POLITECNICO DI MILANO
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MSc. Computer Science and Engineering
Academic Year 2023/2024

POLITECNICO
MILANO 1863

Team CAIPIRINHA
Homework 1

Authors:
Domingos Savio Pinheiro do Nascimento Junior (10880091)
domingossavio.pinheiro@mail.polimi.it

Felipe Azank dos Santos (10919711)
felipe.azank@mail.polimi.it

Felipe Bagni (10912321)
felipe.bagni@polimi.it

Professors:
Matteo Matteuci
Giacomo Boracchi

Nov 2023

1 Introduction

The following report serves as a brief explanation of what was done by the group in regards to completing the first homework competition, which consisted on an Image Classification with the goal of identifying if a leaf image consists on a healthy or unhealthy plant.

Figure 1: Unhealthy (1) and Healthy (0) example

2 Exploratory Data Analysis and Data Cleaning

Given the dataset for the Image Classification problem, an exploratory data analysis is done, so noisy data can be removed and a new dataset with cleaned data can be created.

2.1 Data Inspection

After loading the data into a jupyter notebook, the first step taken was to check the completeness of the data and the label distribution, to check if we would face any data imbalance problems.

Label	# of images
Healthy (0)	3101
Unhealthy (1)	1903

Table 1: Label Distribution after Shrek and Trololo removal

As we can see, there is a small imbalance between the quantity of healthy and unhealthy leafs, but in principle, a 62% concentration of the main class would not, imply the need for oversampling the minor class, given that we can deal with it using augmentation (seen further in the report).

Furthermore, after some inspection of the images, it was possible to notice that there were noisy and inputted data (98 Shreks and 98 Trololos), which should and have been be removed using Numpy masks.

Figure 2: Noisy data example

3 Methodology

At first, the progressive approach was given to the project in which a simple Convolutional Neural Network with a few layers was designed, controlling the bias-variance trade-off with Dropout layers. After that, the Data Augmentation approach was taken, given that it increases the amount of training data and makes more difficult for the model to memorize the training data (overfitting the model). Finally, Transfer Learning was applied in order to take advantage off the pre-trained models. All the architectures were done following Keras documentation [1].

3.1 Data Preparation

Before the creation of all the architectures, a simple preprocessing of the clean data was made in order to prepare the input of the networks.

In order to avoid data leakage, the first process was to split the data into 3 parts: 500 samples for testing, 500 samples for validation and 4004 samples for training. The split was made in a stratified manner, given that there's a clear dataset imbalance and not doing this could damage the validation, testing and training.

After that, all data was normalized between 0 and 1, in order to reduce the computational complexity on Feed-forwarding and Back-propagation, ensuring that each feature contributes more uniformly to the learning process, mitigating gradient problems and improving generalization. Furthermore, the normalization proves to be fundamental to apply the Transfer Learning, as pre-trained models are trained with normalized data as well.

3.2 Simple CNN Approach

The first proposed model to understand the process of neural network creation consisted on applying a sequence of Convolutional Layers in sequence with a constant number of channels, in order to try to extract the main features of the images.

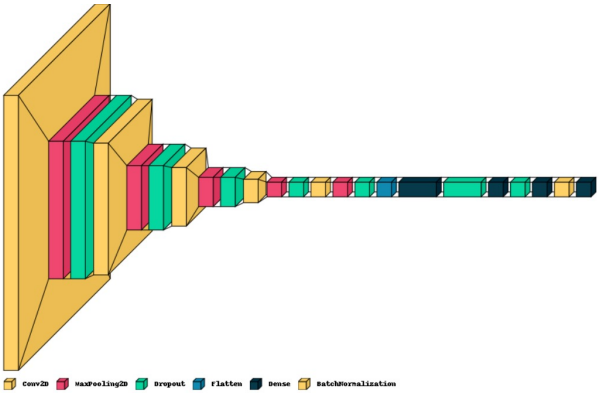

Figure 3: First Model Architecture

After testing, it was possible to see that a network with this amount of parameters was leading to an overfit. To avoid this, a few layers of dropout were added, reducing the variance by ”ensembling” the network.

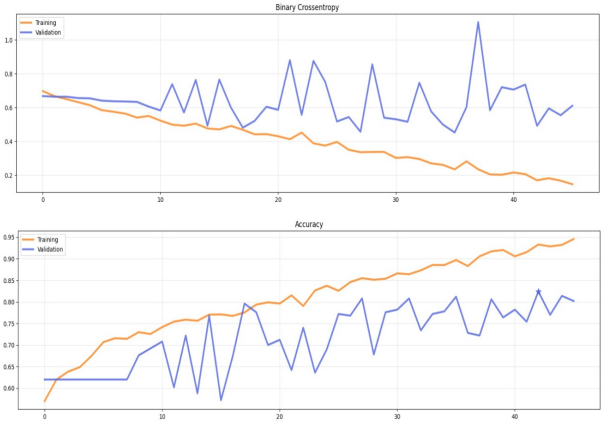

Figure 4: First Model Results on Test Set: Accuracy (0.778), Precision (0.7232), Recall (0.6737), F1 (0.6975)

As we can see, the model indeed has learned something regarding the patters from the process, however, it still struggles on obtaining a deeper understanding of the patterns while overfits the data. Given that, we explore a new model applying data augmentation, so that we can obtain a better performance with different generated images.

3.3 Simple CNN with Data Augmentation

Now that we created a simple model and seen the performance of the task, we can now use data augmentation to check if the performance can be increased [2]..

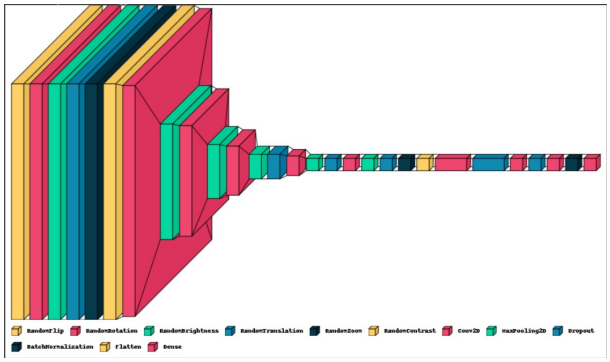

Figure 5: Second Model Architecture

The augmentation presented is extremely important given that it effectively addresses the challenges posed by limited training data. By artificially expanding the dataset through diverse transformations, it mitigates the risk of overfitting and equips the model with a richer understanding of potential variations in the input data, avoiding data memorization.

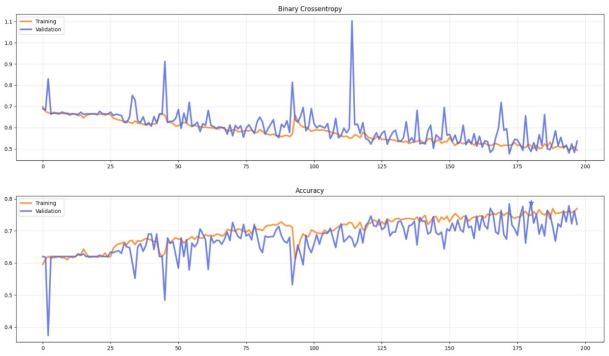

Figure 6: Second Model Results on Test Set: Accuracy (0.814), Precision (0.814), Recall (0.647), F1 (0.721)

3.4 Transfer Learning with Data Augmentation

Now that we seen how augmentation can be added to improve the model, its time to use already built in networks that have been trained with bigger datasets in order to try extracting certain patterns that could be useful for the problem at hand [3].

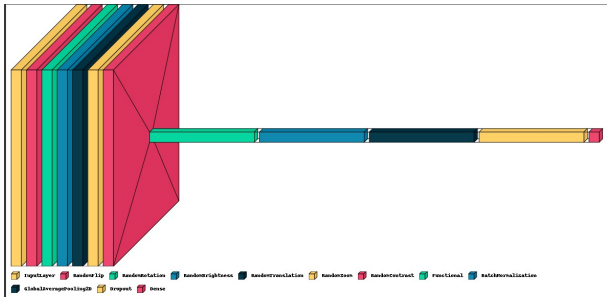

Figure 7: Third Model Architecture

In order to build the new model, we import a well-known architecture from `keras.applications` in

order to import its overall structure without the original fully connected layers on top, since these where changed to fit the current problem. The model chosen for size and processing reasons was the ResNet50.

In addition to the convolutional and process layers inside we will also add the same augmentations presented before in order to extract the best value of our data. Some simple initialization techniques in order to help convergence and avoid symmetry on the network training where also used. Furthermore, we also used a simple GlobalAveragePooling2D layer, that downsamples each channel into a single value, simplifying the input for the classification process.

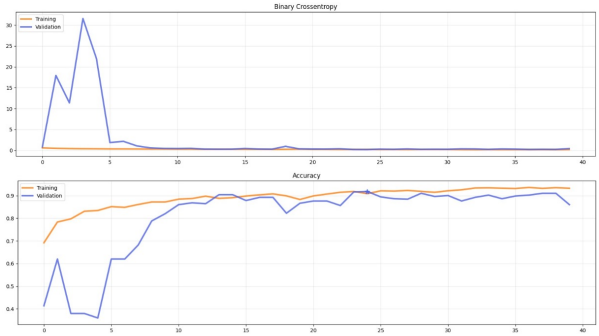

Figure 8: Third Model Results on Test Set: Accuracy (0.904), Precision (0.8901), Recall (0.8526), F1 (0.871)

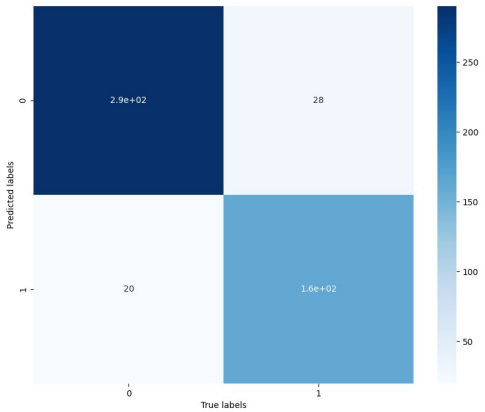

Figure 9: Confusion Matrix on Test Set: Accuracy (0.904), Precision (0.8901), Recall (0.8526), F1 (0.871)

4 Conclusion

From the results obtained, it is possible to see that Data Augmentation and Transfer Learning can improve drastically the performance of Convolutional Neural Networks on Image Classification tasks, being a definitive turning point for the Deep Learning development and research.

Model	Accuracy	Precision	Recall	F1
3.2	0.778	0.7232	0.6737	0.6975
3.3	0.814	0.814	0.647	0.721
3.4	0.904	0.8901	0.8526	0.871

Table 2: Model Results on Test Set

It's also fundamental to notice that, the Transfer Learning approach presented a much better performance, specially with regards to the recall, by reducing the quantity of False Negative occurrences, that usually can be hard to detect given a data imbalance.

Finally, it is also important to note that the solution development is a iterative process that requires to control the variance-bias trade-off by searching methods to control overfitting (something that happens with considerable frequency on large neural networks). During this project, this tradeoff was explored by using Dropout layers, Dense Layer addition and removal, Number of neurons control, Data Augmentation and weight conservation of pre-trained layers.

5 Contributions

All members were involved during all steps of the project, as we worked together. Notably:

Felipe Azank dos Santos: Data Preprocessing, CNN architectures and hyper parameter tuning.

Felipe Bagni: Exploratory Data Analysis, finding the problematic data, image augmentation and CNN architectures.

Domingos Savio: Running and exploring alternative CNN architectures and Transfer Learning benchmarking.

References

[1] Keras applications. `https://www.tensorflow.org/api_docs/python/tf/keras/applications`. Accessed: 18^{th} *December*, 2023.

[2] S. Hamida, O. El Gannour, A. Maafiri, Y. Lamalem, I. Haddou-Oumouloud, and B. Cherradi. Data balancing through data augmentation to improve transfer learning performance for skin disease prediction. In *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 1–7, 2023. doi: 10.1109/IRASET57153.2023.10152920.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.