POLITECNICO
MILANO 1863

AN2DL - Homework 1
Image Classification - TensorFlex

Sironi Alessandro, Stefanizzi Giovanni, Stefanizzi Tomaso, Villa Ismaele

---

Dataset specification

The dataset employed in this neural network challenge comprised a collection of 5200 leaf images with dimensions of 96x96 pixels and 3 channels corresponding to the red, green, and blue (RGB) color channels. The dataset was organized into two distinct classes for binary classification: *healthy* and *unhealthy*.

Inspecting data

After loading the dataset, we proceeded by removing all the outliers using the “Mean Squared Error” (mse) function. This way, once we found the first occurrence of the two types of outliers (Shreks and Trololos), all identical among them, we removed all images with mse equal to zero from the dataset.
The sanitized images are then casted to be float tensors of pixels with values in the [0-255] range, while the labels are encoded with values between 0 and 1 (‘0’: healthy, ‘1’: unhealthy).
Eventually, the dataset is split into training, validation and test sets with the following shapes:

- images_train shape: (4160, 96, 96, 3), labels_train shape: (4160, 2)
- images_val shape: (936, 96, 96, 3), labels_val shape: (936, 2)
- images_test shape: (104, 96, 96, 3), labels_test shape: (104, 2)

Image augmentation

Using TensorFlow's Keras Sequential API, a comprehensive image augmentation strategy was implemented to diversify the dataset for training the Convolutional Neural Network (CNN).
After various attempts, the final augmentation pipeline consisted of multiple transformations, including random translations within a 15% range horizontally and vertically, 10% random zooming, horizontal and vertical flips, and random rotations up to a maximum angle of 0.2 radians.
These varied transformations introduced substantial diversity into the dataset, broadening the range of training samples. Consequently, this augmented dataset facilitated the CNN's ability to learn distinctive features, minimize overfitting, and improve its adaptability to novel leaf images, ultimately enhancing the model's robustness and generalization during training.

Building the model

Two main approaches were taken in the development of the neural network:

- Convolutional Neural Network from scratch.
- Transfer Learning and Fine Tuning using a known network.

CNN

Our first approach was to build a CNN from scratch. The model consisted of 4 convolutional layers, an activation layer using 'ReLu' activation function, a pooling layer using the MAX operation and eventually a GAP (Global Average Pooling) and a dropout layer were added.
This approach did not provide very good performance so we decided to move to a transfer learning model.

Transfer Learning and Fine Tuning

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 96, 96, 3)]	0
preprocessing (Sequential)	(None, 96, 96, 3)	0
convnext_large (Functional)	(None, 3, 3, 1536)	196230336
dropout (Dropout)	(None, 3, 3, 1536)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
Output (Dense)	(None, 2)	3074

Total params: 196233410 (748.57 MB)
Trainable params: 3074 (12.01 KB)
Non-trainable params: 196230336 (748.56 MB)

Switching strategy from the aforementioned network, we decided to explore the transfer learning techniques by starting from the example seen in the laboratories. The weights of all the networks mentioned below were initialized with *imagenet* weights. At the edge of the network a dropout layer was included to mitigate overfitting.
The first network we included in our model was *mobileNetV2* [1]. This model didn't give us the expected results, as the accuracy score was the lowest we ever had.
We promptly switched to *efficientNet* [2], testing both the Medium and Large models, settling on the largest one. Using efficientNet, we kept refining our parameters to obtain a higher score in the submissions.

In the last days of the competitions, we decided to try out *convNeXt* [3] and we discovered that this network consistently gives out the best score results.

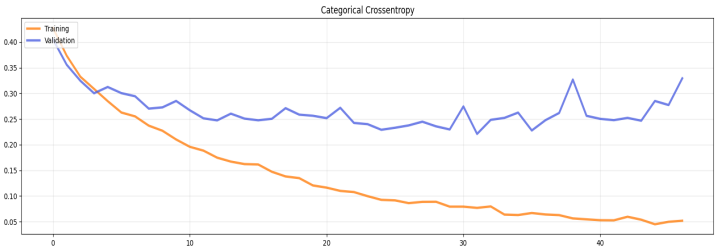
Model Performance Comparison - Local test set

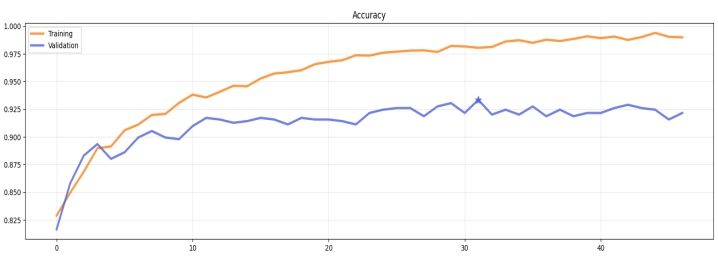
Model	Test Accuracy	Precision	Recall	F1
CNN from scratch	0.8212	0.8173	0.7994	0.8061
MobileNetV2	0.6220	0.5894	0.5067	0.4071
EfficientNetV2S	0.9020	0.8947	0.8986	0.8965
ConvNeXtLarge	0.9067	0.9028	0.9239	0.9050

Training the model

For model training we set the batch size and the epochs hyperparameters respectively to 32 and 500 as these values provide us the best performance in terms of accuracy. While we use EarlyStopping callbacks API with patience equal to 100.
The model is compiled using 'CategoricalCrossentropy', as this suits our design choice of having a dense layer of two neurons as the last layer of the CNN with Adam optimizer function with 1e-3 learning rate for the former phase, while 1e-5 for the latter.
We then added a GAP layer, a dropout layer and an l2-norm regularizer, reaching an accuracy of 0.92. By increasing both the dropout percentage from 0.2 to 0.3 and regularization strength parameter from 0,01 to 0.03, we finally reached an accuracy of 0.94 on the Codalab test set in phase 1.

Results





The final results we achieve on Codalab are overall satisfying. We managed to reach a 0.8850 accuracy score on Codalab in the second phase of the challenge, which was over 0.94 in the first phase of the competition. A particular choice we applied during the local training was to split the dataset in just training and validation sets, and directly test the model on the competition platform to rely on a major number of data for the training/validation part. This way we slightly increased our score on the last day of the competition.

References

[1] MobileNetV2: Inverted Residuals and Linear Bottlenecks ,The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520

[2] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proceedings of the International Conference on Machine Learning, 2019*, pp. 6105-6114, 2019.

[3] X. Yan, J. Yang, K. Zhang, and Z. Zhang, "ConvNeXt: A Convolutional Neural Network with a Flexible Context Aggregation Layer," *Proceedings of the European Conference on Computer Vision (ECCV), 2018*, pp. 768-784, 2018.

Contributions

DATASET INSPECTION AND OUTLIERS REMOVAL: *Sironi Alessandro, Villa Ismaele*

DATA AUGMENTATION: *Sironi Alessandro, Villa Ismaele*

CNN FROM SCRATCH: *Sironi Alessandro, Stefanizzi Giovanni, Stefanizzi Tomaso, Villa Ismaele*

TRANSFER LEARNING + FINE TUNING: *Sironi Alessandro, Stefanizzi Giovanni, Stefanizzi Tomaso, Villa Ismaele*

DOCUMENTATION AND OVERALL CODE: *Sironi Alessandro, Stefanizzi Giovanni, Stefanizzi Tomaso, Villa Ismaele*