Kinga Marek 10844847 **Andrés Pasinetti** 10468985 **Marco Verzeni** 10577271

DATASET ANALYSIS

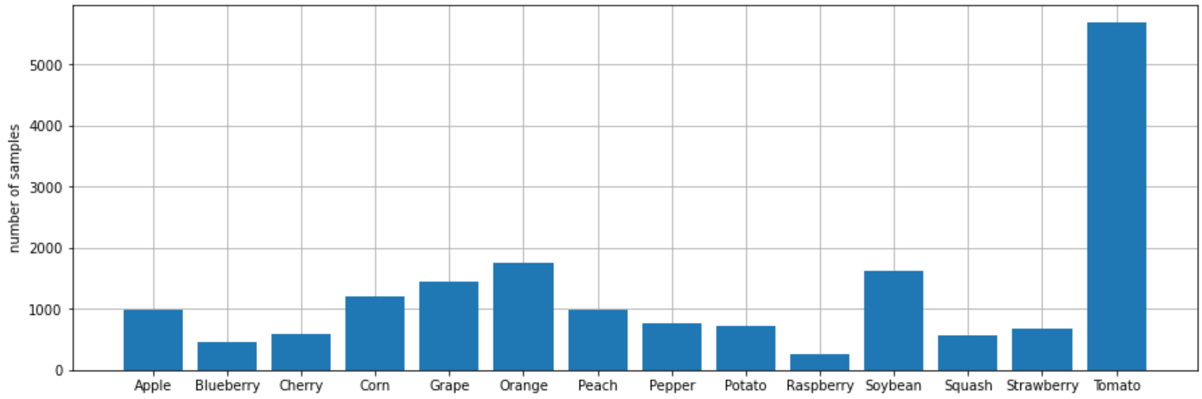

Figure 1. The number of samples in a class

Analysing the data distribution over the classes, it immediately comes up that tomatoes are by far the most common class among the dataset, constituting 32% of the whole training dataset. In contrast, raspberries are the least common class.

Indeed, the dataset is imbalanced, but there is also another possible problem we will have to deal with: data scarcity. For example, the number of samples of raspberries is only 264.

MOTIVATION FOR MODEL CHOICE

Given the complexity of the problem as well as the data scarcity, we decided to leverage transfer learning. Simultaneously, we decided to devise our own CNN model similar to the one presented during the laboratory session. In both cases, we used keras.tuner to find the best number of neurons in the hidden layers. Nevertheless, only the first stages of development showed the superiority of the transfer learning model. Each model was trained using data augmentation techniques. Above, you can see the comparison of the training

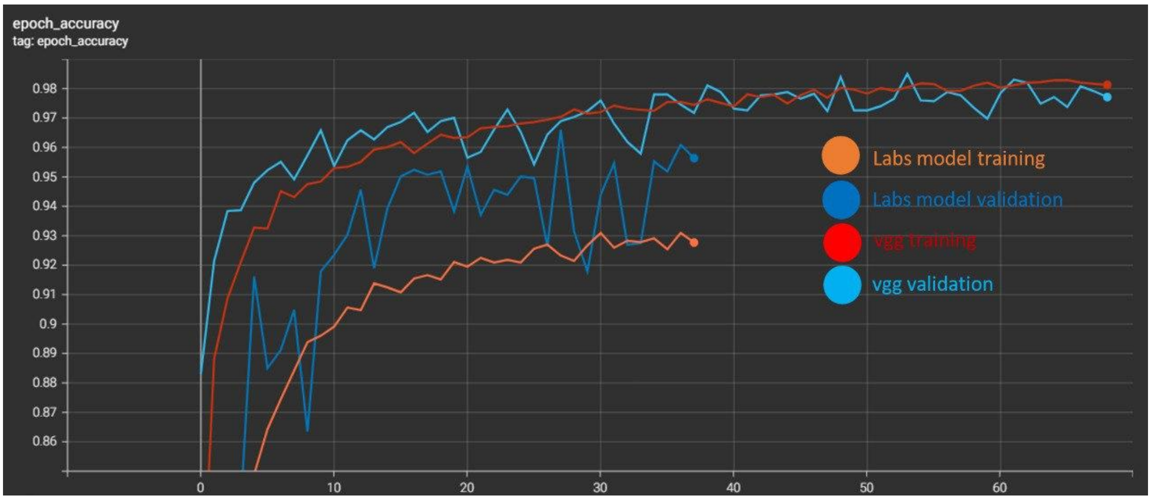

Figure 2. Training and validation accuracies of our two base models

process of both models. The plot shows that the transfer learning model achieved a better accuracy score during training than our own CNN model. Hence, in the next part of the development phase, we were focused on improving the transfer learning model.

While interpreting Figure 2 we tried to understand why the validation accuracy is greater than the training accuracy. A possible explanation is that when training, a percentage of the features are set to zero due to the Dropout layer. When testing on the validation set, all features are used. So the model at test time is more robust. Furthermore, the validation set is really small, so it is possible that its distribution does not represent the entire dataset.

Our base model for transfer learning was VGG16 with one fully connected layer (1024 neurons - number identified with the keras.tuner) and softmax layer on top. To prevent the network from overfitting, we added dropout layers with rates 0.3. Increasing the dropout layer rate to 0.4 did not provide significant improvements.

Since we did not expect the VGG16 to match our classification problem, we fine-tuned the VGG16's convolutional layers starting from the 14th layer. Lower learning rate turned out to be a crucial part of this step. As a result, we chose 1e-3 learning rate for the first part of the training (only fully connected layer) and 1e-5 for fine-tuning.

APPROACHES TO ADDRESS DATA IMBALANCE

The preliminary analysis of the dataset showed that specific solutions must be implemented to compensate for data scarcity and imbalanced classes. For dealing with data scarcity, we decided to use a pre-trained model VGG16 and to apply data augmentation as discussed in the previous section. In order to tackle the class imbalance issues, we considered three approaches: bagging in ensemble learning, class weights and a balanced validation set.

In the bagging ensemble approach, copies of the same model were trained on different balanced datasets composed by randomly undersampling all the classes to the size of the minority class (raspberry with 264 images), then the predictions were averaged. A comparison of the results obtained by using three models/datasets are plotted on Fig. 3.

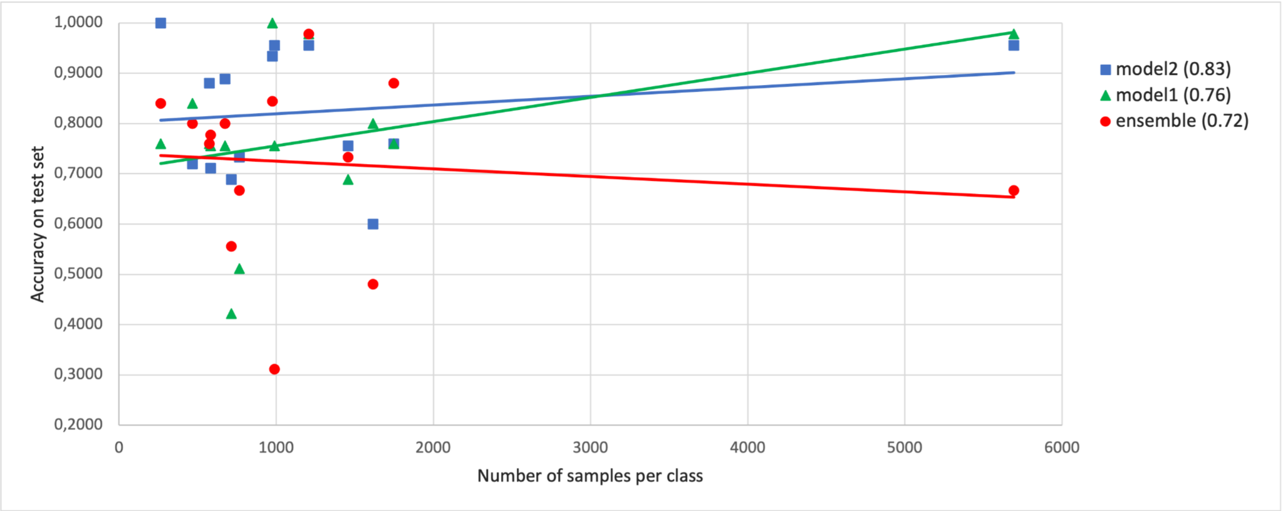

Figure 3. Effect of model selection on class imbalance

Each point indicates the accuracy score in the test set for a single class, defined by its sample size. The plot shows that the test accuracies of the ensemble (Fig. 3 “ensemble”) were indeed more balanced through the classes, i.e., the slope of the regression line was less skewed than the one obtained from another model trained on the whole dataset (Fig.3 “model1”). However, the overall accuracy of the ensemble method was lower. One of the possible explanations could be that the three datasets used didn’t include all the available samples.

For this reason, the bagging approach was abandoned in favour of class weights. The new model trained on the entire imbalanced dataset and improved with fine-tuning and class weights (Fig 3. “model2”) got a better overall accuracy score and the regression line had a smaller slope, thus reducing the imbalance.

OTHER IMPROVEMENTS

To reduce overfitting, we introduced regularization in the dense layer of the VGG16. Specifically, we used the l2 kernel and bias regularization. The performance of the model was compared between two values of regularization factors: 0.0005 and 0.001. The difference in performance between these two regularization factors was quite low — the regularization factor of 0.001 gave better results by about 0.1% on the validation set. In the end, the introduction of regularization improved the overall accuracy of the model by 1.3% on the test set.

Aiming at further improvement of the overall accuracy of the model, we prepared a notebook to utilize different transformations in the test time augmentation. A subset of 70 images found on the web was used for testing[1]. However, finding transformations that would actually improve the final accuracy turned out not to be a trivial task. Finally, a subset of transformations was found that seriously improved the accuracy (to see detailed comparison and selected transformations, see the attached notebook *TTA-tests*). The major drawback of TTA is that it significantly increases the evaluation time, and the limit of execution time on Codalab allowed us to make the final prediction using only 10 different augmentations. Finally, the TTA did not succeed in improving the test time accuracy. The accuracy with TTA was lower by about 0.006% on the final test set.

Finally, test accuracy score was improved by using LeakyReLU activation function instead of Relu to possibly avoid dead neurons, also the loss metric was substituted by F1 score to compensate for class imbalance while keeping an unbalanced validation set.

[1] A dataset with images similar to the ones provided on Codalab was found here. In order to reduce the computation time of TTA tests, only 5 images from each class were selected.