PROJECT 3: IMAGE CLASSIFICATION - REPORT

APRIL 20, 2023

UNM Spring 2023

Course:

CS429 - Introduction to Machine Learning

Instructor:

Trilce Estrada

Teacher Assistant:

Adnan Bashir

Group Members:

Libra Vento

Sarathi TS

CONTENTS

I INTRODUCTION

Image classification is a supervised learning problem in which a model, using labeled example photos, learns to recognize to what target class an image belongs to.

The objective of this project is to train a model that receives a photo of a plant as input and returns the name of the species it belongs to as an output. The species this classifier considers are:

- Black-grass
- Charlock
- Cleavers
- Common Chickweed
- Common wheat
- Fat Hen
- Loose Silky-bent
- Maize
- Scentless Mayweed
- Shepherds Purse
- Small-flowered Cranesbill
- Sugar beet

Different CNN architectures as well as hyperparameter settings are compared with each other to find the optimal way to train the model.

The following libraries are used in this project:

- **os** to load data.
- **cv2** to read the images.
- **numpy** to manipulate the data.
- **matplotlib** to visualize graphs.
- **random** to get the randomized samples.
- **sklearn** to obtain the One Hot Encoder and Train/Test split function.
- **keras** to obtain the functions the layers and pretrained models, among other similar neural network related functions.

II DATA

II-A Exploration

The data to be used is the one from the Plant Seedlings database, which contains annotated RBG images have a ”physical resolution of roughly 10 pixels per mm.” of ”960 unique plants belonging to 12 species at several growth stages” [1]. This database was produced as part of a study that aimed to provide a freely available database for agricultural processes [2].

Some samples for each species in the dataset were extracted and a graph with the number of training images sorted by species was produced.

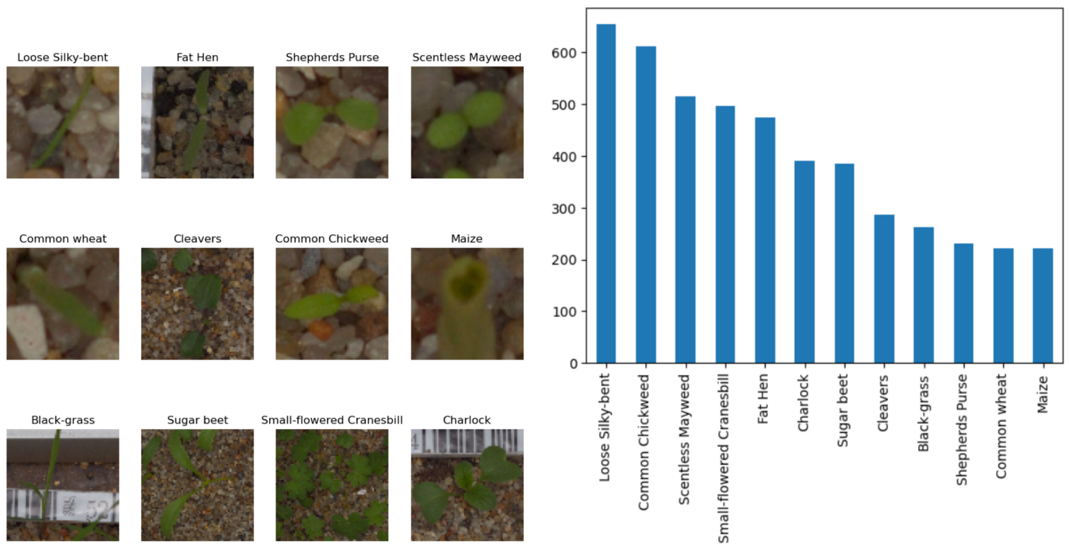


Fig. 1: Dataset images examples

It is evident that there is an imbalance between the number of samples per category, with the Loose Silky-bent having over 600, and both the Common wheat and Maize having the least amount of samples.

II-B Preprocessing

In order to avoid confusion when it came to identifying the important features in the images, preprocessing was necessary. This task consists of blurring, converting from RGB to HSV and adding a mask.

A gaussian blur is first applied over the image in order to reduce noise and speckles in the image, as well as removing ambiguity between different segments of an image. In other words, it reduces the possibility of running into false edges, thus making it easier to recognize where the object to be identified is.

After that, the color model of the image is converted from RGB (Red, Green, Blue) to HSV (Hue, Saturation, Value). Since the objects the model is to classify all have the same color, this information is not as relevant to make a distinction between categories and instead it is better to focus on other features, specially luminance, and separate these from the color information. Plus, it becomes easier for cv2 to read. The use of HSV helps with this.

With both of these aspects nailed down, a mask to hide the background can now be applied. This allows only the important part of the images to be visible and focus solely on them. An example of the final results is shown below.

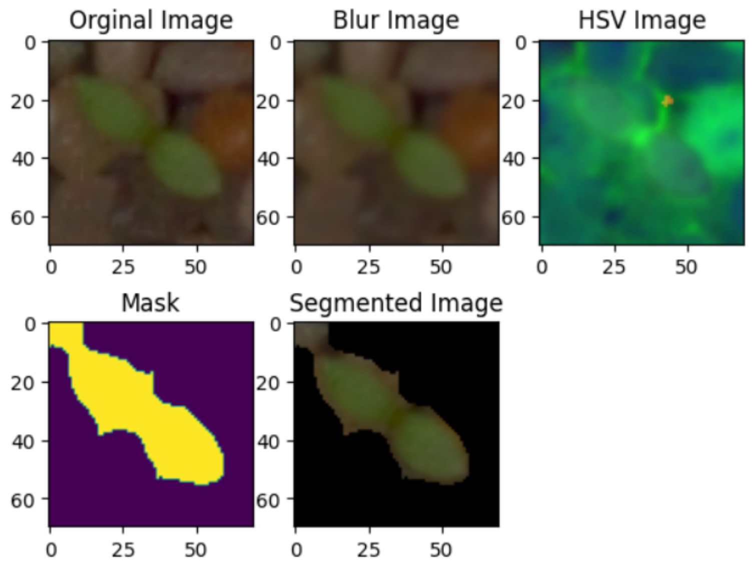


Fig. 2: Preprocessing sequence example

Once this was done, the dataset was split in training and validation, with the validation data being 10% of the original.

Given the results of the exploration, data augmentation was applied to balance out the distribution of the samples according to their class. This was done by applying random transformations over them, including rotation, zooming in or out and shifting and reflecting the images horizontally and vertically.

Regarding classes, these were converted into the following one-hot labels with 0 ”the species was not found” and 1 ”the species was found”.

III MODEL

III-A Theoretical Framework

Because of its nature, image classification requires a classifier to be solved, but due to the input it receives, it requires a more sophisticated approach than classifying text does. The most basic way for images to be analyzed is to use pixel data. **Convolutional Neural Networks (CNNs)** are often used due to their ability to reduce dimensionality without losing information.

CNNs are ”neural networks that use convolution in place of general matrix multiplication in at least one of their layers” [3]. The convolution operation consists of obtaining an smoothed estimate of the effects the shape a function has over another one. It receives an input function, works with a kernel during processing and outputs a ”feature map”. The kernel works as a ”filter” the input function goes through multiple times, with each iteration focusing on a different section of the input layer. A map is produced from the resulting activations of its application, that highlights the most important features in the network. Thus, the model is able to learn which are the optimal values for the filter matrix to extract the most meaningful features. [4] It is expressed as

$$s(t) = (x * w)(t)$$

with x: input, w: kernel and s feature map. prominent features. [3]

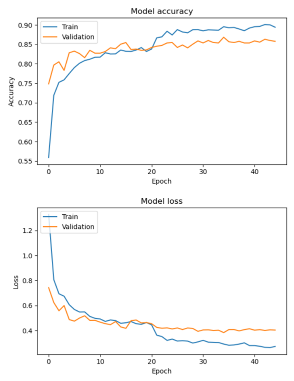
III-B Selection

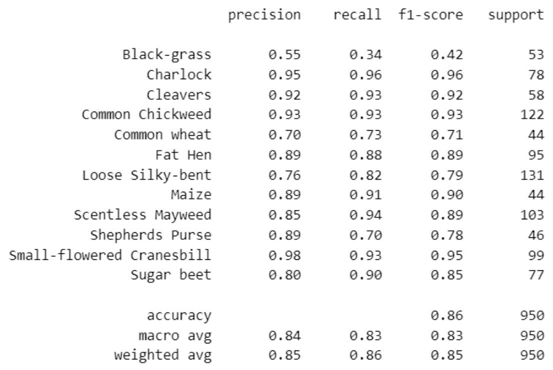
Different models were tested on in order to determine which one was the best to tackle the problem. In order to filter a first set of architectures to try with, a very brief revision of the state of the art was made. The articles revised highlighted VGG, Resnet50, Xception [5], MobileNetv2, SqueezeNet, AlexNet and ShuffleNet [6]. Out of these, only the former four models were implemented. There was an attempt to implement SqueezeNet as well, using an already existing pre-trained model for it, however there were issues with library compatibilities that were difficult to debug, so the model was dropped. Due to being similar except for the fact that it introduces non-linearity, the Inception model was also considered.

Due to the scope of the study not pertaining the creation of the models, but the comparison of the results they produce, most of the code used for the implementation of the models was either directly extracted from the libraries used (Keras), embedded from transfer learning or directly retrieved from already existing sources. In the case of the latter, those are cited accordingly.

1) MobileNet-v2

This model was imported directly from the Keras library and implemented using transfer learning, considering an image shape of (224, 224). Its results are shown below:



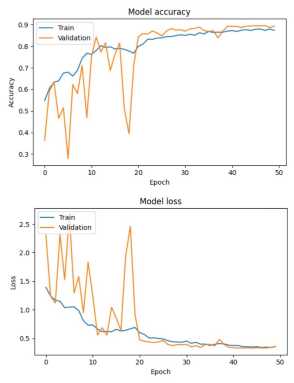


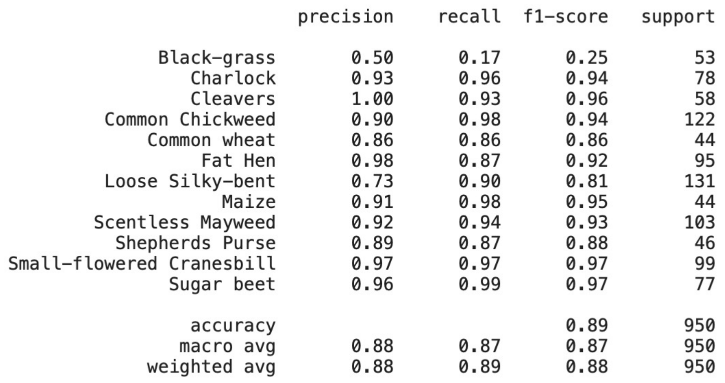
```
                        precision    recall  f1-score   support

           Black-grass       0.55      0.34      0.42        53
              Charlock       0.95      0.96      0.96        78
              Cleavers       0.92      0.93      0.92        58
      Common Chickweed       0.93      0.93      0.93       122
          Common wheat       0.70      0.73      0.71        44
               Fat Hen       0.89      0.88      0.89        95
      Loose Silky-bent       0.76      0.82      0.79       131
                 Maize       0.89      0.91      0.90        44
     Scentless Mayweed       0.85      0.94      0.89       103
       Shepherds Purse       0.89      0.70      0.78        46
Small-flowered Cranesbill    0.98      0.93      0.95        99
            Sugar beet       0.80      0.90      0.85        77

              accuracy                           0.86       950
             macro avg       0.84      0.83      0.83       950
          weighted avg       0.85      0.86      0.85       950
```

Fig. 3: MobileNet-v2 results

2) Inception V3

While this model was imported from the Keras library, it was implemented without transfer learning as applying it produced poor results of at most 0.20 accuracy. The final implementation's results are as follows:



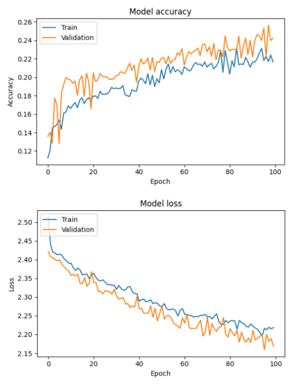


```
                        precision    recall  f1-score   support

           Black-grass       0.50      0.17      0.25        53
              Charlock       0.93      0.96      0.94        78
              Cleavers       1.00      0.93      0.96        58
      Common Chickweed       0.90      0.98      0.94       122
          Common wheat       0.86      0.86      0.86        44
               Fat Hen       0.98      0.87      0.92        95
      Loose Silky-bent       0.73      0.90      0.81       131
                 Maize       0.91      0.98      0.95        44
     Scentless Mayweed       0.92      0.94      0.93       103
       Shepherds Purse       0.89      0.87      0.88        46
Small-flowered Cranesbill    0.97      0.97      0.97        99
            Sugar beet       0.96      0.99      0.97        77

              accuracy                           0.89       950
             macro avg       0.88      0.87      0.87       950
          weighted avg       0.88      0.89      0.88       950
```

Fig. 4: Inception V3 results

3) ResNet 50

While this model was imported from the Keras library and implemented using transfer learning, considering an image shape of (70, 70). Its results are presented accordingly:

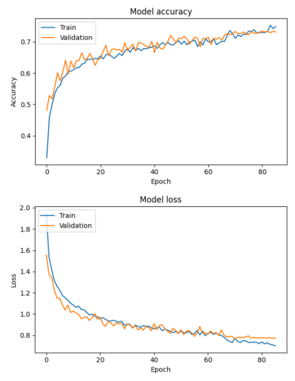


```
                           precision    recall  f1-score   support

              Black-grass       0.00      0.00      0.00        87
                 Charlock       0.25      0.29      0.27       129
                 Cleavers       0.00      0.00      0.00        95
         Common Chickweed       0.17      0.43      0.24       201
             Common wheat       0.00      0.00      0.00        73
                  Fat Hen       0.27      0.27      0.27       157
         Loose Silky-bent       0.42      0.51      0.46       216
                    Maize       0.00      0.00      0.00        73
        Scentless Mayweed       0.36      0.03      0.05       170
          Shepherds Purse       0.00      0.00      0.00        76
Small-flowered Cranesbill       0.21      0.57      0.30       164
               Sugar beet       0.62      0.04      0.07       127

                 accuracy                           0.24      1568
                macro avg       0.19      0.18      0.14      1568
             weighted avg       0.24      0.24      0.19      1568
```

Fig. 5: ResNet 50 results

4) VGGNet-16

At first there was an attempt to implement this model from scratch, however it did not perform as well as expected, it was very slow -which is a well recognized issue for this type of model- [7] and, on top of that, there were some issues when training it. Due to this reason it was ultimately imported from the Keras library and implemented using transfer learning considering a shape of (70, 70). Its results show the following:

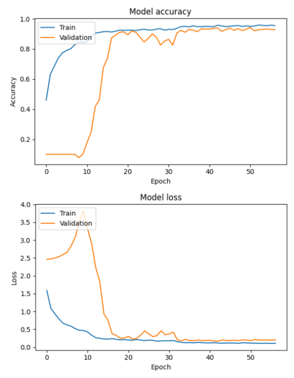


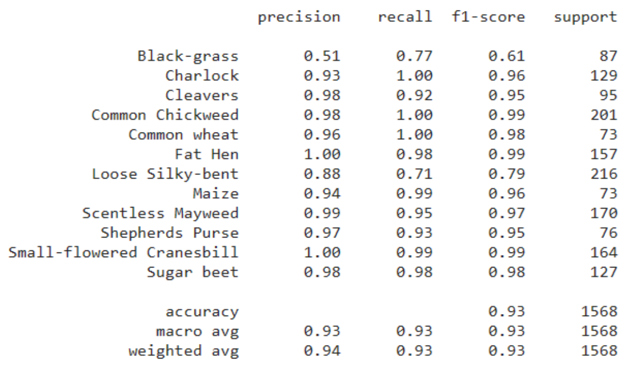
```
                           precision    recall  f1-score   support

              Black-grass       0.57      0.15      0.24        87
                 Charlock       0.76      0.86      0.81       129
                 Cleavers       0.76      0.69      0.73        95
         Common Chickweed       0.77      0.92      0.83       201
             Common wheat       0.57      0.53      0.55        73
                  Fat Hen       0.75      0.73      0.74       157
         Loose Silky-bent       0.68      0.84      0.75       216
                    Maize       0.75      0.71      0.73        73
        Scentless Mayweed       0.81      0.62      0.70       170
          Shepherds Purse       0.52      0.46      0.49        76
Small-flowered Cranesbill       0.82      0.94      0.88       164
               Sugar beet       0.71      0.72      0.72       127

                 accuracy                           0.73      1568
                macro avg       0.70      0.68      0.68      1568
             weighted avg       0.72      0.73      0.72      1568
```

Fig. 6: VGGNet-16 results

5) Xception

The implementation of this model was not through the use of an already trained model, but from scratch using the building blocks provided by Keras and code retrieved from an article by A. Sarkar [8]. The results it produced are displayed below:





```
                          precision    recall  f1-score   support

              Black-grass       0.51      0.77      0.61        87
                 Charlock       0.93      1.00      0.96       129
                 Cleavers       0.98      0.92      0.95        95
         Common Chickweed       0.98      1.00      0.99       201
             Common wheat       0.96      1.00      0.98        73
                  Fat Hen       1.00      0.98      0.99       157
         Loose Silky-bent       0.88      0.71      0.79       216
                    Maize       0.94      0.99      0.96        73
        Scentless Mayweed       0.99      0.95      0.97       170
          Shepherds Purse       0.97      0.93      0.95        76
Small-flowered Cranesbill       1.00      0.99      0.99       164
               Sugar beet       0.98      0.98      0.98       127

                 accuracy                           0.93      1568
                macro avg       0.93      0.93      0.93      1568
             weighted avg       0.94      0.93      0.93      1568
```

Fig. 7: Xception results

6) Comparison

A chart was made with the results obtained:

Model	Accuracy	Loss
MobileNet-v2	0.8579	0.4031
Inception V3	0.8946	0.3204
ResNet50	0.2423	2.169
VGGNet-16	0.7309	0.7744
Xception	0.9196	0.2281

TABLE I: Comparison of the models tested in terms of accuracy and loss

The results obtained during this first exploration of different types of models reflected that Xception was the best one to work with for this problem. Not only did it show the best results after training, but during training it also managed to obtain accuracy values over 0.9 and it also had the minimal loss. This coincides with the results observed in another study over the same dataset [5].

It is worth mentioning that ResNet showed the worst results, with it not being able to overpass 0.2 accuracy and having the highest loss, multiple repetitions and even trying without applying transfer learning.

III-C Training

We chose to work with the Xception model. The pipeline used for its training is shown as follows:

- Loss function: Categorical Crossentropy
- Optimizer: Adam
- Learning rate: 0.0001; validation loss reduction by 0.2

IV Evaluation

IV-A Metrics

As visualized previously, the metrics for accuracy, precision, recall and F1 score show very satisfactory results for the model, with Fat Hen and Small-flowered Cranesbill achieving perfect precision, even. However the Black-grass class showed lower scores than the rest of classes. A confusion matrix was made to help identify why.

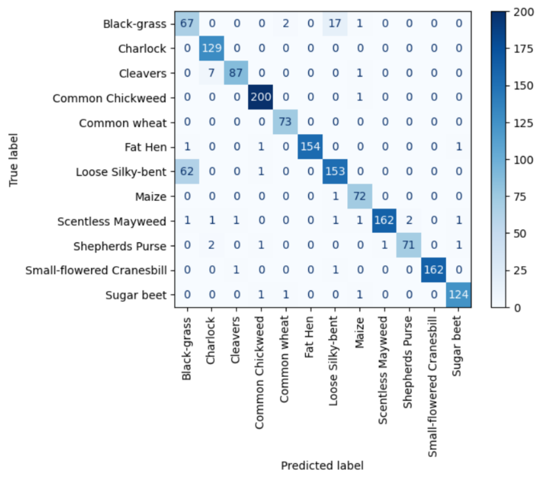
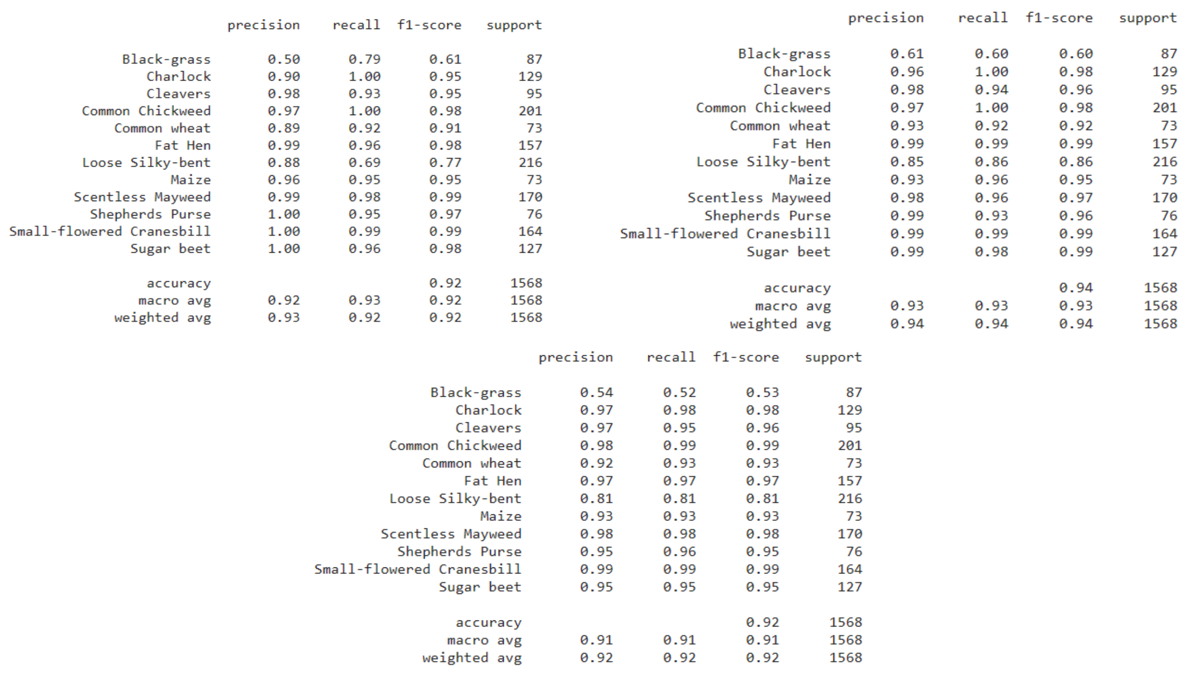


Fig. 8: Confusion Matrix

Noticing the mistakes in classification shown by the confusion matrix, an attempt to minimize biases by fixing class imbalance further through the application of techniques such as SMOTE was made. However, despite the successful results in getting an equal number of samples as pictured in the image below, performance of the model worsened, with the model only being able to achieve up until 0.3 accuracy. For this reason tests with the other models were also applied, but they led to similar results.

IV-B Parameter Tuning

As seen in previous sections, while accuracy, precision and f1-scores were rather high, the loss and accuracy were constantly diverging throughout the training. Due to this reason, the training rate was modified to test if the results changed. The following results were obtained during training:

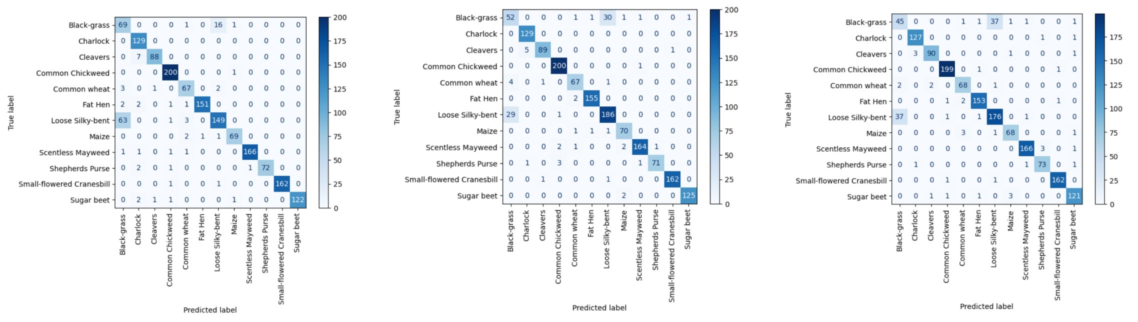


```
                           precision    recall  f1-score   support

              Black-grass       0.50      0.79      0.61        87
                 Charlock       0.90      1.00      0.95       129
                 Cleavers       0.98      0.93      0.95        95
         Common Chickweed       0.97      1.00      0.98       201
             Common wheat       0.89      0.92      0.91        73
                  Fat Hen       0.99      0.96      0.98       157
         Loose Silky-bent       0.88      0.69      0.77       216
                    Maize       0.96      0.95      0.95        73
        Scentless Mayweed       0.99      0.98      0.99       170
          Shepherds Purse       1.00      0.95      0.97        76
Small-flowered Cranesbill       1.00      0.99      0.99       164
               Sugar beet       1.00      0.96      0.98       127

                 accuracy                           0.92      1568
                macro avg       0.92      0.93      0.92      1568
             weighted avg       0.93      0.92      0.92      1568
```

```
                           precision    recall  f1-score   support

              Black-grass       0.61      0.60      0.60        87
                 Charlock       0.96      1.00      0.98       129
                 Cleavers       0.98      0.94      0.96        95
         Common Chickweed       0.97      1.00      0.98       201
             Common wheat       0.93      0.92      0.92        73
                  Fat Hen       0.99      0.99      0.99       157
         Loose Silky-bent       0.85      0.86      0.86       216
                    Maize       0.93      0.96      0.95        73
        Scentless Mayweed       0.98      0.96      0.97       170
          Shepherds Purse       0.99      0.93      0.96        76
Small-flowered Cranesbill       0.99      0.99      0.99       164
               Sugar beet       0.99      0.98      0.99       127

                 accuracy                           0.94      1568
                macro avg       0.93      0.93      0.93      1568
             weighted avg       0.94      0.94      0.94      1568
```

```
                           precision    recall  f1-score   support

              Black-grass       0.54      0.52      0.53        87
                 Charlock       0.97      0.98      0.98       129
                 Cleavers       0.97      0.95      0.96        95
         Common Chickweed       0.98      0.99      0.99       201
             Common wheat       0.92      0.93      0.93        73
                  Fat Hen       0.97      0.97      0.97       157
         Loose Silky-bent       0.81      0.81      0.81       216
                    Maize       0.93      0.93      0.93        73
        Scentless Mayweed       0.98      0.98      0.98       170
          Shepherds Purse       0.95      0.96      0.95        76
Small-flowered Cranesbill       0.99      0.99      0.99       164
               Sugar beet       0.95      0.95      0.95       127

                 accuracy                           0.92      1568
                macro avg       0.91      0.91      0.91      1568
             weighted avg       0.92      0.92      0.92      1568
```

Fig. 9: Classification Report by Learning Rate (0.001, 0.01, 0.1)



	Black-grass	Charlock	Cleavers	Common Chickweed	Common wheat	Fat Hen	Loose Silky-bent	Maize	Scentless Mayweed	Shepherds Purse	Small-flowered Cranesbill	Sugar beet
Black-grass	69	0	0	0	1	0	16	1	0	0	0	0
Charlock	0	129	0	0	0	0	0	0	0	0	0	0
Cleavers	0	7	88	0	0	0	0	0	0	0	0	0
Common Chickweed	0	0	0	200	0	0	0	1	0	0	0	0
Common wheat	3	0	1	0	67	0	2	0	0	0	0	0
Fat Hen	2	2	0	1	1	151	0	0	0	0	0	0
Loose Silky-bent	63	0	0	1	3	0	149	0	0	0	0	0
Maize	0	0	0	0	2	1	1	69	0	0	0	0
Scentless Mayweed	1	1	0	1	1	0	0	0	166	0	0	0
Shepherds Purse	0	2	0	1	0	0	0	0	1	72	0	0
Small-flowered Cranesbill	0	0	0	1	0	0	1	0	0	0	162	0
Sugar beet	0	2	1	1	0	0	0	1	0	0	0	122

	Black-grass	Charlock	Cleavers	Common Chickweed	Common wheat	Fat Hen	Loose Silky-bent	Maize	Scentless Mayweed	Shepherds Purse	Small-flowered Cranesbill	Sugar beet
Black-grass	52	0	0	0	1	1	30	1	1	0	0	1
Charlock	0	129	0	0	0	0	0	0	0	0	0	0
Cleavers	0	5	89	0	0	0	0	0	0	0	1	0
Common Chickweed	0	0	0	200	0	0	0	0	1	0	0	0
Common wheat	4	0	1	0	67	0	1	0	0	0	0	0
Fat Hen	0	0	0	0	2	155	0	0	0	0	0	0
Loose Silky-bent	29	0	0	1	0	0	186	0	0	0	0	0
Maize	0	0	0	0	1	1	1	70	0	0	0	0
Scentless Mayweed	0	0	0	2	1	0	0	2	164	1	0	0
Shepherds Purse	0	1	0	3	0	0	0	0	1	71	0	0
Small-flowered Cranesbill	0	0	1	0	0	0	1	0	0	0	162	0
Sugar beet	0	0	0	0	0	0	0	2	0	0	0	125

	Black-grass	Charlock	Cleavers	Common Chickweed	Common wheat	Fat Hen	Loose Silky-bent	Maize	Scentless Mayweed	Shepherds Purse	Small-flowered Cranesbill	Sugar beet
Black-grass	45	0	0	0	1	1	37	1	1	0	0	1
Charlock	0	127	0	0	0	0	0	0	0	1	0	1
Cleavers	0	3	90	0	0	0	0	1	0	0	0	1
Common Chickweed	0	0	0	199	0	1	0	0	0	0	1	0
Common wheat	2	0	2	0	68	0	1	0	0	0	0	0
Fat Hen	0	0	0	1	2	153	0	0	0	0	1	0
Loose Silky-bent	37	0	0	1	0	1	176	0	1	0	0	0
Maize	0	0	0	0	3	0	1	68	0	0	0	1
Scentless Mayweed	0	0	0	0	0	0	0	0	166	3	0	1
Shepherds Purse	0	1	0	0	0	0	0	0	1	73	0	1
Small-flowered Cranesbill	0	0	0	1	0	0	1	0	0	0	162	0
Sugar beet	0	0	1	1	0	1	0	3	0	0	0	121

Fig. 10: Confusion Matrix by Learning Rate (0.001, 0.01, 0.1)

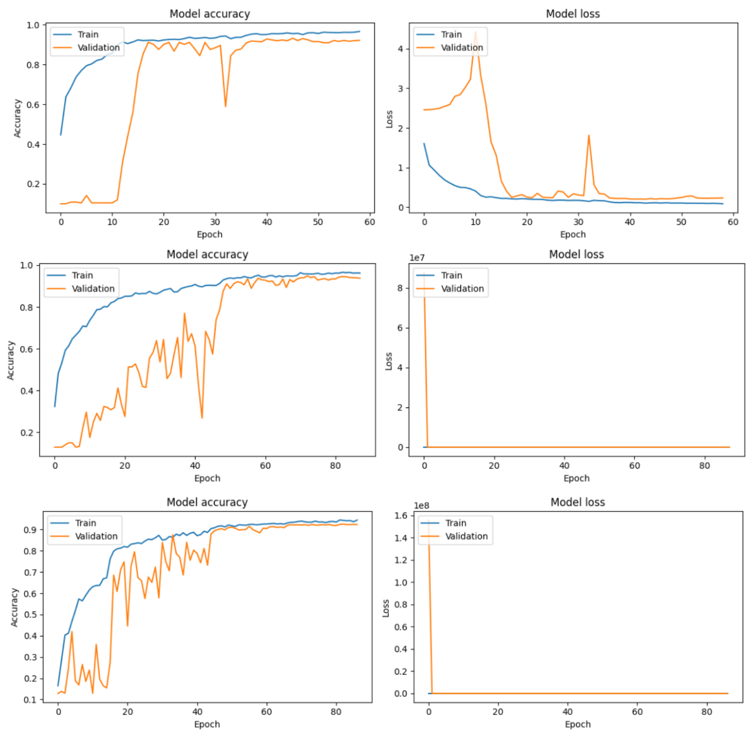


Fig. 11: Accuracy/Loss Graphs by Learning Rate (0.001, 0.01, 0.1)

Learning Rate	Accuracy	Loss	Submission Accuracy
0.001	0.920	0.2314	0.91183
0.01	0.9375	0.2059	0.93073
0.1	0.9235	0.2203	0.92695

TABLE II: Comparison learning rates in terms of accuracy and loss

The results reflect that working with a learning rate of 0.01 produced the most accurate results.

V CONCLUSIONS

The designed Xception model with categorical crossentropy as its loss function, ADAM as its optimizer and a learning rate of 0.01 with a validation loss reduction by 0.2 was able to produce satisfactory results for the classification of different species of plants for the Plant Seedlings database, achieving about 90% accuracy in the testing data, results better than the other architectures and hyperparameters it was compared with.

It was noticed that the probable source of the misclassifications for black-grass images was due to their shape being difficult to distinguish, and their mask not covering everything it should properly. Therefore, further scrutinization of the masking process and how to deal with this specific type of plant would be advised.

Aside from that, it would be important to try working with other architectures, specially SqueezeNet, since the state of the art suggested that it worked well in images.

VI APPENDIX & REFERENCES

APPENDIX

(A) Link to main notebook

REFERENCES

[1] T. M. Giselsson, R. N. Jørgensen, P. K. Jensen, M. Dyrmann, H. S. Midtiby. *A Public Image Database for Benchmark of Plant Seedling Classification Algorithms.* 2017. Retrieved from https://arxiv.org/abs/1711.05458

[2] M. Dyrmann. *Plants Seedlings Dataset*. 2018. Retrieved from https://vision.eng.au.dk/plant-seedlings-dataset/

[3] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. 2016. MIT Press. Available: http://www.deeplearningbook.org

[4] Google Developers. *Introducing Convolutional Neural Networks*. 2022. Retrieved from /urlhttps://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

[5] Q. Li, Y. Hu, Q. Wang, Z. Liu. *Plant Seedling Classification*. 2019. Retrieved from http://noiselab.ucsd.edu/ECE228_2019/Reports/Report16.pdf

[6] Z. Fan, M. Jamil, M. Tariq Sadiq, X. Huang, X. Yu, *Exploiting Multiple Optimizers with Transfer Learning Techniques for the Identification of COVID-19 Patients*, Journal of Healthcare Engineering, vol. 2020, Article ID 8889412, 13 pages, 2020. https://doi.org/10.1155/2020/8889412

[7] A. Rosebrock. *ImageNet: VGGNet, ResNet, Inception, and Xception with Keras*. 2017. Retrieved from https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/

[8] A. Sarkar. Xception: Implementing from scratch using Tensorflow https://towardsdatascience.com/xception-from-scratch-using-tensorflow-even-better-than-inception-940fb231ced9