**Plant Seedlings Classification Using Deep Learning**

**Abstract**- Agriculture is crucial to human survival and continues to be a major economic force worldwide, particularly in undeveloped and developing nations. There is a need to enhance plant production while lowering costs because of the growing global population and the difficulties caused by climate change. Weed control plays a crucial role in this scenario. Identifying weeds is one of the most crucial tasks after a few days of plant germination because it allows farmers to implement early-stage weed control to decrease negative effects on crop growth. Classification of seedlings is important in the agriculture and botany fields. This study uses a dataset of about 4750 pictures of 12 species (9 weeds and 3 crops) in various developmental stages to offer an approach for categorising plant seedlings. Several classification techniques are proposed in this project to forecast the plant sort outcome from the test data, which include CNN (convolutional neural networks), ResNet50 (residual networks), and Xception models. From the comparison of the models, Xception with a 300x300 pixel size generated the best results with an average F1 score of 97% and an accuracy score of 97%, which demonstrates that this deep learning model can accurately categorise 12 different plant seedlings in their early growth phases and that farmers may find this tool beneficial in weed identification.

Keywords: Resnet50, Convolutional neural network, Xception, and Plant seedling classification.

I.INTRODUCTION

Agriculture is a major industry in numerous nations worldwide. Many people have chosen agriculture as their profession in order to support themselves and the world's food supply. The demand for food rises annually in direct proportion to population growth. However, farmers encounter numerous difficulties when trying to boost productivity. One of the major challenges is controlling weed plants.

The production of agricultural crops is significantly harmed by weed plant species. These plants are capable of capturing scarce natural resources like light, water, soil nutrients, and space quickly. Due to traits like deep roots, drought and frost resilience, and high nutrient usage efficiency, they can reproduce more quickly than cultivated plants. Also, weeds have the ability to produce compounds known as allelopathics in the soil, which can help pests and crop diseases grow. The quantity of loss can vary depending on the crop type. Therefore, it is essential to find and remove weeds from fields as soon as possible. Herbicides can be used to control weeds. But it also creates a new issue. Costs go up, and the environment suffers greatly. Also, a lot of labour is required. One of the biggest problems is when herbicides aren't available. Furthermore, farmers get into issues when it comes time to trim the grass if they let the weeds grow unchecked.

Plants include weeds as well. Since they are so similar to the plants in their early stages, untrained eyes cannot differentiate them apart. An effective solution to nearly all issues is to automatically detect weeds in their early stages. In order to classify and identify weeds and crops quickly, accurately, cost-effectively, and safely, deep learning techniques must be used.

Plant seedling classification is implemented in this work using the self-built CNN model, ResNet-50 as a feature extractor, a finely tuned pre-trained ResNet-50 model, and Xception as a feature extractor model, and the performance of each model is evaluated in the classification of plant seedlings. The project’s main goal is to discover the optimal architecture that can distinguish between weeds and plants precisely enough to develop an automated system that could be applied in the field.

II. LITERATURE REVIEW

The authors of [1] developed a neural network that is trained from scratch. The resized 224 x 224 images are fed to the network. The network contained seven layers, out of which five are convolutional layers and two are fully connected layers. The performance of classifying seedling images from 12 different species is investigated. The model achieved an accuracy rate of 90.15%.

In order to recognise different plant species in coloured photos, the authors of [2] created a deep convolutional neural network from the start. The network was fed with 128x128 RGB images. The network contained a convolutional layer, a max pooling layer, and two residual blocks, each with two convolutional layers and one max pooling layer. ReLU and batch normalisation were connected to each convolutional layer, respectively. Two categorization layers were the network's final component. The network was trained and tested on 10,413 photos of 22 different types of crops and weeds. An accuracy of 86.2% was attained by the network in classification.

The authors of [3] used a sizable, 17,509 image mosaic dataset that was made available to the public. Images were gathered from various northern Australian regions and divided into eight types. Higher accuracy for ResNet-50 was achieved after implementing CNN architectures.

Authors in [2] implemented techniques using binary image

conversion and SVM to differentiate between crop and weed

and obtained an accuracy of around 50%.

In order to distinguish between crops and weeds, the authors in [4] created methods employing binary image conversion and SVM and achieved an accuracy of about 50%.

Resnet, VGG16, and Inception V3 deep learning architectures were employed by the authors of [5] to detect banana diseases. Resnet-152 has achieved high accuracy for the given dataset.

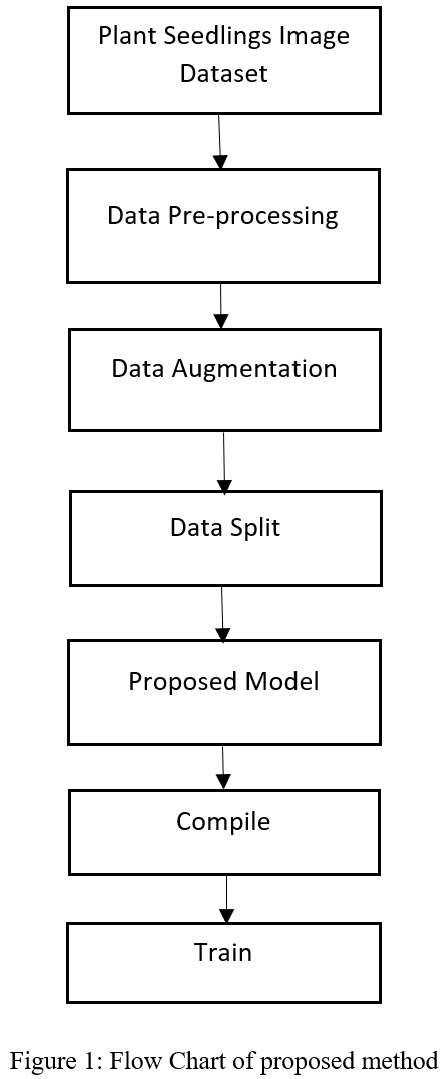
Authors in [4] used deep learning

architectures, namely Resnet, VGG16, and InceptionV3 for

banana disease detection.

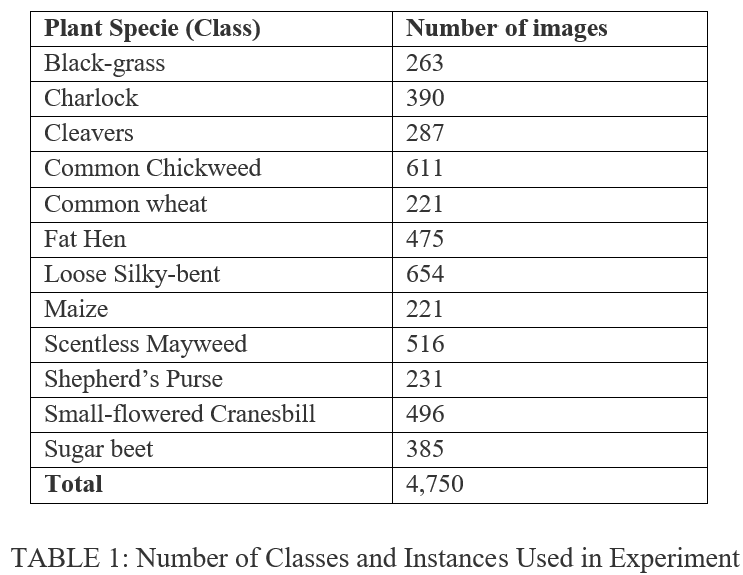
III METHODOLOGY

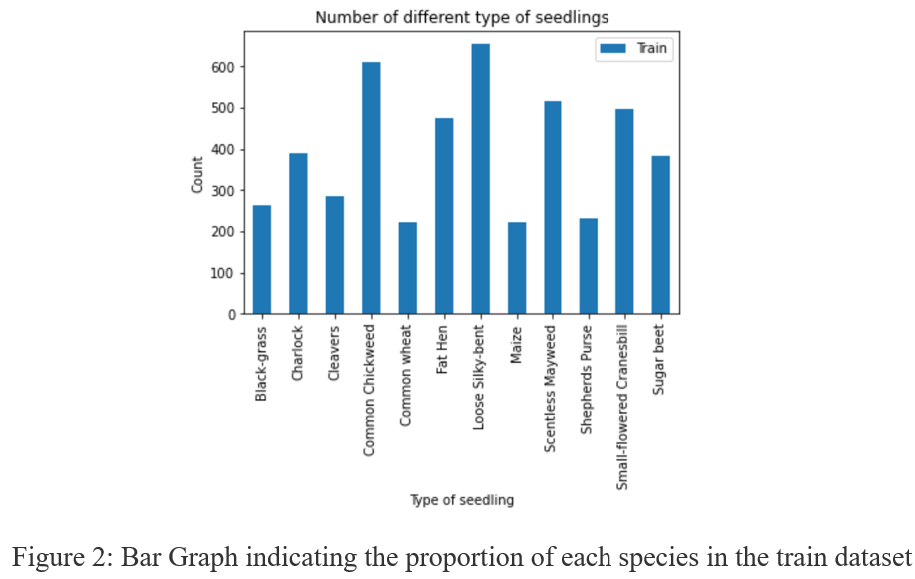
This section outlines our method from dataset collection to plant seedling image classification using deep learning. The flow of our proposed approach is depicted in Fig. 1.

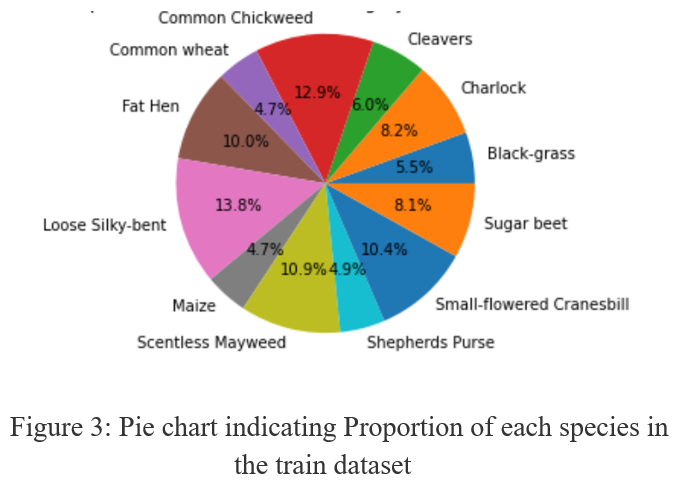


1. Dataset:

To evaluate the effectiveness of the architectures investigated in this study, a publicly available dataset from Kaggle is used, which has approximately 5544 photos, of which 794 images represent the test set and 4750 images represent the training set. The images depict distinctive plants from 12 different plant species. Table 1 lists the 12 species of plant seedlings. These are Charlock, Black-grass, Cleavers, Common wheat, Common Chickweed, Fat Hen, Maize, Loose Silky-bent, Scentless Mayweed, Small-flowered Cranesbill, Shepherds Purse and Sugar beet.







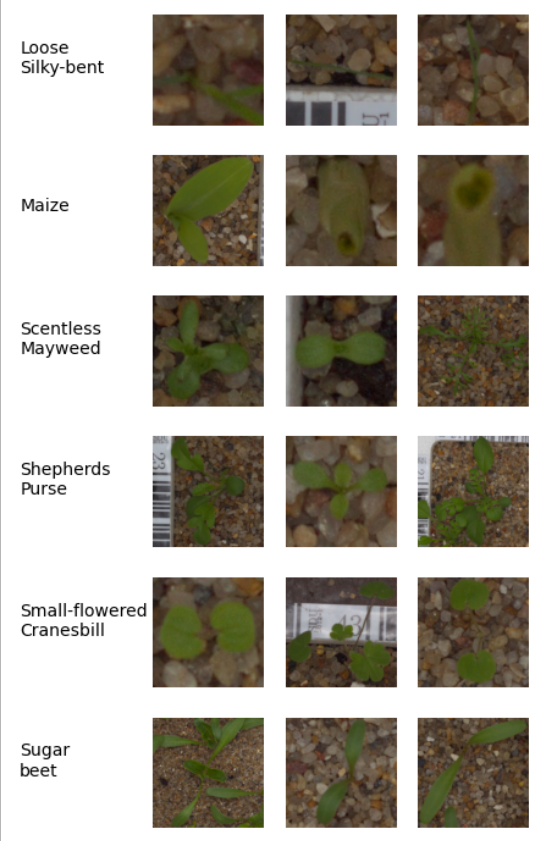
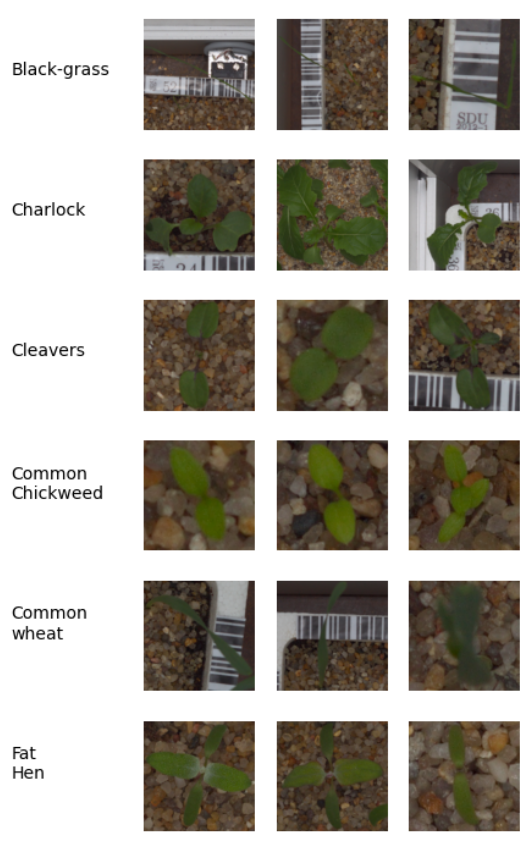


Figure 4: Some of the plant seedling images.

1. Image Pre-processing

The photos are resized to 224 x 224 pixels and 300 x 300 pixels. Pixel values are normalised as per the model, which gives a better training result.

1. Data Augmentation:

A data augmentation technique is performed to reduce over-fitting during training. This procedure is very helpful when dealing with an unbalanced dataset. The augmentation methods applied are: rotation by 180 degrees; zooming in a range of 0–30%; width-shift and height-shift in a range of 0–30%; horizontal and vertical flips.

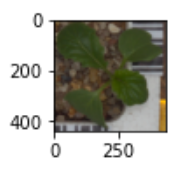


Figure 5: Actual image of Charlock

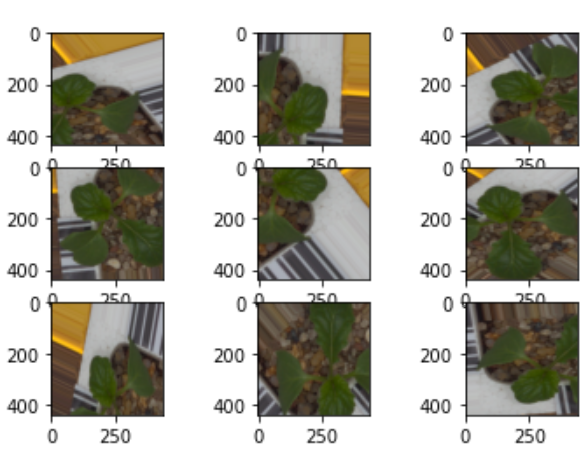


Figure 6: Some of the images generated after performing augmentation on figure 5

1. Data Split:

The training set and test set are already present in the dataset. The training set is divided into a 9:1 ratio, where 90% is used for training and 10% for validation. After the data is split, the training set, validation set, and test set contain 4279, 471, and 794 images, respectively, belonging to 12 classes.

1. Model Building

In this project, several models (classifiers) are employed: a self-built CNN model, a ResNet-50 model, and an Xception model.

1. Convolutional Neural Network:

Convolutional Neural Network (CNN) works better with images since it can discover features by itself. A CNN employs 2D convolutional layers and integrates input data with learned characteristics, making it an excellent architecture for processing 2D data, such as photos.

A convolutional neural network consists of an input layer, hidden layers, and an output layer.

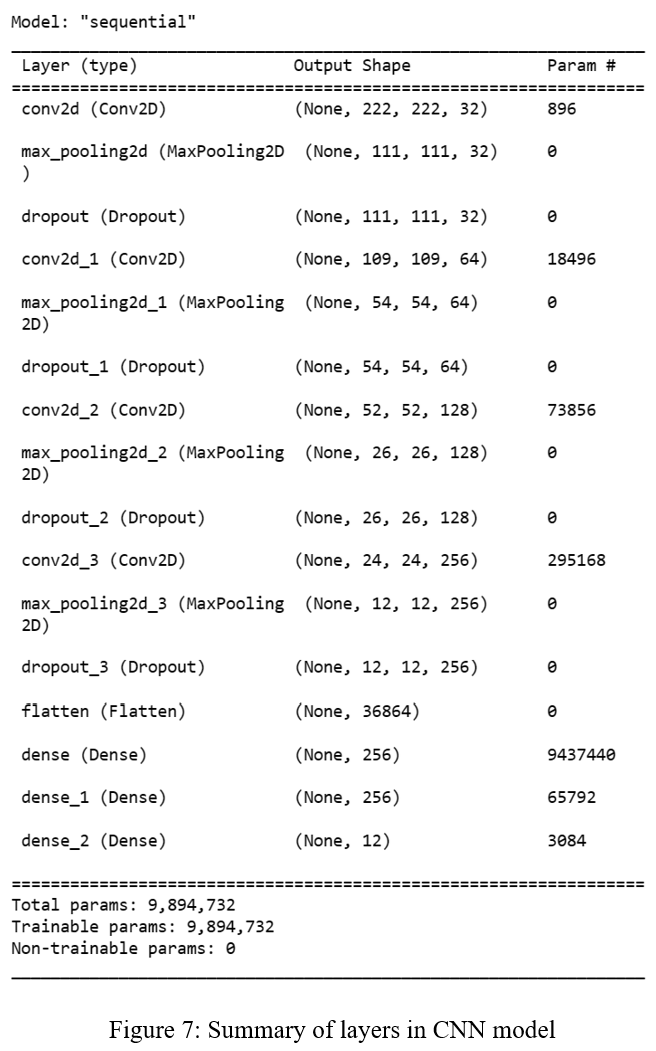


Figure 7 depicts the proposed model, which includes four convolutional layers along with max pooling and dropout layers, followed by three dense layers. The layers that were used to build the CNN model are listed below.

• CONV layer

The CONV layer calculates the output of neurons that are linked to local regions in the input, with each neuron computing a dot product between their weights and the small region they are linked to in the input volume.

• POOL layer

The pool layer produces a smaller volume than the preceding layer by downsampling along the spatial dimensions (width and height). These are employed to lessen computational expense and reduce overfitting to some extent.

• Dense layer

In a dense layer, each neuron is connected to every other neuron in the preceding layer, so it is also called a "fully connected layer." Each node of the dense layer generates a score value that corresponds to a class score value.

• Dropout layer

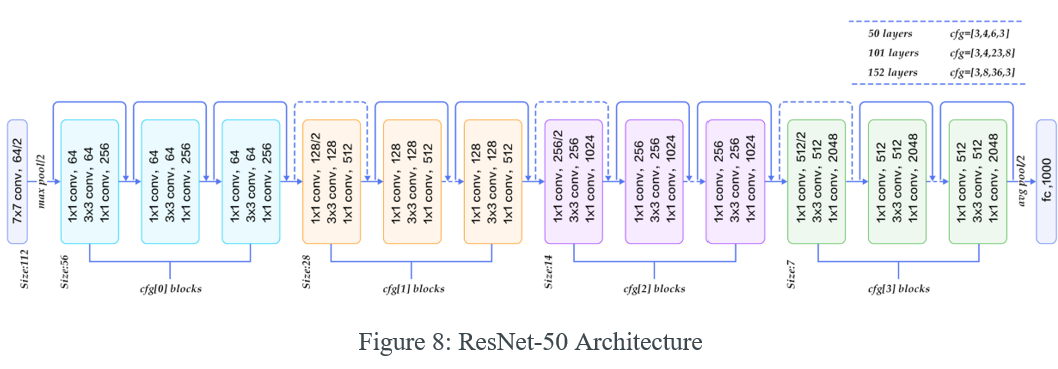
Dropout layer is used as a technique for regularising the training set's overfitting. It makes CNN less complex after each iteration by "dropping" neurons at random (by setting weights to zero), which also makes it difficult for the model to overfit.

Transfer Learning:

The phrase "transfer learning" refers to applying a model that has already been trained to a different problem. Transfer learning is the process through which a computer enhances its prediction of a new task using the knowledge it has learned from a previous task.

1. ResNet50

The residual network is known as ResNet. ResNet has numerous variations that use the same idea but have various numbers of layers. Resnet50 is the name given to the version that supports 50 neural network layers. It has forty-eight convolution layers, one average pool layer, and one MaxPool layer. The floating-point operation count is 3.8 x 109. It’s a common ResNet model.



1. **ResNet-50 as feature extractor:**

In this technique, all the convolutional blocks are frozen, and just the fully connected layers are trained on the fresh dataset. In practise, this equates to employing a specific layer or layers of the pre-trained ResNet-50 as a fixed feature extractor.

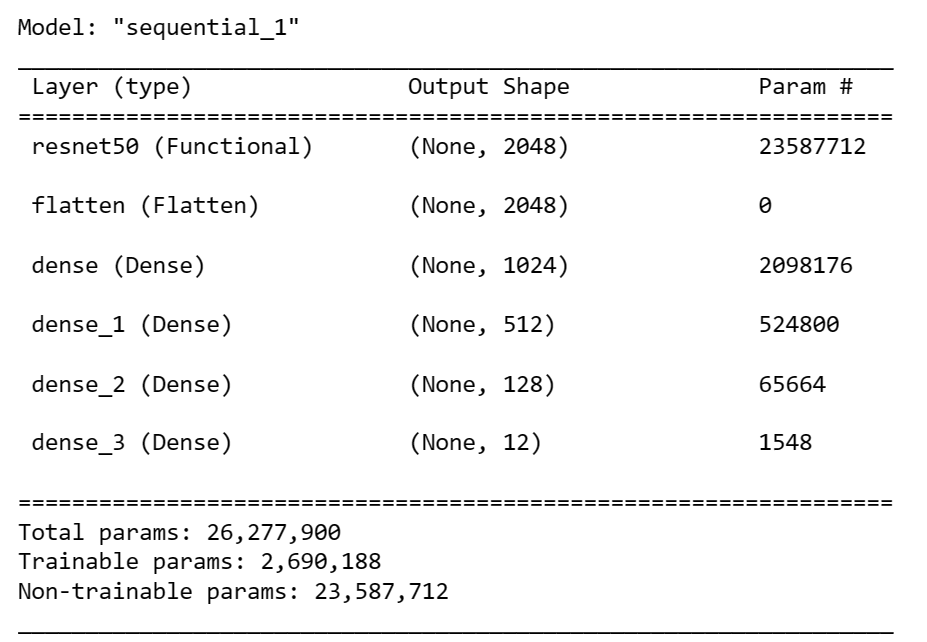
****

Figure 9: Summary of layers in Resnet50 as feature extractor model.

As shown in Figure 2, ResNet50 is used to train the model. The pre-trained networks which are pre-loaded with weights that are trained on a large dataset is used as base model.

The pre-trained model's fully linked layers are replaced to fit the particular categorization issue. The fully connected classifier included four layers, each having 1024, 512, and 128 nodes with ReLU activation function, and a layer containing 12 nodes with Softmax activation function.

1. Fine-tuning the ResNet50

Other typical transfer learning strategy entails not only retraining the classifier on top of the network with the new dataset, but also fine-tuning the network by simply training the higher-level segment of the convolutional layers and continuing backpropagation.

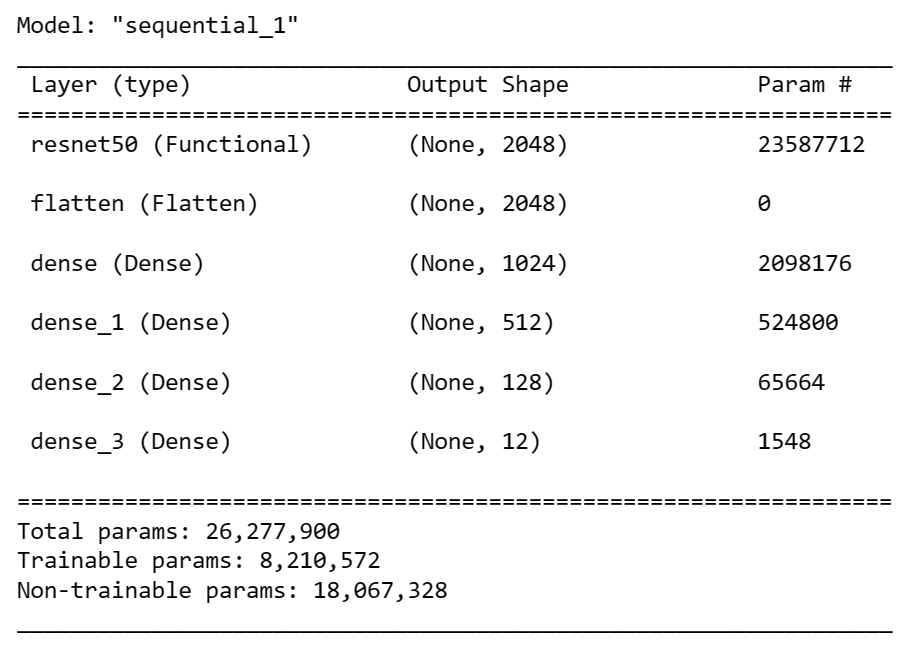


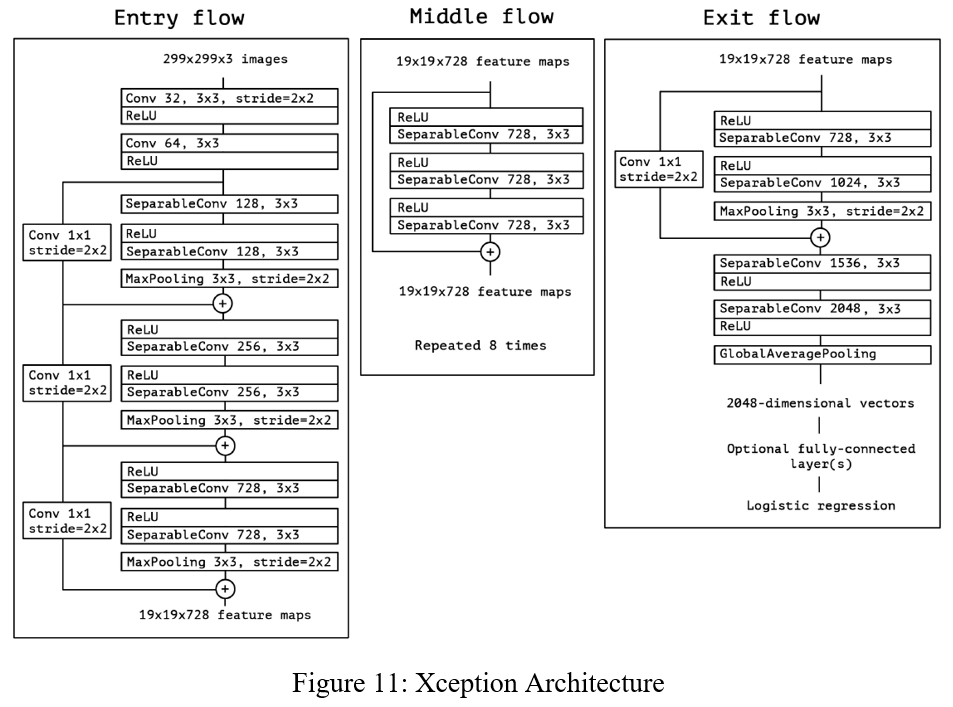
Figure 10: Summary of layers in ResNet-50 finely tuned model.

In this work, lower-level layers of the network are frozen since they contain more generic features of the dataset. Only the top layers of the network are trained due to their ability to perform the extraction of more specific features. As shown in Figure 1, only the last 4 layers are trainable, which are the dense layers having 1024, 512, and 128 units with activation function ReLU and a layer having 12 units with activation function ReLU.

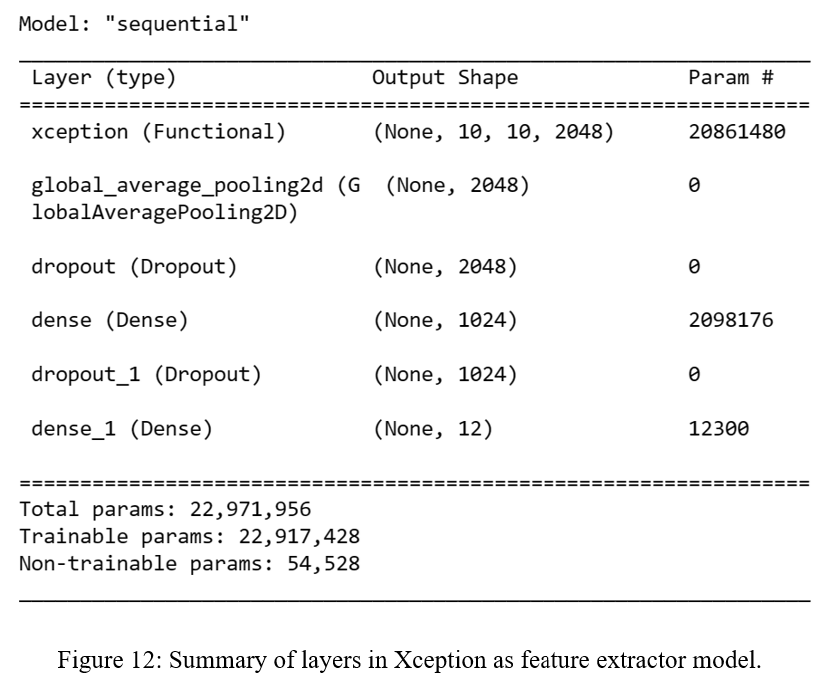
1. Xception

Xception, short for "extreme inception," is a concept that takes the principles of Inception to an extreme. In Inception, the initial input was compressed using 1x1 convolutions, and the depth space was then created from each of the input spaces using various types of filters. This step is simply reversed by Xception. Instead, it first filters each depth map individually before using 1X1 convolution to finally condense the input space across the depth. A depthwise separable convolution, an operation that was employed in neural network creation as early as 2014, is almost the same as this technique.

There is yet another difference between Inception and Xception. whether or not there is a non-linearity after the original procedure. While Xception doesn't introduce any non-linearity, the Inception model has a ReLU non-linearity that follows both procedures.



1. **Xception as feature extractor:**

****

The proposed Xception model's design is depicted in Figure 12. The pre-trained networks that have been preloaded with weights and trained on a sizable dataset are used to build the basic model. The pre-trained model is topped with a new classification layer to ensure that the final layer corresponds to the classification task. To do this, the pre-trained model's final layer is removed, and its place is taken by a global average pooling layer, which creates a feature map that is then fed into the fully connected classifier. The fully connected layer was altered in such a way to fit the specific classification problem; two layers containing 1024 units, 12 units with relu and softmax activation functions, respectively, and some dropouts in the middle are included in the fully connected classifier.

**Compiling and Training the model**

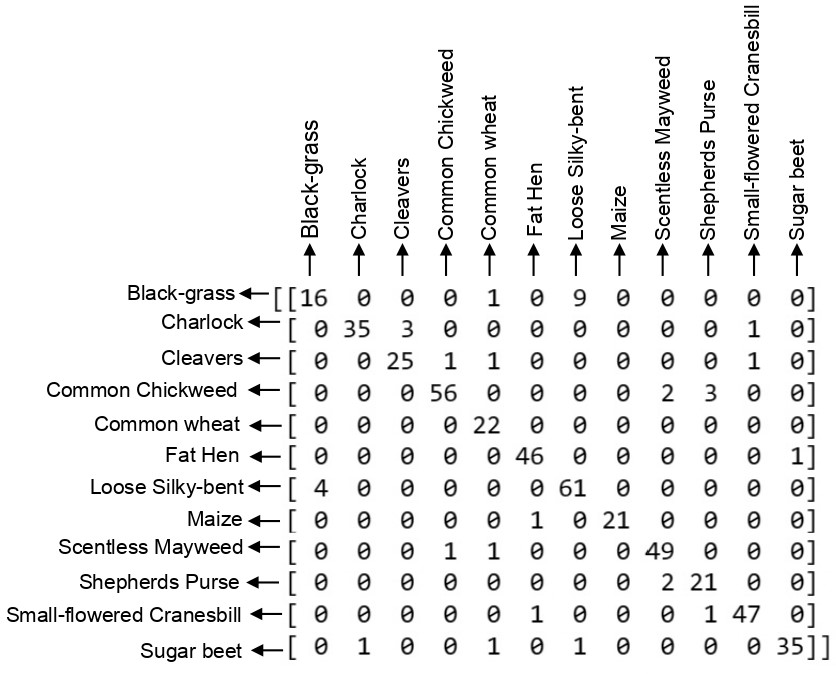
After building the model, it is executed using the training and validation datasets. The models are trained for 100 epochs using an early stopping method to monitor the validation accuracy and avoid overtraining. The Adam optimizer and categorical cross-entropy loss function are used, along with a batch size of 16. After compilation and training, the models are evaluated.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

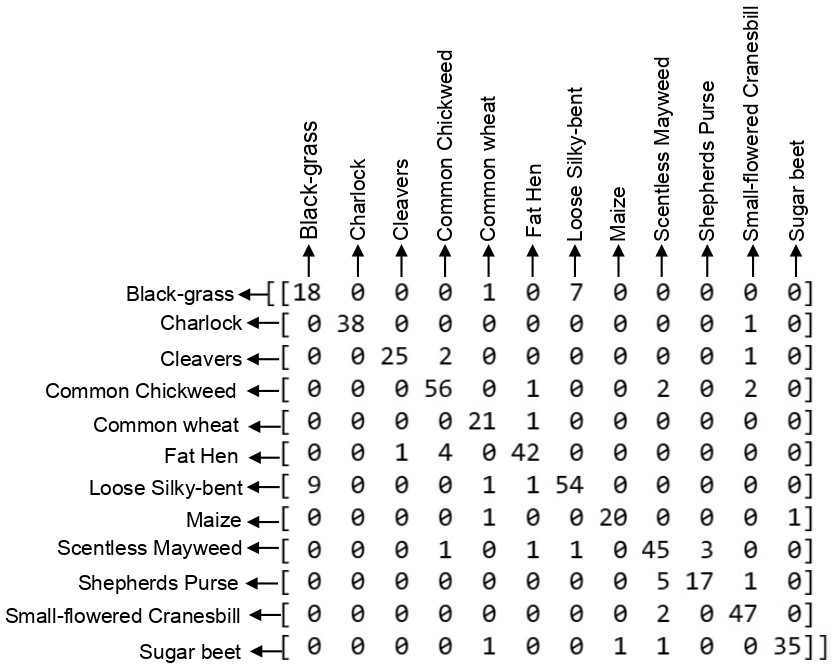
The performance of each of the four proposed CNN architectures for the classification of plant seedlings is evaluated and recorded. The metrics used to evaluate the model are the precision, confusion matrix, F1-score, recall, and accuracy.

Confusion Matrix: A confusion matrix denoted by C is a n\*n matrix where n denotes the number of classes, such that each entry in the ith row (which indicates the actual class) and the jth column (which indicates the predicted class) denoted by Ci,j is equal to the number of input observations originally of class i but predicted as class j.

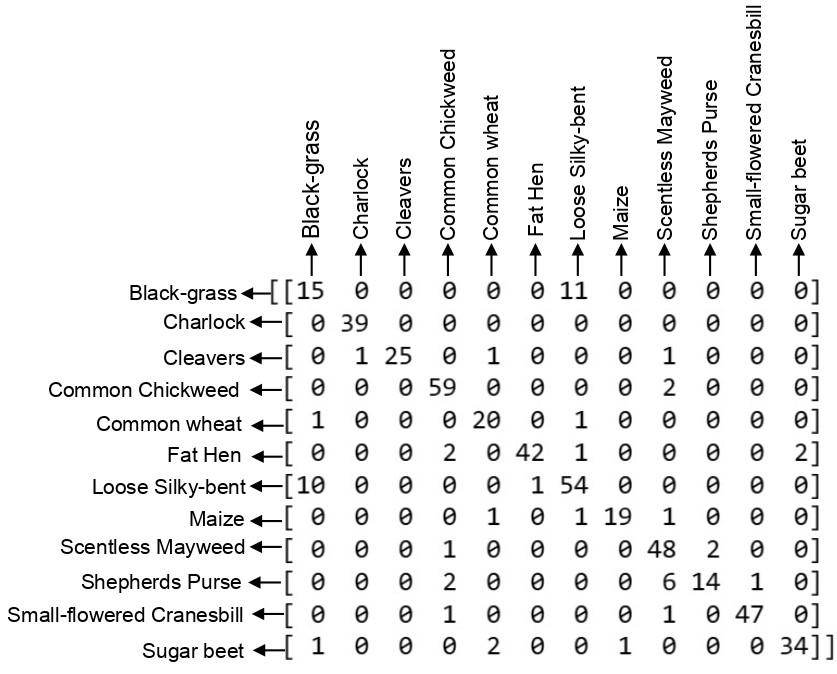
For our classification problem, the confusion matrix has six rows and six columns.



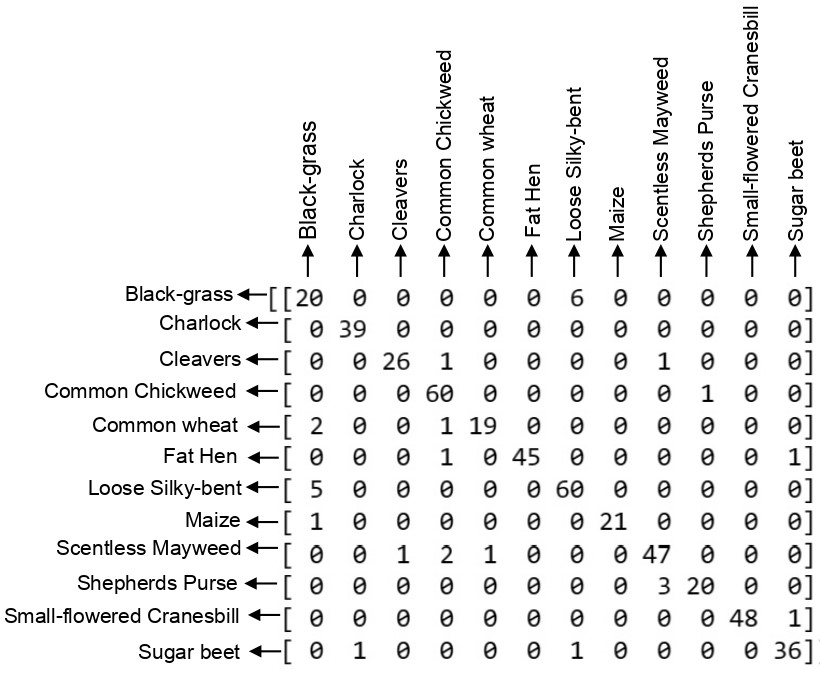
(i) Confusion Matrix of self-built CNN model with 224x224 pixel size



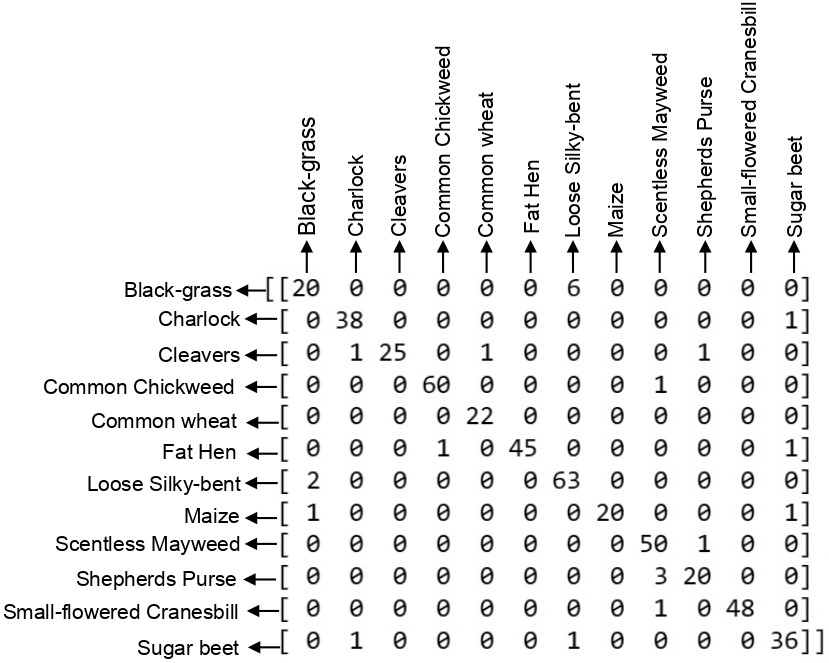
(ii) Confusion Matrix of ResNet-50 with 224x224 pixel size



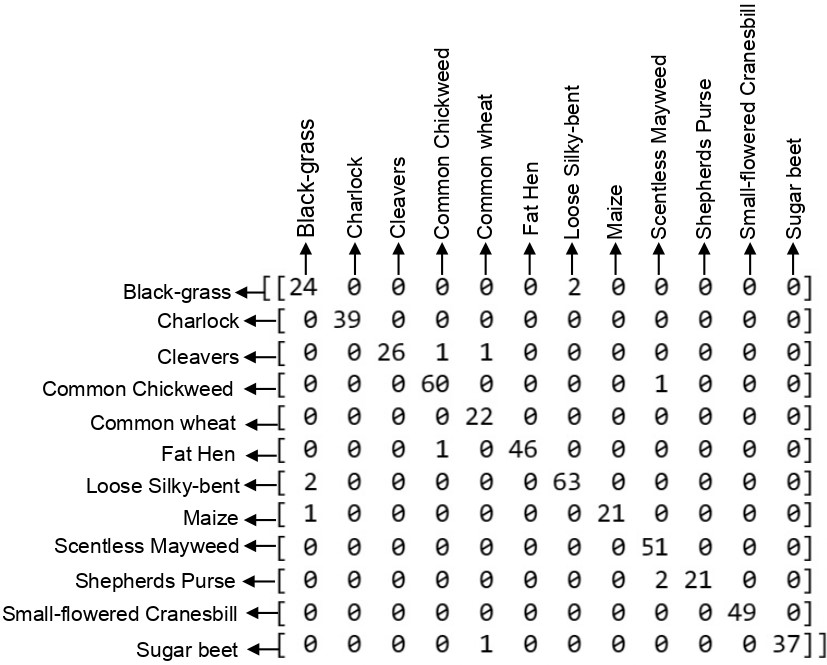
(ii) Confusion Matrix of Resnet-50 with 300x300 pixel size



(iii) Confusion Matrix of finely tuned ResNet-50 (Last 15 layers trainable)+ three dense layers with 224x224 pixel size



(iv) Confusion Matrix of finely tuned ResNet-50 (Last 15 layers trainable) + three dense layers with 300x300 pixel size



(v) Confusion Matrix of Xception with 300x300 pixel size

Figure 13: Confusion matrices of different proposed models calculated on validation dataset

From the confusion matrices of the self-built CNN and ResNet-50 models, we can see that black-grass is more often misclassified as silky-bent and silky-bent as black-grass. Some of the Shepherd’s Purse were misclassified as scentless mayweed, and there were very small misclassifications among other classes. Most of these misclassifications are removed by the Xception model.

Precision = (TP / (TP+FP)) (1)

Recall = (TP / (TP+FN)) (2)

F1-score = (2 x ((Precision x Recall) / (Precision+Recall))) (3)

Accuracy = ((TP+TN)/(TP+TN+FP+FN)) (4)

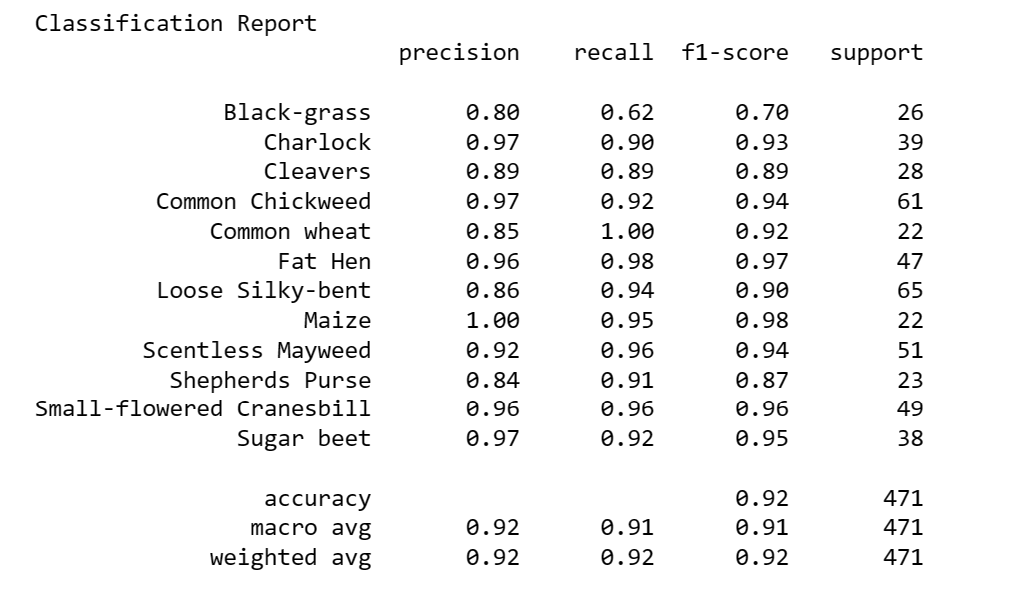
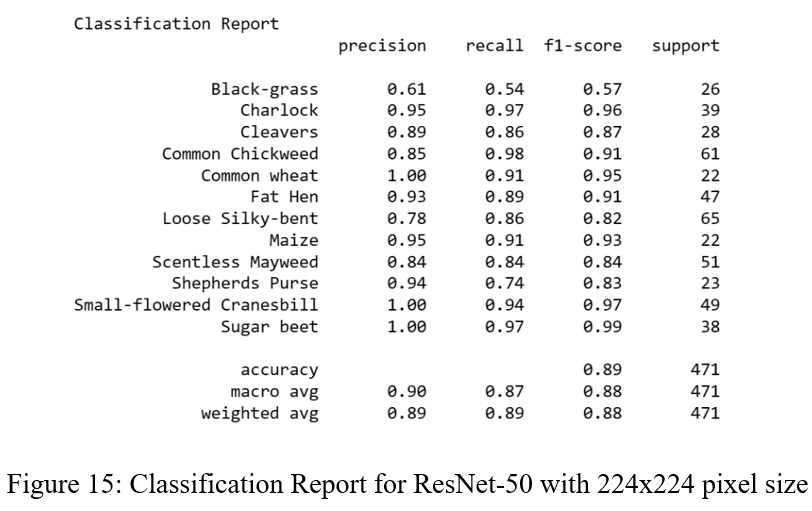
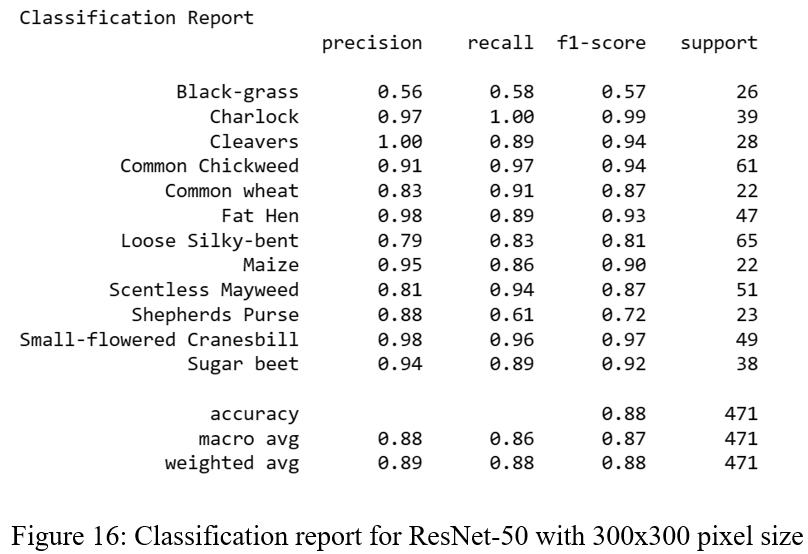


Figure 14: Classification Report for self-built CNN model with 224x224 pixel size





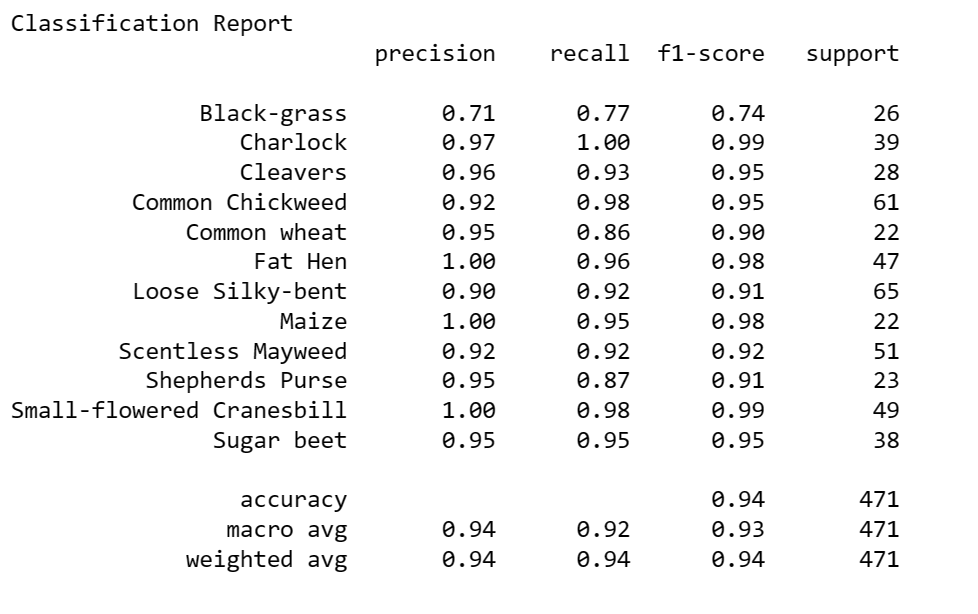


Figure 17: Classification Report for Finely tuned ResNet-50(Last 15 layers trainable)+ three dense layers with 224x224 pixel size

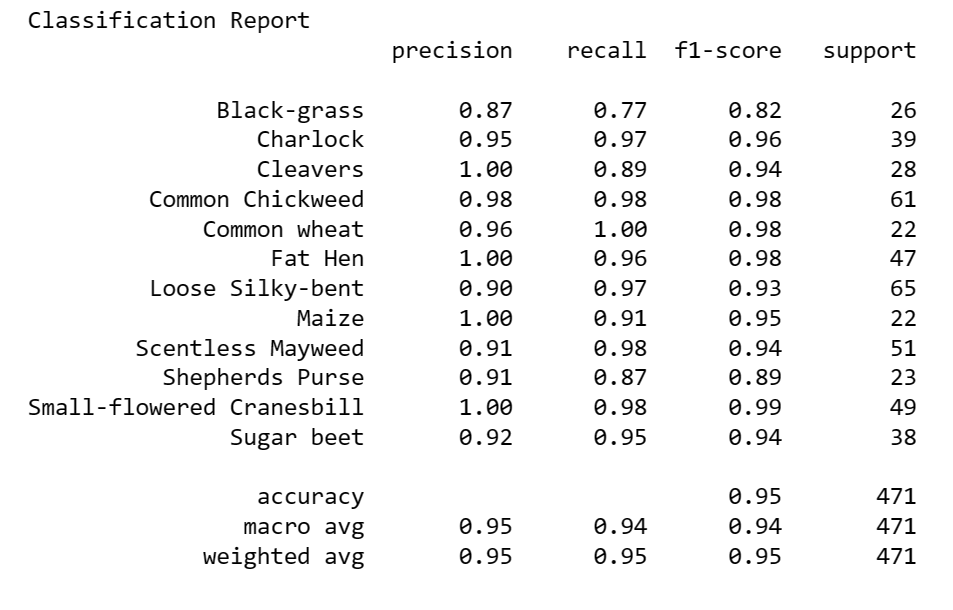
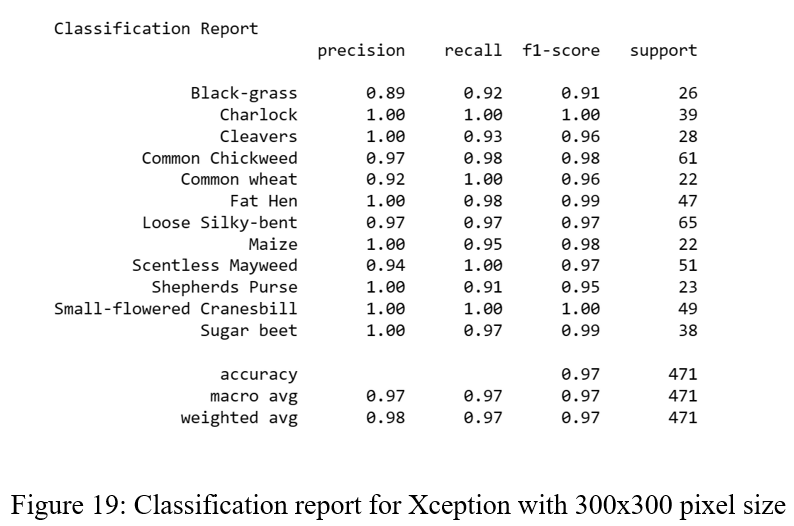


Figure 18: Classification Report for Finely tuned ResNet-50(Last 15 layers trainable) + three dense layers with 300x300 pixel size



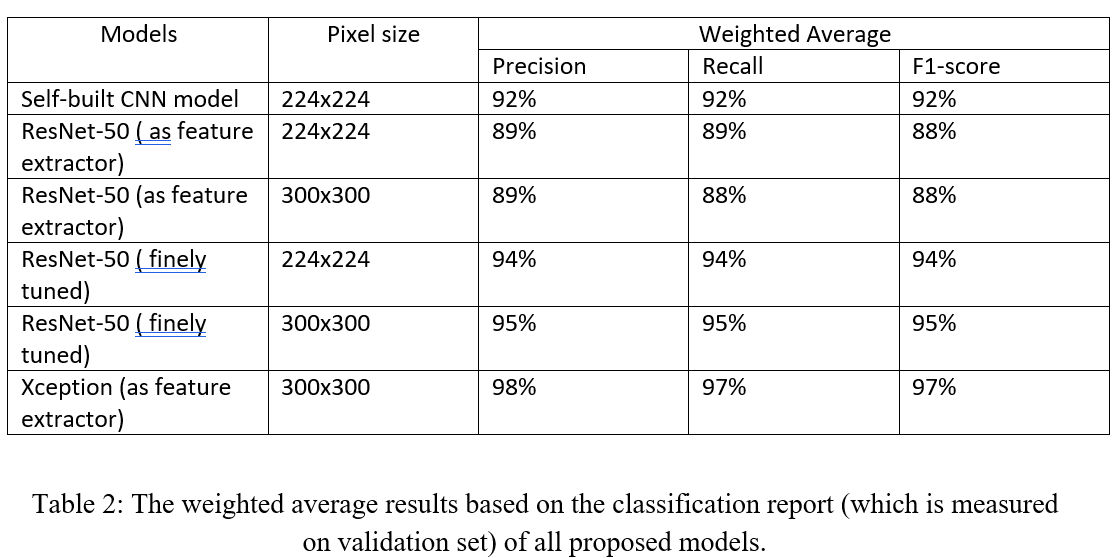


Table 2 displays the averages for all classes. According to the results, the Xception with a 300x300 input pixel size produces the best results with a precision of 98%, a recall of 97%, and an F1-score of 97%. According to the Xception Classification report, Charlock, Common Wheat, Small-flowered Cranesbill, and Scentless Mayweed have the highest recall values. And the Small-flowered Cranesbill and Charlock both received the highest F1-score. The calculated Accuracy values based on eq (4) is shown in table 3

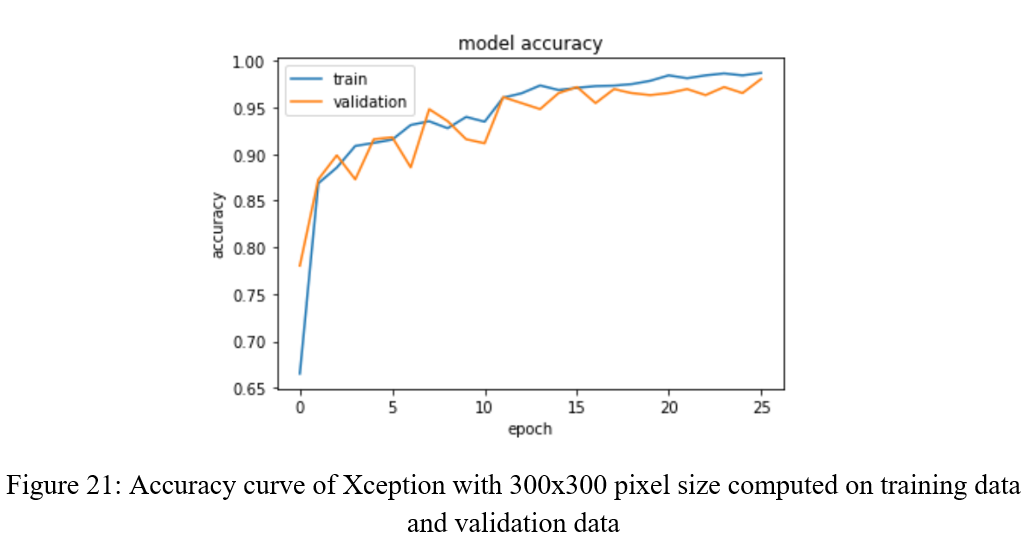
|  |  |  |
| --- | --- | --- |
| Model | Pixel size | Accuracy |
| Self-built CNN model | 224x224 | 92% |
| ResNet-50 ( as feature extractor) | 224x224 | 89% |
| ResNet-50 (as feature extractor) | 300x300 | 88% |
| ResNet-50 ( finely tuned) | 224x224 | 94% |
| ResNet-50 ( finely tuned) | 300x300 | 95% |
| Xception (as feature extractor) | 300x300 | 97% |

Table 3: Accuracy of all proposed models

The accuracy values for the various models utilising various pixel sizes are shown in Table 3. In comparison to other models, Xception produces the best results when employed as a feature extractor. With an input size of 224x224, the accuracy of the CNN model is 92%. ResNet50 obtains 89% accuracy when used as a feature extractor with 224x224 pixels, but decreases to 88% accuracy with 300x300 pixels. At 224x224 pixels, a precisely adjusted ResNet-50 model achieves 94% accuracy, while at 300x300 pixels, it reaches 95% accuracy. With a 97% accuracy rating, Xception performs best overall for recognising plant seedlings while using a 300x300 pixel size.

In order to have a deeper understanding of how well Xception performs, we computed its loss curve, which is displayed in Fig. 20. The validation loss is seen to decrease as the number of epochs rises, and overfitting is not observed.



From Fig. 21, it is observed that accuracy grows along with the number of epochs, and an overfitting problem is not observed.

V. CONCLUSION

Four CNN architectures are used in this work to accurately distinguish plants and weeds: self-built CNN, ResNet-50 as a feature extractor, finely tuned ResNet-50, and Xception as feature extractor models.

First, a self-built CNN model was employed for categorization, as demonstrated above, but it performed poorly. The ResNet model is then utilised to increase the accuracy; for 224x224 and 300x300 pixel sizes, respectively, the finely tuned Resnet-50 provided a better accuracy of 94% and 95%. Additionally, the Xception model is used to improve accuracy. The Xception model yields the highest accuracy. Farmers could use this model to automatically classify weeds and seedlings.

REFERENCES

[1] C. R. Alimboyong and A. A. Hernandez, "An Improved Deep Neural Network for Classification of Plant Seedling Images," 2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA), 2019, pp. 217-222, doi: 10.1109/CSPA.2019.8696009.

[2] M. Dyrmann, H. Karstoft, and H. Midtiby, "Plant species classification using deep convolutional neural network," Biosystems Engineering, 2016, pp. 72-80.

[3] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny,J. Whinney et al., “Deepweeds: A multiclass weed species image dataset for deep learning,” Scientific reports, vol. 9, no. 1, pp. 1–12, 2019.

H. A. Elnemr, "Convolutional Neural Network Architecture for Plant

Seedling Classification," International Journal of Advanced Computer

Science and Applications, vol. 10, no. 8, pp. 319–325, 2019,

https://doi.org/10.14569/IJACSA.2019.0100841.

H. A. Elnemr, "Convolutional Neural Network Architecture for Plant

Seedling Classification," International Journal of Advanced Computer

Science and Applications, vol. 10, no. 8, pp. 319–325, 2019,

https://doi.org/10.14569/IJACSA.2019.0100841.

H. A. Elnemr, "Convolutional Neural Network Architecture for Plant

Seedling Classification," International Journal of Advanced Computer

Science and Applications, vol. 10, no. 8, pp. 319–325, 2019,

https://doi.org/10.14569/IJACSA.2019.0100841.

[4] S. Murawwat, A. Qureshi, S. Ahmad, and Y. Shahid, “Weed Detection Using SVMs”, Eng. Technol. Appl. Sci. Res., vol. 8, no. 1, pp. 2412–2416, Feb. 2018.

[5] S. L. Sanga, D. Machuve, and K. Jomanga, “Mobile-based Deep Learning Models for Banana Disease Detection”, Eng. Technol. Appl. Sci. Res., vol. 10, no. 3, pp. 5674–5677, Jun. 2020.

[6] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” pp. 1–14, 2014.

[7] N. Khoza, M. Khosa, T. Mahlangu and N. Ndlovu, "Plant Seedling Classification Using Machine Learning," 2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), 2022, pp. 1-6, doi: 10.1109/icABCD54961.2022.9856067.