Classifying Weed Development Stages Using Deep Learning Methods

Classifying Weed Development Stages with DenseNET, Xception, SqueezeNET, GoogleNET, EfficientNET CNN Models Using ROI Images

Yasin ÇİÇEK[1], Eyyüp GÜLBANDILAR[2], Kadir ÇIRAY[3], Ahmet ULUDAĞ[4]
Dept. of Computer Engineering, Eskişehir Osmangazi University, Eskişehir, Turkey[1, 2]
Sinanpaşa Vocational School, Afyon Kocatepe University, Afyonkarahisar, Turkey[3]
Dept. of Plant Protection, Çanakkale Onsekiz Mart University, Çanakkale, Turkey[4]

***Abstract*—The control of harmful weeds holds a significant place in the cultivation of agricultural products. A crucial criterion in this control process is identifying the development stages of the weeds. The technique to be used is determined based on the weed's growth stage. This study addresses the application of deep learning methods in classifying growth stages using images of various weed species to predict their development periods. Four different weed species, obtained from seeds collected in Turkey-Afyonkarahisar-Sinanpaşa Plain, were used in the study. The images were captured with a Nikon D7000 camera equipped with three different lenses, and the ROI extraction was performed using Lifex software. Using these ROI images, deep learning models such as DenseNet, EfficientNet, GoogleNet, Xception, and SqueezeNet were evaluated. Performance metrics including accuracy, F1 score, precision, and recall were employed. In the 4-class dataset with ROI annotations, DenseNet and Xception achieved an accuracy of 86.57%, while EfficientNet demonstrated the highest performance with an accuracy of 89.55%. Following the initial tests, it was concluded that classes 3 and 4 exhibited extreme similarity caused most of the prediction errors. Merging the said classes significantly increased the accuracy and F1 scores across all models. In image classification tests, SqueezeNet and GoogleNet demonstrated the shortest processing times. However, while EfficientNet lagged slightly behind these models in terms of speed, it exhibited superior accuracy. In conclusion, although the use of ROI improved classification performance, class merging strategies resulted in a more significant performance enhancement.**

***Keywords*—Deep learning; weed development stages; classification; DenseNET; Xception; SqueezeNET; GoogleNET; EfficientNET; ROI**

I. INTRODUCTION

One of the foremost factors affecting yield in agriculture is weeds [1]. If weeds are not controlled in the crop plant, this effect can be more significant. Integrated weed management (IWM) is a system that provides economically bearable, biologically effective and environmentally sound weed control using possible techniques in a sustainable manner. Among the tools IWM relies on is the critical period for weed control (CPWC): a duration during the crop's growth cycle when weed control is essential [2]. In many crops, CPWC for higher yields (more acceptable yield loss) starts from the crop emerging, which requires earlier detection of weeds to achieve effective weed control [3,4].

Identifying weeds at an early stage can be challenging because they may closely resemble crop plants. This task becomes even more difficult for farmers who must search for and identify weeds across the entire field, as it is an incredibly monotonous activity, leading to decreased performance and efficiency [5]. To mitigate this challenge, modern computer methods are employed. Artificial intelligence techniques used in computer-based imaging make it easier to detect weeds, thus facilitating the goal of product identification. Indeed, information technologies are being utilized in various fields of agriculture [6].

In image-based classifications, CNN and RNN deep learning models are more frequently used [7]. In agriculture, deep learning is applied in areas such as product diseases, pests, spraying, and the classification of crops and weeds [8].

A. Related Work

Espejo Garcia and colleagues developed a classification model using machine learning techniques such as Support Vector Machines (SVM), XGBoost, and logistic regression, along with convolutional networks like Xception, Inception-ResNet, VGNets, MobileNet, and DenseNet, applied to two crops and two weed species. In the dataset composed of photographs taken under natural light conditions with RGB cameras, DenseNet and SVM achieved an F1 Score of 99.29% [9].

Sunil and colleagues utilized deep learning models such as Xception, DenseNet, MobileNetV3Large, EfficientNet, and ConvNeXt to classify six weed species and eight crop types. They reported that, except for DenseNet, the other four models demonstrated strong performance, with the macro average F1 scores ranging between 0.85 and 0.87 and the weighted average F1 scores ranging between 0.87 and 0.88 [10].

Pandey and colleagues achieved classification in datasets containing maize and radish plants along with weeds using deep learning models such as InceptionV3, Xception, ResNet152V2, VGG16, and their proposed CNN model. They reported an accuracy of 97.66% for maize and 98.75% for radish [11].

Garibaldi-Marquez and colleagues conducted a classification study on maize and weeds. In their study, they used both regular images and ROI images, employing deep learning models such as ResNet101, VGG16, Xception, and MobileNetV2. Among the classification algorithms, Xception provided the best accuracy result of 97.43% [12].

Trong and colleagues conducted a study using datasets of plant seedlings and weeds from Chonnam National University. They employed five different deep learning methods, including NasNet and ResNet, to classify weeds. They achieved an accuracy of 97.31% for plant seedlings and 98.77% for weeds [13]. In another study involving ResNet, VGG16, and Xception, Peteinatos and colleagues used 93 000 images obtained with an RGB camera of 12 different weed species and applied deep learning models, reporting an accuracy range of 77% to 98% [14].

Chen and colleagues utilized a dataset comprising 5,187 images of 15 weed classes found in cotton fields. They classified these weeds using deep learning methods such as ResNeXt, Xception, and MnasNet. They achieved high accuracy, with F1 scores exceeding 95% [15].

One of the commonly used models for weed classification is the YOLO architecture. Gao and colleagues used the YOLOv3 architecture to train a model for identifying field bindweed and sugar beet plants. They based their training on 452 field images and generated 2,271 synthetic images. Their tests with 100 field plant images demonstrated that the average precision metric could reach 0.829 [16].

Ahmad and colleagues used a YOLOv3-based object detection model to identify four different weed species within maize and soybean fields. They employed pre-trained deep learning models, including VGG16, ResNet50, and InceptionV3. They reported that VGG16 achieved the highest accuracy at 98.90% and an F1 score of 99 [17].

Zhang and colleagues utilized the YOLOv3 and YOLOv3-tiny architectures to detect weeds in wheat fields. Based on the images obtained from UAVs, they indicated that YOLOv3-tiny is more suitable for mobile devices [18].

Sharpe and colleagues utilized the YOLOv3-tiny deep learning architecture to detect goosegrass in strawberries and tomatoes. They reported the following metrics for whole plants: precision = 0.93; recall = 0.88; F1-score = 0.90; accuracy = 0.82. For leaf blades, the metrics were precision = 0.39; recall = 0.55; F1-score = 0.46; accuracy = 0.30 [19]. Additionally, Osorio and colleagues used SVM and R-CNN architectures alongside YOLOv3 to detect weeds in lettuce. They found that R-CNN outperformed the other two methods, achieving an F1-score of 94% [20].

There are numerous publications related to deep learning in the agricultural sector, especially regarding the classification of weeds. Alongside the abundance of these publications, review studies that analyze, group, and evaluate these works have also started to appear in the literature [8, 21, 22].

The aim of current work is to predict the developmental stages of the plant or classify them based on the stages. Using the obtained images of weed species, the developmental stages of the weeds are going to be predicted employing deep learning methods. In addition to the plant images, predictions are also going to be made using ROI analysis on the same leaf images to enhance recognition.

II. Material and Method

The study consists of three main stages. The first stage is cultivating and photographing the weeds, the second stage is the extracting the ROI images and radiomics from the photographs, and the last stage is preparing the data set, developing the models and evaluating the performances.

A. Dataset

Seeds of Lamb's Quarters (Chenopodium album), Jerusalem Oak Goosefoot (Dysphania botrys), Prickly Lettuce (Lactuca serriola), and Sow Thistle (sonchus oleraceus), which are four common weed species in sugarbeet fields in Turkey-Afyonkarahisar-Sinanpaşa plains, were collected, dried, and threshed. Once the seeds were sown under suitable conditions, data collection began immediately as the weeds emerged, creating a comprehensive dataset. The development of the weeds was meticulously tracked twice a week using a Nikon D7000 camera equipped with three different lenses: 'Nikon 50mm 1.8 mm', 'Tokina 11-16 mm', and 'Sigma 105 mm'.

The weeds were grown in 16 different units ($2^4 = 16$), each with varying conditions of soil, irrigation, light, and fertilizer. The plants were photographed in various growth stages. In addition to the images, the leaves were analyzed using Region of Interest (ROI) techniques. Images were tested both with original and ROI marked images.

In line with this objective, ROI images of the leaves were extracted from the photos in the dataset using the Lifex program [23]. This resulted in a total of 448 photos. Five different deep learning models were applied to this dataset.

Based on the confusion matrix results, the accuracy rate was lower for the leaves in the 4th class due to their resemblance to those in the 3rd class. Consequently, the data for the 3rd and 4th classes were combined to create a new dataset. The same deep learning methods were applied to this new dataset as well.

Finally, a dataset was created using the raw, unmarked versions of the same images, and the above-mentioned deep learning models were applied again in both 3-class and 4-class formats.

B. Deep Learning Concepts

A short summary of Artifical Intelligence (AI) concepts to clarify where this study falls among AI concepts would be in order.

Machine learning is the learning stage of AI. Algorithms are developed and used on datasets to learn to perform certain tasks. It encompasses a variety of methods such as Supervised Learning, Unsupervised Learning, Reinforcement Learning and Deep Learning [24].

1) Deep learning: Deep learning is a subset of machine learning that involves training neural networks with many layers (hence "deep") to process and learn from large amounts of data. It focuses on learning representations of data through

these multiple layers. Deep learning encompasses Neural Networks [25].

2) Neural networks: Artificial Neural Networks (ANN) are machine learning algorithms consisting of a multitude of nodes where each node is connected to other nodes. Nodes communicate with each other in a predetermined pattern and their weights are adjusted every stage of the learning process depending on the success of the desired result. It is much like a biological neural system with neurons. It has been named "neural" due to this similarity [24].

3) Image classification models: There is a wide variety of ANN models especially designed for image classifications such as Resnet, GoogleNet, Xception, DenseNet, EfficientNet, SqueezeNet etc.

a) DenseNet201: The DenseNet network can be considered an extension of the ResNet model, which is a significant milestone in deep networks. The difference between the DenseNet network and the ResNet model is that DenseNet combines layers by concatenation rather than summation.

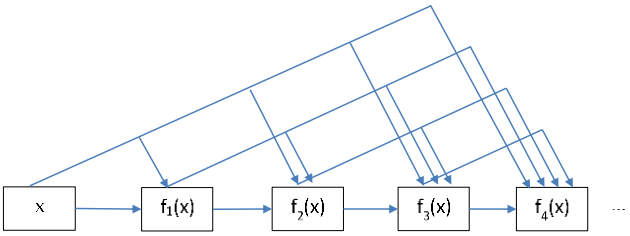

Fig. 1. DenseNET connections in DenseNET model.

Fig. 1 shows the progressively dense layer connections of DenseNet. The name "Dense" derives from these increasingly dense connections. The groups of layers where such connections are used are called dense blocks. However, the sequential addition of dense layers significantly increases the number of channels, i.e., the data pathways between layers, to astronomical levels, necessitating the need to control complexity. Transition layers are used to reduce this increased number of channels to one [26, 27].

In the DenseNet network, data is trained using RGB format images with a resolution of 224x224x3 pixels.

b) XCeption (Depthwise Separable Convolution): Xception aims to achieve the same or better performance with fewer parameters by using depthwise separation. The model is designed to have each convolution layer operate through two pathways: depthwise convolution and pointwise convolution. In the depthwise convolution pathway, a single input slice produces a single output slice, mapping only the spatial (height-width) dimensions. In the pointwise convolution pathway, a 1x1 convolution maps the color channels. This separation of the spatial and color channels results in a more lightweight model. The network comprises a total of 71 layers. In the XCeption network, data is trained using RGB format images with a resolution of 299x299x3 pixels [28, 29].

c) GoogleNet (Inception Networks): The idea behind this model was to effectively combine the methods of Network in Network and Repeated Blocks that preceded it. It was based on the concept of using combinations of convolution kernels of different sizes, ranging from 1x1 to 11x11, used in previous models. The basic structure of the inception blocks, which combine convolution kernels of different sizes, is shown in Fig. 2.

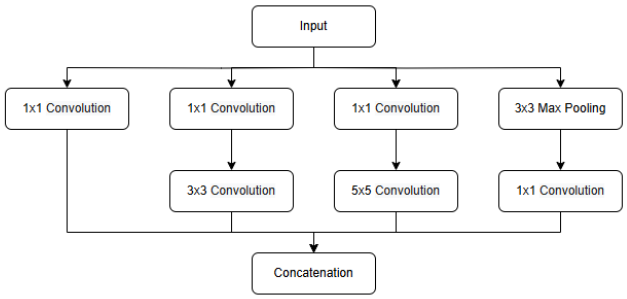

Fig. 2. Inception block structure.

Different sized convolution kernels can be thought of as filters capturing features of different sizes. Since 1x1, 3x3, and 5x5 convolution kernels capture different features in an image, inception blocks provide an efficient method for capturing features of various sizes.

The number of channels in the inception blocks was determined based on numerous experiments conducted on the ImageNET dataset. The overall structure of GoogleNET was created by serially connecting meticulously designed inception blocks to other blocks. The first version of the GoogleNET model contained 22 layers, but later versions may have more layers.

In the GoogleNet network, data is trained using RGB format images with a resolution of 224x224x3 pixels. The 224x224 image size and 3 color channels are commonly used input dimensions in the ImageNet dataset, which is widely utilized by GoogleNET and many other image classification models [27, 30].

d) SqueezeNet: SqueezeNet is a deep neural network designed to achieve high accuracy with significantly fewer parameters compared to existing models like AlexNet. The authors aimed to create a smaller, more efficient network that could be easily deployed on devices with limited memory and computational power. SqueezeNet achieves AlexNet-level accuracy on the ImageNet dataset while being 50 times smaller and requiring less computational resources. The model uses a unique architecture called the Fire Module, which consists of a squeeze layer (with 1x1 filters) followed by an expand layer (with a mix of 1x1 and 3x3 filters). The network has 18 layers and processes images with a resolution of 256x256 pixels. This makes it suitable for applications in resource-constrained environments such as mobile devices and embedded systems2 [31].

e) EfficientNet: EfficientNet is a new approach to scaling convolutional neural networks (CNNs) for better performance than previous models. Traditional methods scaled CNNs by increasing depth, width, or resolution individually, but EfficientNet proposes a balanced scaling method that uniformly scales all three dimensions using a compound coefficient. This

method, called compound scaling, allows EfficientNet to achieve state-of-the-art accuracy with fewer parameters and computational resources. The EfficientNet family includes models from B0 to B7, with B0 being the smallest and B7 being the largest. The smallest model, B0, has 24 layers and processes images with a resolution of 224x224 pixels, while the largest model, B7, has 264 layers and processes images with a resolution of 600x600 pixels. EfficientNet models outperform previous CNN architectures on various benchmarks, including ImageNet and CIFAR-1003 [32].

The selected neural network models were trained using the following parameters: the Adam optimizer, with an initial learning rate set at 0.0001, a mini-batch size of 25, and validation data provided by an augmentedImageDatastore object (referred to as augimdsValidation). The validation frequency configured at every 5 iterations. The training progress was visualized using plots, the performance metric used was accuracy, and the verbosity setting was disabled to reduce output during training. The number of epochs was maintained at the default value of 30 epochs. To ensure a fair and consistent comparison, all models were evaluated under these identical training conditions.

These settings were applied uniformly across all models to ensure the reliability and validity of the comparisons.

C. ROI (Region of Interest)

ROI is typically used to define the boundaries of a significant area within an image. Similar to its use in medical imaging to measure tumor boundaries and sizes, ROI allows for focused analysis on specific areas of interest rather than the entire image, thereby facilitating the extraction of various numerical features. Calculations such as mean and maximum values can be performed based on the defined ROI boundaries [33, 34].

In this study, ROIs of the different shape leaf images corresponding to the same plant and time intervals were used to determine the developmental stages of the leaves.

D. Why These Particular Models

Neural networks have their own advantages and weaknesses compared to each other. There are three main characteristics to consider when selecting neural networks: accuracy, speed, and size. The choice of network for use depends on the priority of these three characteristics in the intended application area. Classification accuracy and speed measurements are often conducted using the ImageNet database. However, these results do not always achieve the same success for different tasks and datasets.

It is not possible to find very precise criteria for selecting CNN models. The number of layers in the model, structural complexity, and the number of operations (flops) do not show a linear relationship with the final accuracy rates. There are models that provide high accuracy with low complexity and number of operations. Additionally, parameter settings do not yield the same level of results in every model. Model complexity can only serve as an indicator for the computer's memory usage level [35]. Considering the context in which the model is intended to be trained, it is not possible to predict which model will yield better results. Therefore five different models corresponding to promising accuracy and speed and size values. In addition, five different models that have been selected are widely used with mostly successful results.

E. Data Augmentation and Preprocessing

Deep learning methods enable the solving of various problems based on data obtained from information systems. Regardless of innovations in the design and training processes of the deep learning model, the data always affects the results. Training with a small amount of data with low representational power leads to poor generalization, while training with a large amount of data with high representational power shows higher generalization performance even with less complex algorithms [36].

The main goal of data augmentation is to enhance the robustness and accuracy of deep learning models and to allow them to perform well on small, weakly representative datasets. Effective data augmentation strategies also reduce the requirements related to model complexity, enabling high generalization performance with simpler deep learning architectures [37].

One of the biggest issues encountered in data training is overfitting. Overfitting occurs when the model makes correct predictions for training data but fails to do so for new data. On the other hand, underfitting is another error type that arises when the model fails to identify meaningful relationships between input and output data. Data augmentation helps prevent overfitting, underfitting, and the model memorizing the exact details of the training images [38]. Fig. 3 shows an example of data augmentation.

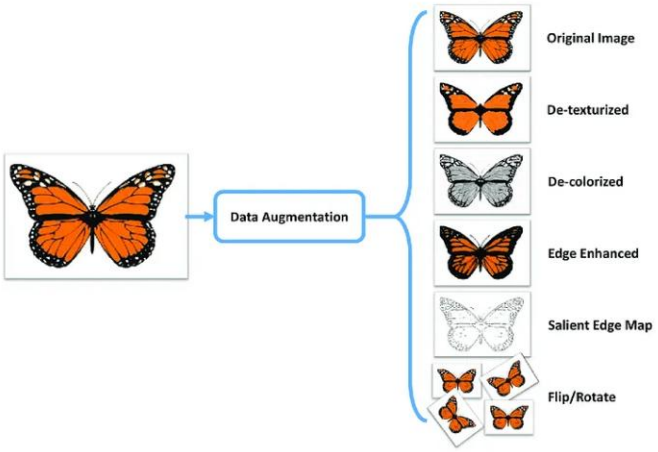

Fig. 3. Data augmentation samples [39].

The obtained datasets were used with the Matlab program to apply deep learning models such as DenseNet, EfficientNet, GoogleNet, Xception, and SqueezeNet. Four different datasets were run with the models, including 2, 3, and 4-class datasets, both with and without ROI markings. The datasets were divided into 70% for training, 15% for testing, and 15% for validation.

All modeling was conducted using an Nvidia RTX 4060 graphics card with CUDA cores. The 3-class dataset contained 128, 128, and 192 photos, respectively, while the 4-class dataset contained 128, 128, 128, and 64 photos, respectively. The batch

size was set to 25, and the validation frequency was set to 5. All models were trained in 30 epochs. In all models, the results of Precision, Recall, F1 score, and Accuracy metrics [40], along with confusion matrices and accuracy graphs that included the runtime, were recorded.

III. RESULTS

The average performance of the four-class, five-class models obtained using ROIs for each class is shown in Table I, and the confusion matrices of the models are depicted in Fig. 4.

TABLE I. THE PERFORMANCE OF 4-CLASS OUTPUT VARIABLES WITH ROI MARKED IMAGES FOR EACH MODEL

	Precision	Recall	F1 Score	Acc	Time
Densenet	0,8224	0,8480	0,8280	0,8657	12:47
Efficient	0,8724	0,8807	0,8759	**0,8955**	06:07
Googlenet	0,7711	0,7906	0,7731	0,8209	**03:29**
Xception	0,8224	0,8660	0,8306	0,8657	21:02
Squeezenet	0,6684	0,6875	0,6610	0,7313	**03:27**

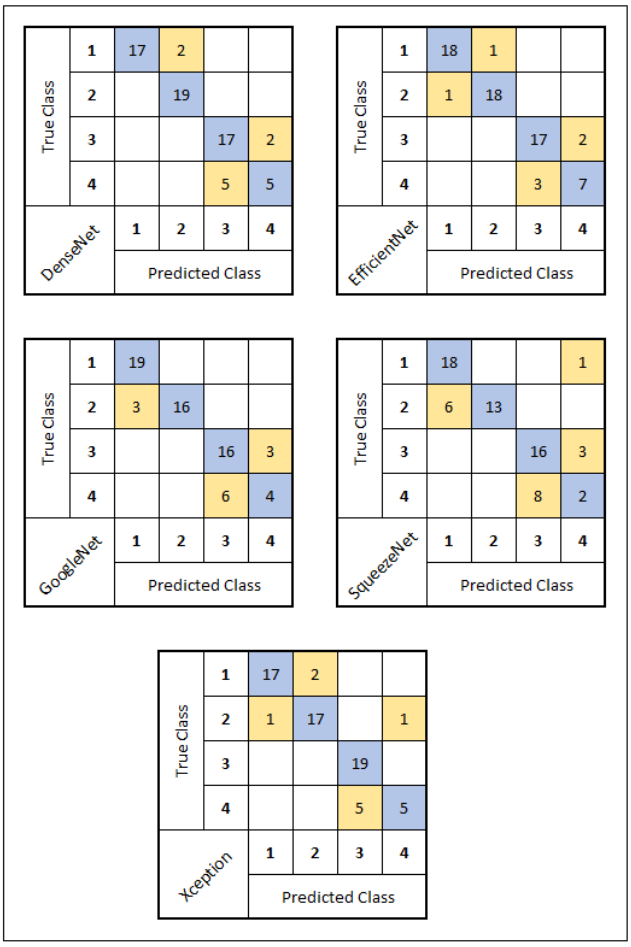

Fig. 4. The confusion matrices of the 4-class output variables with ROI marked images.

From Table I and Fig. 4, it is observed that accuracy measures belong to the DenseNet, EfficientNet, GoogleNet, and Xception models are quite high (>0.8). The accuracy rate of the SqueezeNet model is around 0.73. A similar situation is seen in the F1 score, precision, and recall values. These results indicate that the performance of the first four models is significantly higher than that of the SqueezeNet model. However, when looking at the runtime, it is observed that the runtimes of the SqueezeNet and GoogleNet models are significantly shorter than others. According to these results, the EfficientNet model is seen to have a high superiority both in terms of performance and runtime.

TABLE II. THE PERFORMANCE OF 3-CLASS OUTPUT VARIABLES WITH ROI MARKED IMAGES FOR EACH MODEL

	Precision	Recall	F1 Score	Acc	Time
Densenet	0,9298	0,9328	0,9296	0,9403	19:06
Efficient	0,9474	0,9545	0,9470	0,9552	06:06
Googlenet	0,9649	0,9649	0,9649	**0,9701**	03:42
Xception	0,9298	0,9360	0,9317	0,9403	13:33
Squeezenet	0,9123	0,9168	0,9142	0,9254	**01:47**

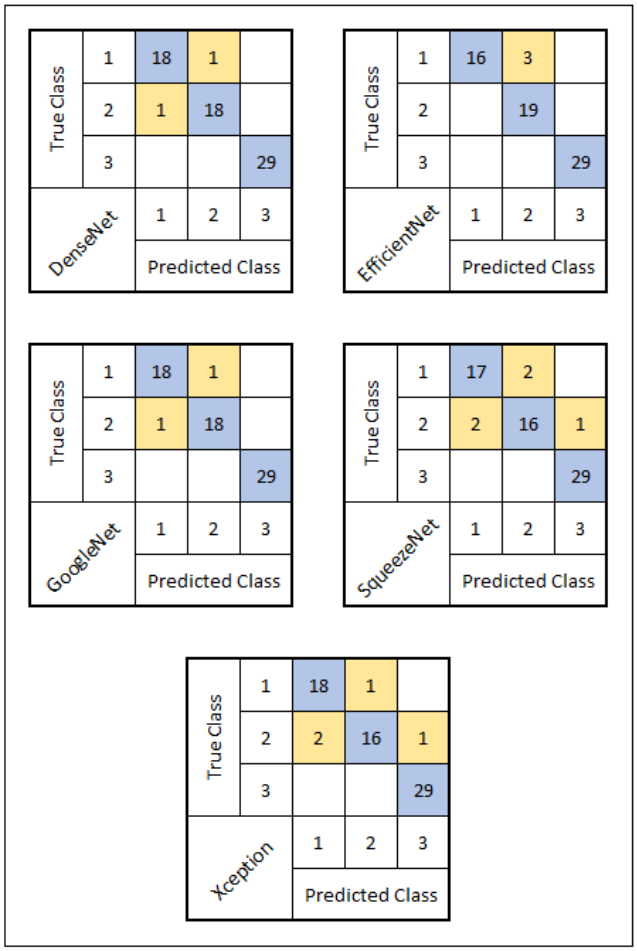

Fig. 5. The confusion matrices of the 3-class output variables with ROI marked images.

From Fig. 4, the prediction error rate between the 3rd and 4th classes is relatively high. This error is most likely due to the similar leaf sizes of the plants in these classes. To improve the accuracy rates of the developed model performances, the dataset was updated by merging the 3rd and 4th classes. Upon examining Table II and Fig. 5, it is observed that all models show high performance measurement metrics (>0.9). However, when looking at the runtime, it is noted that the runtime of the SqueezeNet models is quite short. According to these results, the GoogleNet model has a high superiority in terms of both performance and runtime.

TABLE III. THE PERFORMANCE OF THE 4-CLASS OUTPUT VARIABLES WITH ORIGINAL IMAGES WITHOUT ROI MARKINGS FOR EACH MODEL

	Precision	Recall	F1 Score	Acc	Time
Densenet	0,9487	0,9406	0,9440	**0,9552**	11:22
Efficient	0,8842	0,8748	0,8770	0,8955	04:29
Googlenet	0,7276	0,7272	0,7158	0,7313	02:19
Xception	0,9342	0,9260	0,9287	0,9254	13:22
Squeezenet	0,9605	0,9611	0,9605	**0,9552**	**00:53**

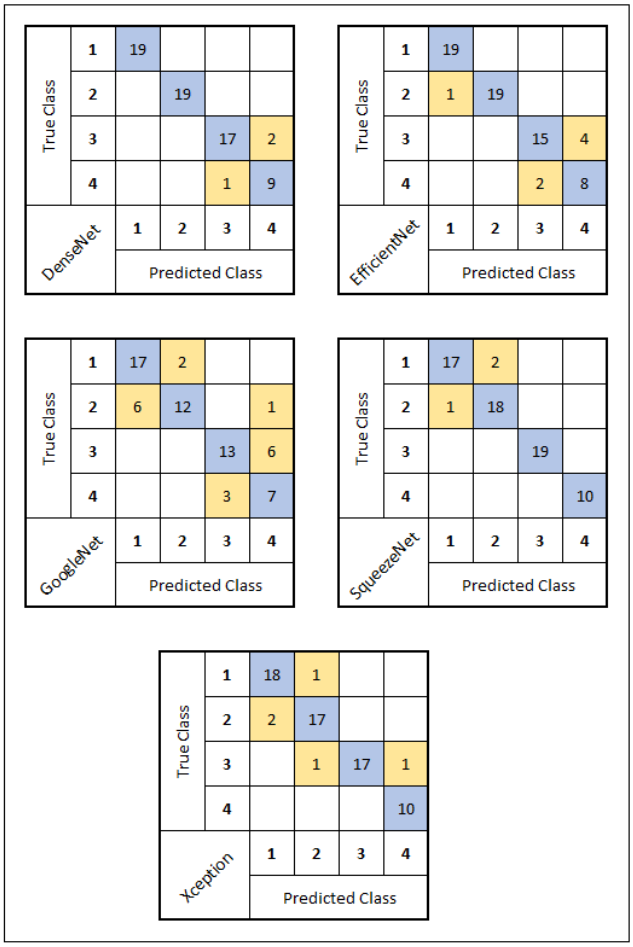

Fig. 6. The confusion matrices of the 4-class output variables with original images without ROI markings.

A 4-class database was created using photos without ROI markings. The objective here is to observe the effect of ROI marking on model performance. Upon examining Table III and Fig. 6(a), it is observed that the performance measurement metrics of the DenseNet, EfficientNet, Xception, and SqueezeNet models are higher than those of the GoogleNet model (>0.89). However, looking at the runtime, it is noted that the runtime of the SqueezeNet model is quite short. These results indicate that the SqueezeNet model has a high superiority in terms of both performance and runtime.

TABLE IV. THE PERFORMANCE OF THE 3-CLASS OUTPUT VARIABLES WITH ORIGINAL IMAGES WITHOUT ROI MARKINGS FOR EACH MODEL

	Precision	Recall	F1 Score	Acc	Time
DenseNet	0,9649	0,9683	0,9648	0,9701	23:34
Efficient	1,0000	1,0000	1,0000	**1**	07:26
GoogleNet	0,9474	0,6781	0,9473	0,9552	01:45
XCeption	0,9474	0,9545	0,9470	0,9552	16:14
Squeezenet	0,9359	0,9431	0,9355	0,9403	**01:08**

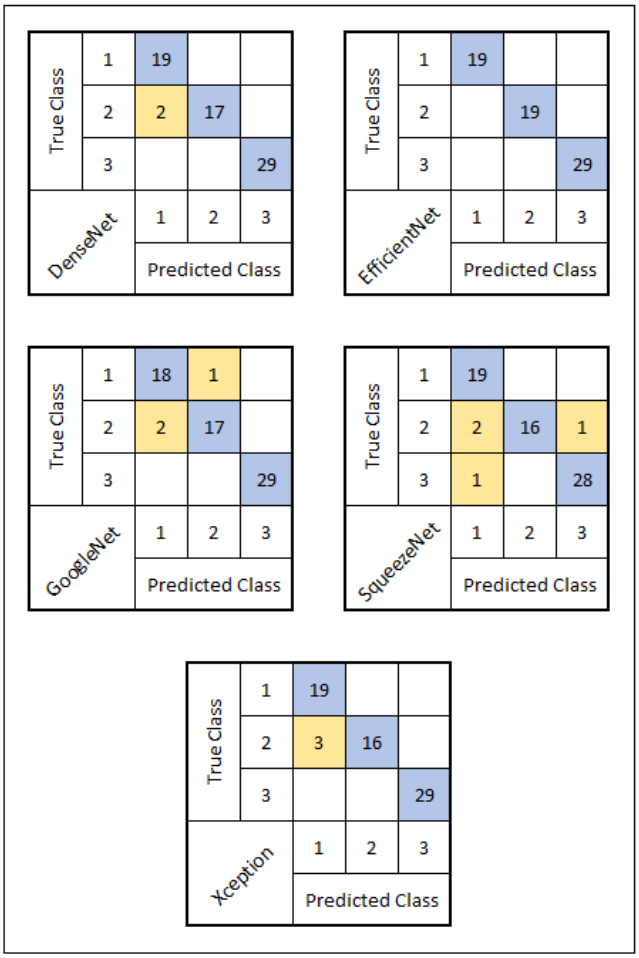

Fig. 7. The confusion matrices of the 3-class output variables with original images without ROI markings.

As observed in the dataset with ROI markings, the prediction error rate between the 3rd and 4th classes in photos without ROI markings is relatively high, as shown in Fig. 6. To improve the accuracy rates of the model performances, the dataset was updated by merging the 3rd and 4th classes, similar to the previous approach. Upon examining Table IV and Fig. 7, it is observed that the performance measurement metrics of all models are high (>0.94). However, looking at the runtime, it is noted that the runtime of the SqueezeNet model is quite short. According to these results, the EfficientNet model has high superiority in terms of performance, while the GoogleNet and SqueezeNet models have high superiority in terms of runtime.

IV. CONCLUSION

The aim of this study was to predict the growth stages of plants using a dataset of weed photographs through deep learning methods. The growth stages of the plants were divided into four different stages (classes). In line with this objective, ROI images of the leaves were extracted from the photos in the dataset using the Lifex program. This resulted in a total of 448 photos. Five different deep learning models were applied to this dataset.

According to the confusion matrix results, the accuracy rate for the leaves in the 4th class was lower because they resembled those in the 3rd class. As a result, the data for the 3rd and 4th classes were merged to form a new dataset, and the same deep learning methods were applied to this revised dataset.

Finally, a dataset was created using the raw, unmarked versions of the same images, and the previously mentioned deep learning models were applied again in both 3-class and 4-class formats.

In general, when the performance metrics of the models applied to the 4-class dataset marked with ROI was evaluated, no significant differences were observed (4 class ROI F1 Score: 0.8759). However, when the 3rd and 4th classes in the dataset were combined, an increase in the performance metrics of all models was observed (3 class F1 Score: 0.9649). Similarly, in the dataset created without ROI marking on the photos, the measurement parameters were found to be superior compared to the other conditions (4 class F1 Score: 96.05- 3 class F1 Score: 1.00).

This study is limited to the 4 weed species which are Lamb's Quarters (Chenopodium album), Jerusalem Oak Goosefoot (Dysphania botrys), Prickly Lettuce (Lactuca serriola), and Sow Thistle (sonchus oleraceus). In future studies, it is planned to extract the leaves from images using segmentation methods with the same dataset moreover other studies may be conducted on other weeds from different regions.

REFERENCES

[1] Zimdahl, R. L., Weed-crop competition: a review, second edition. USA. Blackwell, 2007.

[2] Knezevic, S. Z., & Datta, A., The critical period for weed control: revisiting data analysis. *Weed Science*, *63*(SP1), 188-202.

[3] Uludag, A., Uremis, I., Tursun, N., & Bukun, B. A, review on critical period for weed control in Turkey. Pp 37 *in The Proceedings of 6th International Weed Science Congress [17-22 June 2012, Hangzhou, China]*.

[4] Uremis, I., Uludag, A., Ulger, A., & Cakir, B., Determination of critical period for weed control in the second crop corn under Mediterranean conditions. *African Journal of Biotechnology*, 2009, *8*(18).

[5] Monaco, T. J., Weller, S. C., & Ashton, F. M., Weed science: principles and practices. John Wiley & Sons, 2002.

[6] Güzel, B., & Okatan, E., Agriculture and AI. Dynamics Changed by AI, 2022, 199-224 (in Turkish).

[7] LeCun, Y., Bengio, Y., & Hinton, G., Deep learning. nature, 521(7553), 2015, 436-444.

[8] Vasileiou, M., Kyrgiakos, L. S., Kleisiari, C., Kleftodimos, G., Vlontzos, G., Belhouchette, H., & Pardalos, P. M., Transforming weed management in sustainable agriculture with artificial intelligence: A systematic literature review towards weed identification and deep learning. Crop Protection, 2024, 176, 106522.

[9] Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., & Vasilakoglou, I., Towards weeds identification assistance through transfer learning. Computers and Electronics in Agriculture, 2020, 171, 105306.

[10] Sunil, G. C., Zhang, Y., Howatt, K., Schumacher, L. G., & Sun, X., Multi-species Weed and Crop Classification Comparison Using Five Different Deep Learning Network Architectures. Journal of the ASABE, 2024.

[11] Pandey, S., Yadav, P. K., Sahu, R., & Pandey, P., Improving Crop Management with Convolutional Neural Networks for Binary and Multiclass Weed Recognition. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, 2024, (IDCIoT) (pp. 539-543). IEEE.

[12] Garibaldi-Márquez, F., Flores, G., & Valentín-Coronado, L. M., Corn/Weed Plants Detection Under Authentic Fields based on Patching Segmentation and Classification Networks. Computación y Sistemas, 28(1), 2024, 271-282.

[13] Trong, V. H., Gwang-hyun, Y., Vu, D. T., & Jin-young, K., Late fusion of multimodal deep neural networks for weeds classification. Computers and Electronics in Agriculture, 2025, 175, 105506.

[14] Peteinatos, G. G., Reichel, P., Karouta, J., Andújar, D., & Gerhards, R., Weed identification in maize, sunflower, and potatoes with the aid of convolutional neural networks. Remote Sensing, 2020, 12(24), 4185..

[15] Chen, D., Lu, Y., Li, Z., & Young, S., Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. Computers and Electronics in Agriculture, 2022, 198, 107091.

[16] Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., & Pieters, J. G., Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. Plant methods, 2020, 16, 1-12.

[17] Ahmad, A., Saraswat, D., Aggarwal, V., Etienne, A., & Hancock, B., Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems. Computers and Electronics in Agriculture, 2021, 184, 106081

[18] Zhang, R., Wang, C., Hu, X., Liu, Y., & Chen, S., Weed location and recognition based on UAV imaging and deep learning. International Journal of Precision Agricultural Aviation, 2020, 3(1).

[19] Sharpe, S. M., Schumann, A. W., & Boyd, N. S., Goosegrass detection in strawberry and tomato using a convolutional neural network. Scientific Reports, 2020, 10(1), 9548.

[20] Tronrio, K., Puerto, A., Pedraza, C., Jamaica, D., & Rodríguez, L., A deep learning approach for weed detection in lettuce crops using multispectral images. AgriEngineering, 2020, 2(3), 471-488.

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

[22] Rai, N., Zhang, Y., Ram, B. G., Schumacher, L., Yellavajjala, R. K., Bajwa, S., & Sun, X., Applications of deep learning in precision weed management: A review. *Computers and Electronics in Agriculture*, 2023, *206*, 107698.

[23] Nioche C, Orlhac F, Buvat I., Texture — User guide, local image features 419 extraction, 2024. Available from: www.lifexsoft.org

[24] Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P., Introduction to machine learning, neural networks, and deep learning. Translational vision science & technology, 2020, 9(2), 14-14.

[25] Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning, MIT Press, 2016.

[26] Huang G, Liu Z, Van Der Maaten L, Weinberger K Q., Densely connected convolutional networks, In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, (4700−4708).

[27] Zhang A, Lipton Z C, Li M, Smola A J., Dive into deep learning, Cambridge University Press.

[28] Vasilev I., Advanced Deep Learning with Python: Design and implement advanced next-generation AI solutions using TensorFlow and PyTorch, Packt Publishing Ltd, 2019.

[29] Chollet F., Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, 1251-1258.

[30] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Rabinovich A., Going deeper with convolutions, In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, 1−9.

[31] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K., Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. cite. arXiv preprint arxiv:1602.07360, 2016.

[32] Tan, M., & Le, Q., 2019, Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, 2019, (pp. 6105-6114). PMLR.

[33] Ravishankar N. Chityala, Kenneth R. Hoffmann, Daniel R. Bednarek, & Stephen, 524 Rudin "Region of interest (ROI) computed tomography", Proc. SPIE 5368, Medical 525 Imaging. Physics of Medical Imaging, 2004, https://doi.org/10.1117/12.534568

[34] Hossain, M. S., Shahriar, G. M., Syeed, M. M., Uddin, M. F., Hasan, M., Shivam, 464 S., & Advani, S., Region of interest (ROI) selection using vision transformer 465 for automatic analysis using whole slide images. Scientific Reports, 2023, 13(1), 11314.

[35] Bianco S, Cadene R, Celona L, Napoletano P., 2018, Benchmark analysis of representative deep neural network architectures. IEEE access, 2018, 6, 64270-64277.

[36] Hasanpour S H, Rouhani M, Fayyaz M, Sabokrou M., Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. arXiv preprint arXiv:1608.06037, 2016.

[37] Mumuni A, Mumuni F., Data augmentation: A comprehensive survey of modern approaches, Array, 2022, 100258.

[38] Rebuffi S A, Gowal S, Calian D A, Stimberg F, Wiles O, Mann T A., Data augmentation can improve robustness, Advances in Neural Information Processing Systems, 2021, 34, 29935−29948.

[39] Kumar, S. (n.d.). Data augmentation increases accuracy of your model, but how? Medium. Retrieved December 23, 2024, from https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722

[40] Talukder, M. A., Layek, M. A., Kazi, M., Uddin, M. A., & Aryal, S., Empowering covid-19 detection: Optimizing performance through fine-tuned efficientnet deep learning architecture. Computers in Biology and Medicine, 2024, 168, 107789.