Convolutional Neural Network: Plants Classification

Artificial Neural Networks and Deep Learning Homework 1 – A.Y. 2023/2024

Marton Barta[*], Al Kamber[†], and Federica Vinciguerra[‡]
Email: [*]marton.barta@mail.polimi.it, [†]al.kamber@mail.polimi.it, [‡]federica.vinciguerra@mail.polimi.it
Student ID: [*]10884623, [†]10637023, [‡]10921587,
Codalab Group: “Gradient Gamblers”

1. Introduction

The dataset comprised 5200 images categorized into two principal classes: healthy and unhealthy specimens. Preliminary examination revealed the dataset was compromised by extraneous image types: memes and images resembling the character Shrek. A suite of clustering methods was evaluated to cleanse the dataset, with DBSCAN emerging as the most efficacious (refer to the accompanying image). Analyzing

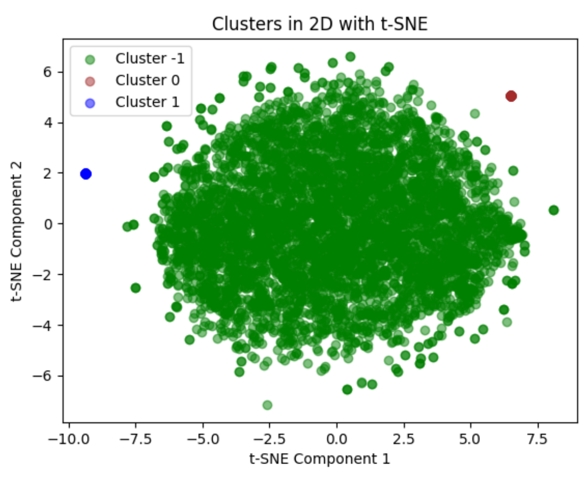

Figure 1. Illustrative figure of the 2D DBSCAN with epsilon = .1.

the image, a distinct clustering pattern is observed, characterized by three discernible clusters. Of these clusters, one is denoted as Cluster -1, representing the data points corresponding to the plants under examination. The remaining two clusters are identified as outliers within the scope of this study and therefore eliminated, culminating in a refined dataset of 5003 images, segmented into 3100 healthy and 1903 unhealthy specimens. The 2D plot was done with the help of t-SNE algorithm.

Our objective was to construct a binary image classifier leveraging a Convolutional Neural Network (CNN), to transmute multidimensional image tensors into definitive probabilistic outputs. A probability exceeding the 50 percent threshold was indicative of the positive class, and conversely for the negative class.

Binary cross-entropy was adopted as the loss function, with Accuracy serving as the primary performance metric. To obtain a more granular evaluation of model performance, the Top N percent metric was introduced (explained later), offering deeper insights into predictive capabilities.

The dataset underwent a partitioning of 85:15 between training and validation subsets, with the ensuing report reflecting the performance on the validation set.

2. Data Pipeline

2.1. Pre-Processing

The dataset was partitioned into training and validation sets with an 80:20 ratio and then normalized by scaling pixel values to the [0, 1] range via division by 255.

Subsequent models employed the `preprocess_input` function from `tensorflow.keras.applications`, incorporating it as a preprocessing layer within the transfer learning architectures.

2.2. Data Augmentation

To compensate for the relatively modest size of the dataset, data augmentation was instituted at the onset across all model iterations, thereby enhancing generalization and performance.

Traditional transformations, including Rotation, Zoom, Flip, Shift, and Shear, were systematically employed. Their ranges were also parameterized and served as hyperparameters during optimization, specifically to tune them with a Bayesian Optimizer from `keras_tuner`. However, transformations that modified color properties were intentionally avoided because they resulted in decreased model performance.

Although advanced augmentation methods like Mix-up were explored, they did not yield accuracy improvements. Consequently, emphasis was redirected towards traditional augmentation strategies.

3. Convolutional Neural Network

We initiated with a prototypical CNN, integrating three blocks of `Conv2D`, `MaxPooling2D`, and `Dropout` layers, followed by a `Dense` layer with 64 neurons and a `Dense` output layer. This baseline model achieved an initial accuracy of 71.52%.

Subsequently, a Bayesian Optimization approach was undertaken to fine-tune the kernel sizes, filter sizes, activation functions, and neuron count in the dense layer.

Utilizing the optimized hyperparameters resulted in an appreciable enhancement of our testing accuracy by 75.43%. Despite this improvement, the performance had not yet met the predetermined benchmark of excellence.

In pursuit of superior accuracy, we pivoted to a strategy known as transfer learning.

3.1. Transfer Learning

We harnessed a selection of preeminent models pre-trained on the ImageNet database, with the following preliminary accuracies documented:

InceptionV3	76.50%
ResNet-50	89.89%
ConvNext	90.2%
ResNet-152	91.2%
EfficientNet-b3	89.34%
ResNet-101	90.65%

Our comparative analysis indicated ConvNext and ResNet-152 as front-runners. To further refine our model's precision, a meticulous approach was undertaken, focusing on the fine-tuning of specific layers conducive to further training. A Bayesian Optimization approach was applied to find the number of layers to unfreeze and the ranges for data augumentation (as mentioned before), as well as the dropout rates and the number of nodes of the dense layers. This culminated in a net increment of 7%.

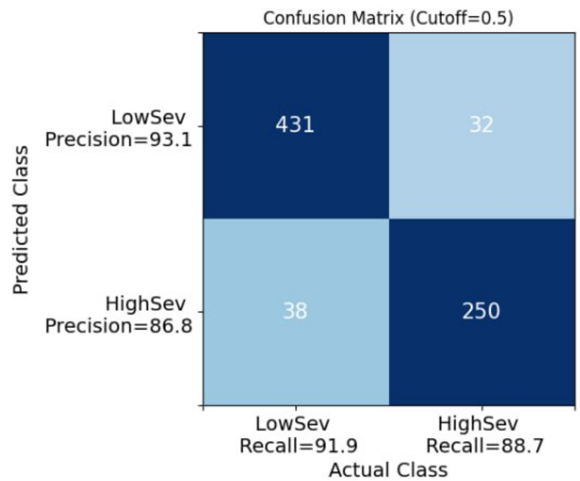

Figure 2. Transfer Learned ConvNextLarge model's Confusion Matrix

3.2. Ensembling

In the final phase of our project, we implemented an ensemble approach by integrating three distinct models: ConvNext, ResNet152 and ResNet101. This strategy hinged on the principle of `majority voting` for classification decisions. Each model independently classified the images, and the final class was determined by the majority vote among the three. This ensemble method significantly enhanced our model's robustness.

Additionally, we explored a hybrid approach that combined prediction probabilities of models EfficientNet-b3 and ConvNextLarge in an equal 50-50% ratio. This method involved `averaging the prediction probabilities` from both models for each class and then selecting the class with the higher average probability as the final prediction. The two specific models were chosen because of their strength in capturing the intricate features and patterns inherent to one particular class. This approach yielded a 94% validation accuracy, showcasing the effectiveness of probabilistic ensembling.

The last facet of our ensembling strategy involved a more nuanced approach, termed the `Top N percent` method. Here, we assigned variable weights to the prediction probabilities of models ConvNext and ResNet-152, based on specific probability thresholds established by model ResNet-152. This method entailed plotting the highest prediction probabilities for each class and evaluating the accuracy at various thresholds. The weight assigned to each model's prediction was adjusted according to these thresholds, with the aim of maximizing accuracy in uncertain scenarios. This percentage-based ensembling approach allowed for a dynamic and context-sensitive classification, further refining our model's predictive capabilities.

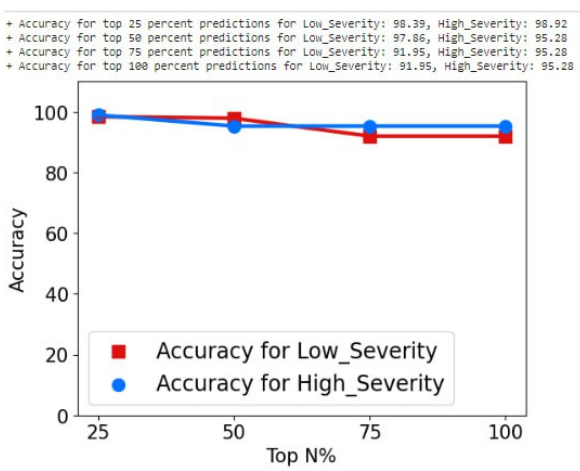

Figure 3. ResNet-152 with 30 unfrozen layers' Top N graph

Each of these ensembling techniques contributed to the robustness and accuracy of our overall classification system, demonstrating the power of combining diverse models and methodologies in machine learning.

3.3. Final Submission

The model that performed the best utilizes the approaches the worked the best out of all the ones described in this report. More specifically, we opted for the ensemble of two models trained via transfer learning from ConvNextLarge and EfficientNet architectures, each of which have 30 unfrozen layers, the optimal parameters found via Bayesian Optimization and the traditional data augmentation methods described before. A weight of 0.5 is assigned to each model and a weighted sum is computed to obtain the final prediction. The submission on Codalab gave us the following results:

Accuracy	0.7990 (93)
Precision	0.6845 (351)
Recall	0.8737 (18)
F1 Score	0.7676 (236)

3.4. Final Conclusions

The current results show promise, but further experimental testing is necessary to fine-tune the model for improved accuracy. The ensemble approach showed good results but it could have been implemented differently in order to obtain a better accuracy. Also, the different data augmentation methods, if implemented correctly, could also improve the model's performance. All in all, this challenge was a great opportunity to work in a practical way on the concepts seen during the course.

4. Contributions

- **Federica Vinciguerra:** Data Augmentation, Ensembling Methods, Transfer Learning, Report Writing
- **Marton Barta:** Preliminary Screening, Report Writing, Graphs Creation, Ensembling Methods
- **Al Kamber:** Model Training, Report Writing, Fine-Tuning, Ensembling Methods, Transfer Learning