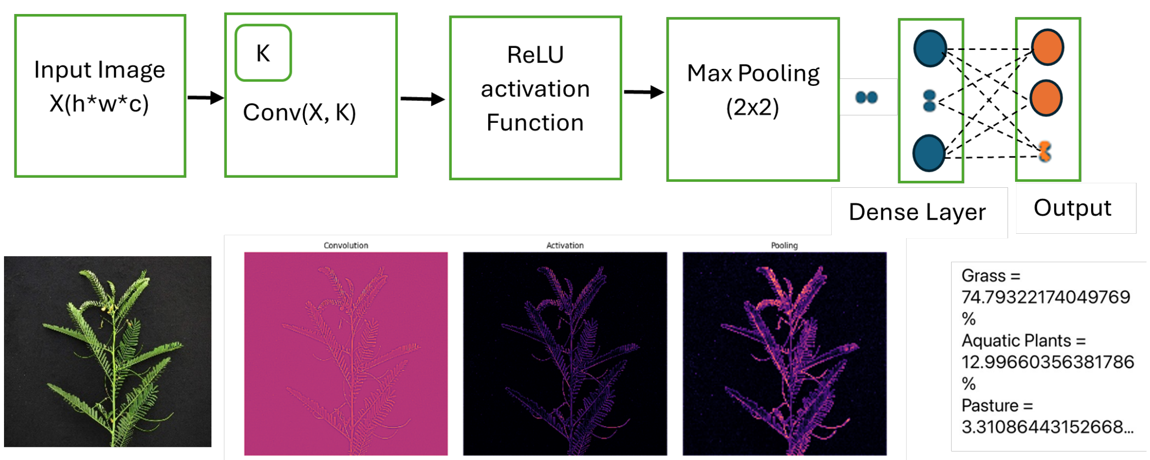

Fig. 2. CNN Architecture

Suppose that the input image is represented as X of dimension (h*w*c), where h, w, c are image height, width, and the number of channels (three channels for RGB images), respectively. Next, a dimension K kernel matrix (m*n*c) is selected, where m, n, c represent rows, columns, and the number of channels (the same number of channels as the image), respectively. Performing the convolution operation between X and K along with a biased parameter ‘b’, produced a feature map F of dimension ((h- m+1) * (w - n +1) * 1). The ij-th entry of the feature map is given as in:

$$F[i,j] = \sum_{x=1}^{h}\sum_{y=1}^{w}\sum_{z=1}^{c} K_{[x,y,z]} X_{[i+x-1,j+y-1,z]} + b \tag{1}$$

To eliminate negative values from the convoluted output, the rectified linear unit (ReLU) activation function is applied as in:

$$ReLU(a) = a^{+} = max(0,a) = \begin{cases} a, & \text{if } a > 0 \\ 0, & \text{if } a \leq 0 \end{cases} \tag{2}$$

where a is the input to the ReLU function. Next, a maximum pooling function is applied to the result obtained from the convolution layer after the activation function. In our work, a 2x2 patch is applied to the input to the max-pooling layer, and from each patch, the maximum value is selected. Thus, the output of the pooling layer 'P' has a dimension of (h+2p-m/s)* (w+2p-n/s)* c where 'p' represents zero padding and 's' as a stride. The max-pooling operation is implemented (if we denote the input as A and the output as P) as:

$$P[i,j,k] = max(A[2i : 2i+2, 2j : 2j+2, k]) \tag{3}$$

The pooled part P is then passed through a fully connected neural network where each neuron performs a dot product between the input vector and the associated balanced weight w along with a biased parameter $\mathbf{b'}$ and has full connectivity with all other neurons in the preceding and succeeding layer. If z is the output of the fully connected layer and l represents the length of the input vector, mathematically z represented as:

$$z = \sum_{i=1}^{l} w_i P_i + b' \tag{4}$$

Then the softmax activation function as in (5) is applied to the raw output (z) of the neural network and returns a vector of probability scores for class labels of the input image. The softmax output vector softmax(z) can be thought of as the predicted probability of the test input belonging to class label i, given as:

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_i}} \tag{5}$$

where N is the number of class labels. Before importing our trained CNN model to the iOS mobile device, we had to convert it into a CoreML model package. We used the CoreML toolkit to convert the CNN model to the mlpackage supported by the iOS neural network model, which is the only format that iOS and MacOS accept and understand.

4 IMPLEMENTATION AND ENVIRONMENT SETUP

4.1 CNN Implementation and Training

The CNN model is implemented using the Keras 3.0 development environment [19]. Keras is an open source neural network library written in Python using TensorFlow 2.16 as the back-end engine. Keras libraries running on top of TensorFlow make it easier for developers to build ML models written in Python.

The CNN model was trained with colored images (RGB) with resized dimensions of 256x256 pixels. We set the batch size 50 images and the number of epochs to 100. Model training was carried out using a Lambda-TensorBook with 1 GPU computer equipped with a 5.1GHz Intel CoreTM i7 CPU processor, 32GB of RAM, and CUDA GPU capability. The training phase took approximately 45 minutes to run 100 epochs. At every 2 epochs, trained weights are recorded to monitor the progress.

4.2 iOS Mobile App

The iOS app based weed plant detector's front-end user interface is implemented using the Swift programming language in Xcode on a MacOS. The iOS mobile app allows farmers to photograph the suspected weed plant or upload an existing image on the iPhone with proper alignment and orientation. The orientation handler, which runs as a background service thread in the mobile app, is responsible for correcting the tilt and camera angle of capturing the plant photo. Once the right image is captured, the app detects the weed classes by applying our CNN model. The captured image is later transferred to the iCloud storage.

5 Result And Analysis

In our work, we first used CreateML running on macOS. The training, validation and testing of the CreateML model are shown in fig. 3, fig. 4, and fig. 5 respectively. The training accuracy is 100%, but the validation accuracy is 72.2.%, this is due to the existence of duplicate images in multiple categories of weed plants. However, the model testing output resulted in the identification of a category of weeds with a 96% accuracy.

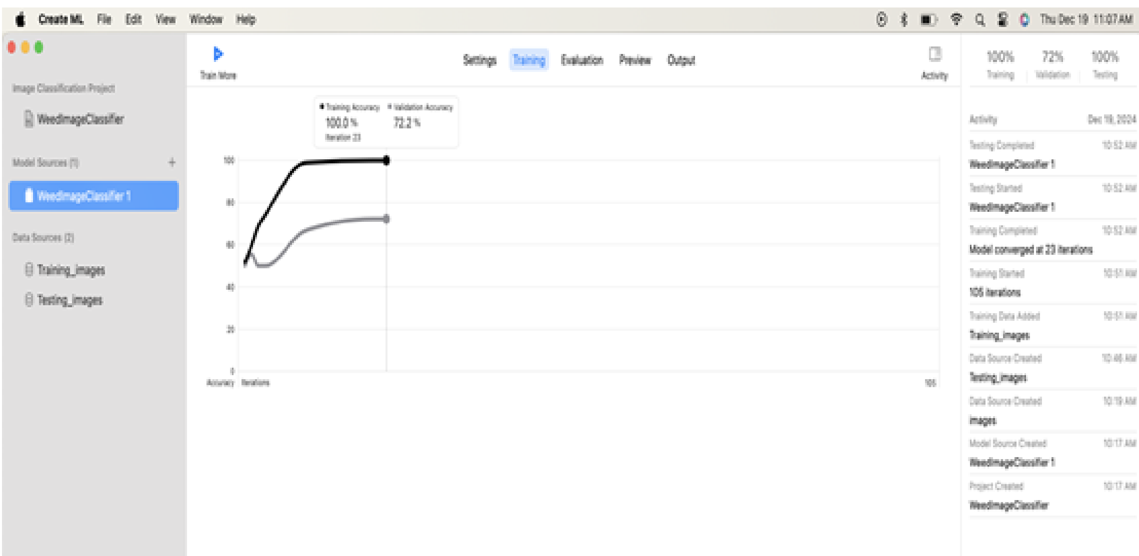

Fig. 3. CreateML training accuracy 100% and validation accuracy 72.2%

Next, we trained, validated, and tested our CNN model implemented in Python on our own dataset before converting the model into a CoreML package (.mlpackage) that was used in the iOS app. The sample image data is shown fig. 6.Training and validation accuracies of the CNN model are shown in fig. 7 and the confusion matrix in fig. 8. The wrong predictions of our model on test set are shown in fig. 9. The iOS mobile app shown in fig. 10. Finally, we compared our iOS app with the commercially available "Plant App: Plant Identifier" [20]. Both apps use AI plant identification technology for weed plant identification. The Plant App claims a 95% accuracy in plant identification accuracy while

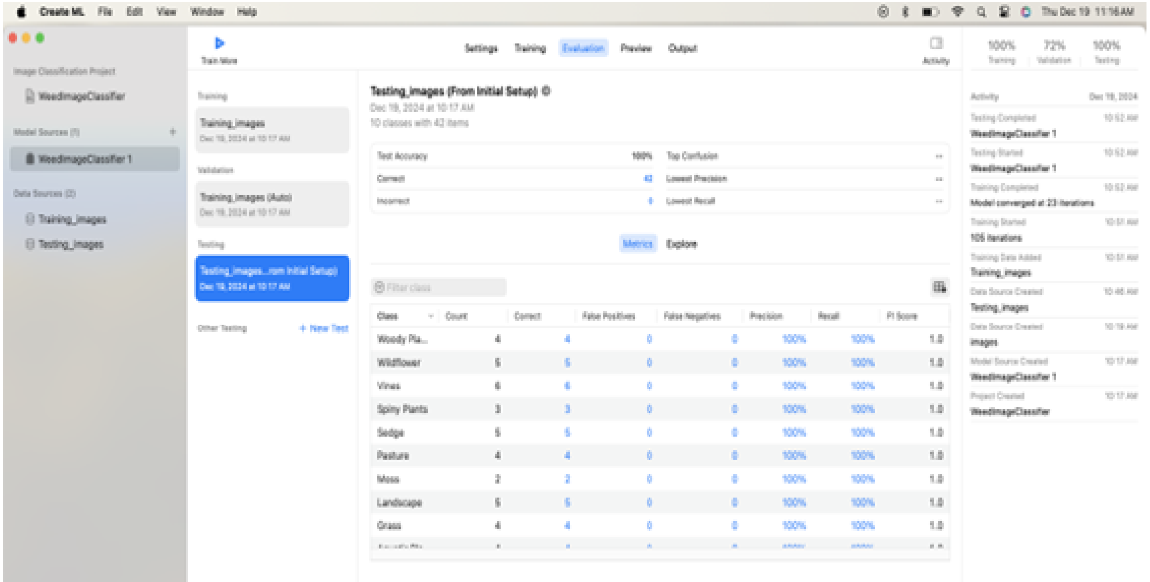

Fig. 4. CreateML Training Validation

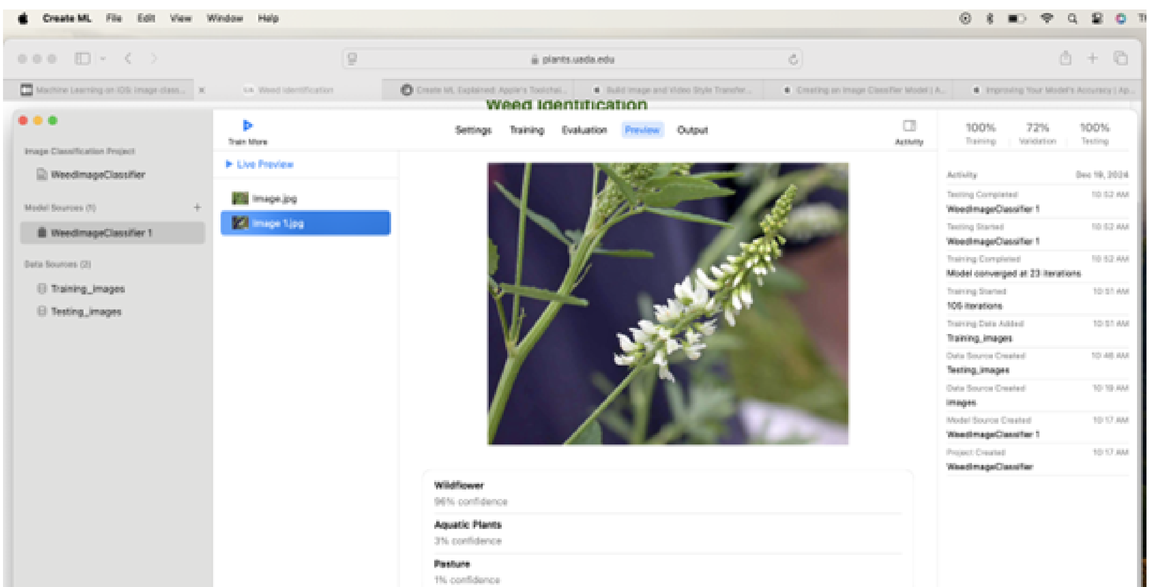

Fig. 5. CreateML Testing accuracy with 96%

our implemented app accuracy is 96%. Both apps tested with 10 input images of weed plants and the Plant App correctly predicted 8 images with a 80% accuracy, while our iOS app predicted with a 90% accuracy. Fig. **??** shows some wrong predictions from the Plant App.

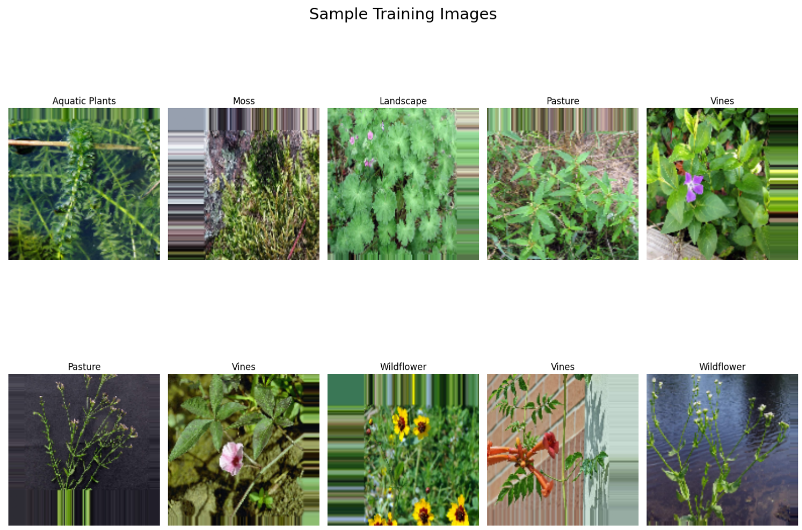

Fig. 6. Sample weed images

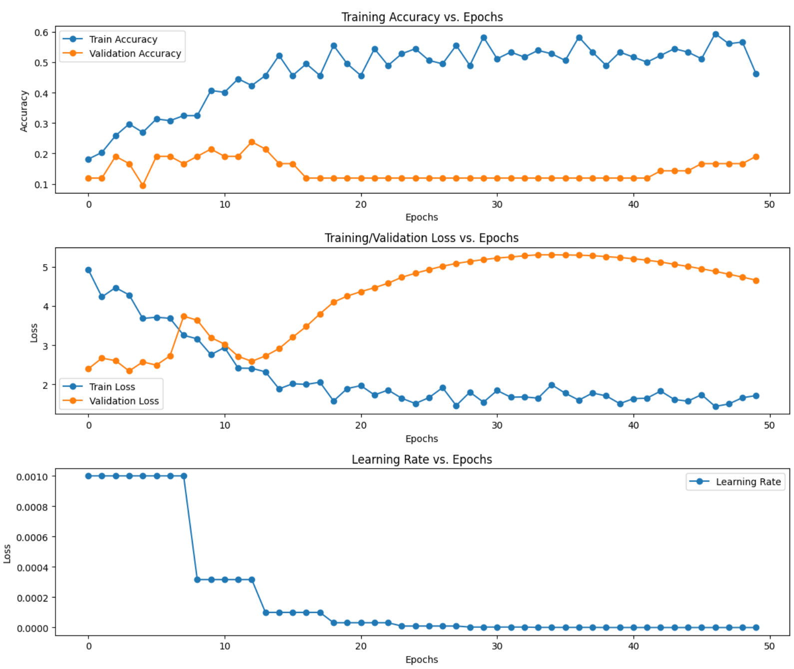

Fig. 7. CNN training and validation accuracies

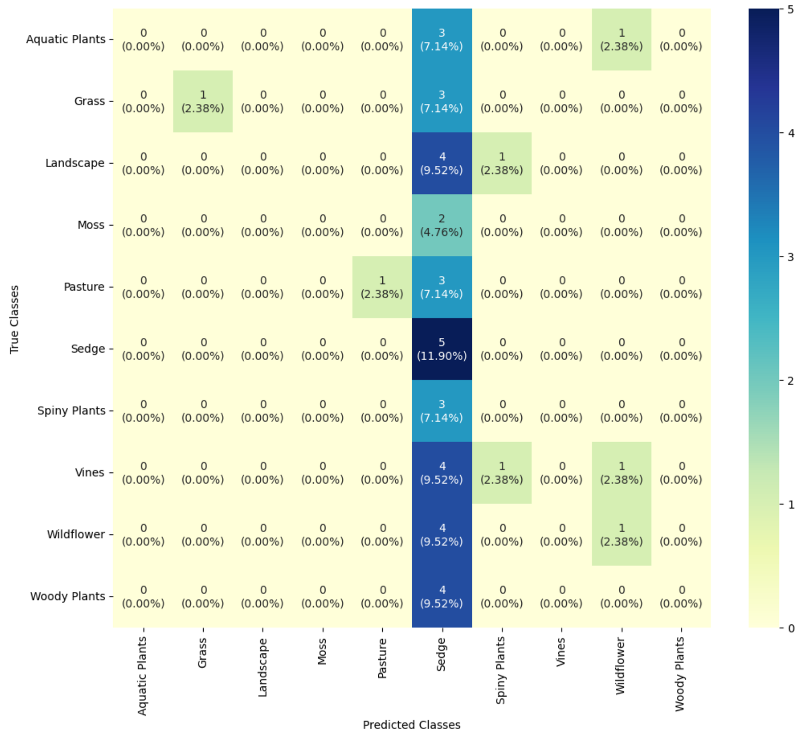

Fig. 8. Confusion Matrix

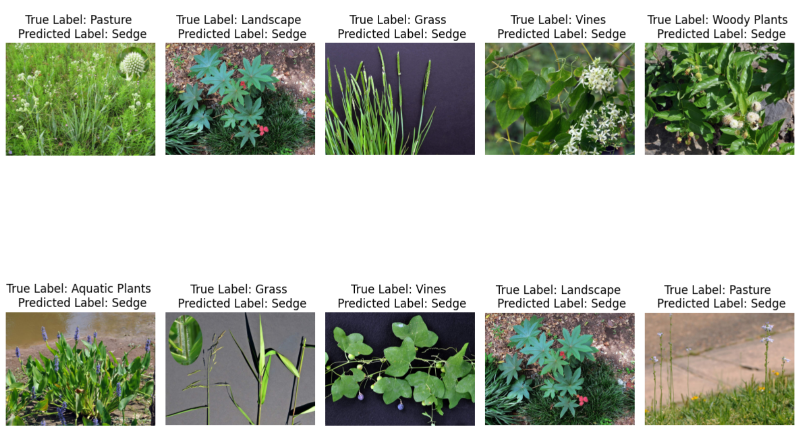

Fig. 9. Wrong predictions on test dataset

6 CONCLUSIONS AND FUTURE WORK

This paper presented an AI-powered iOS weed plant detector that helps local farmers identify the ten common weed categories. We first trained, validated, and tested a CreateML model on our weed plant data to see the performance

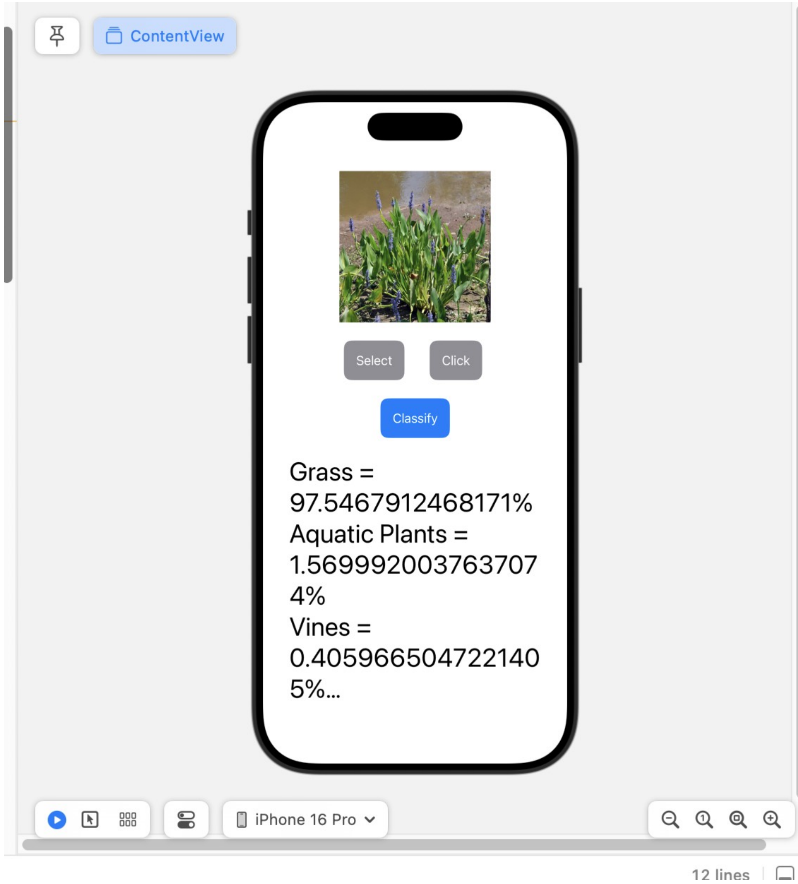

Fig. 10. Screenshot of our iOS App

of the model. Next, we trained, validated and tested a CNN model (faster r-CNN) on our weed plant data to see the model's performance. Finally, both models are integrated into an iOS mobile app developed for the iPhone. The iOS mobile app would create a better opportunity for farmers with limited resources to help them with weed identification. These advanced imaging and computing mobile applications are expected to create a better opportunity for farmers to keep their crops healthy and help to choose the right fertilizers that could contribute to high crop yielding. For our future work, we will collect and build a better database of weeds.

Acknowledgment

This work is supported by the National Institute of Food and Agriculture, U.S. Department of Agriculture, Evans-Allen Project Number SCX-314-05-23.

References

[1] Khan S., Tufail M., Khan M. T., Khan Z. A., Anwar S. (2021). Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. Precis. Agric. 22, 1711–1727. doi: 10.1007/s11119-021-09808-9.

[2] Yuan H., Zhao N., Chen M. (2020). Research progress and prospect of field weed recognition based on image processing. J. Agric. Machinery 51, 323–334.
[3] Gašparović M., Zrinjski M., Barković D., Radočaj D. (2020). An automatic method for weed mapping in oat fields based on UAV imagery. Comput. Electron. Agric. 173, 105385. doi: 10.1016/j.compag.2020.105385.
[4] S. Ren, K. He, and R. G. et al., "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.39, no.6, pp.1137–1149,2017.
[5] Mishra, Anand & Gautam, Vinay. (2021). Weed Species Identification in Different Crops using Precision Weed Management: A Review.
[6] Peteinatos G. G., Reichel P., Karouta J., Andújar D., Gerhards R. (2020). Weed identification in maize, sunflower, and potatoes with the aid of convolutional neural networks. Remote Sens. 12, 4185. doi: 10.3390/rs12244185.
[7] Wang P, Tang Y, Luo F, Wang L, Li C, Niu Q, Li H., "Weed25: A deep learning dataset for weed identification," Front Plant Sci. 2022 Nov 30;13:1053329. doi: 10.3389/fpls.2022.1053329. PMID: 36531369; PMCID: PMC9748680
[8] Elnemr, H.A., "Convolutional Neural Network Architecture for Plant Seedling Classification," Int. J. Adv. Comput. Sci. Appl. 2019, 10.
[9] 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.
[10] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z., "Rethinking the Inception Architecture for Computer Vision," In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
[11] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A., "Going deeper with convolutions," In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
[12] He, K.; Zhang, X.; Ren, S.; Sun, J., "Deep Residual Learning for Image Recognition," In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016.
[13] Dos Santos Ferreira, A.; Freitas, D.M.; da Silva, G.G.; Pistori, H.; Folhes, M.T., "Weed detection in soybean crops using ConvNets," Comput. Electron. Agric. 2017, 143, 314–324.
[14] Rahman, M.; Blackwell, B.; Banerjee, N.; Saraswat, D., "Smartphone-based hierarchical crowdsourcing for weed identification," Computers and Electronics in Agriculture, Volume 113, 2015, Pages 14-23, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2014.12.012.
[15] Jayasundara, S.M.W. & Paranagama, Binuka & Gunarathne, G.W.D.A. & Jayasinghe, P.Y.D. & Ranthilina, Umesh & Gunasinghe, Amali. (2022). "E-KETHA": Enriching Rice Farmer's Quality of Life through a Mobile Application. International Journal of Computer Applications. 184. 68-75. 10.5120/ijca2022922364.
[16] Jin, X., Bagavathiannan, M., Maity, A., Chen, Y., & Yu, J. (2022). Deep learning for detecting herbicide weed control spectrum in turfgrass. Plant Methods, 18(1), 94.
[17] A. A. Ahmed and J. Ahmed, "An Image-Based Plant Weed Detector Using Machine Learning," 2024 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 2024, pp. 193-197, doi: 10.1109/ICNC59896.2024.10555934.
[18] "Plants Database," accessed June 25, 2024. [Online]. Available:https://plants.uada.edu/weeds/select/
[19] "Keras: A python deep learning api," accessed December 19, 2024. [Online]. Available:https://keras.io/
[20] "Plant App: Plant Identifier," accessed May 16, 2025. [Online]. Available:https://apps.apple.com/us/app/plant-app-plant-identifier/id1595795215