Deep Learning for Fruit Classification and Freshness Detection: A Study on Model Optimization and Explainability

Alessio De Luca
ID: 919790
University of Milan-Bicocca
a.deluca63@campus.unimib.it

Luca Pivetti
ID: 914199
University of Milan-Bicocca
l.pivetti@campus.unimib.it

***Abstract*—This paper explores deep learning techniques for fruit classification and freshness detection using CNNs. We compare a custom CNN and fine-tuned ResNet-18, applying pruning at 30%, 50%, and 70% to enhance efficiency. Additionally, we employ Explainable AI (XAI) techniques to analyze model decision-making, improving interpretability. Results highlight the trade-off between accuracy and computational cost.**

I. Introduction

In this paper, we focus on a supervised learning project based on the *Fresh and Rotten Classification* challenge on Kaggle, which involves developing machine learning models for fruit and vegetable classification. Our goal was to build a system capable of performing two tasks: first, recognize the type of fruit or vegetable represented in the image given, and second, determine whether it is fresh or rotten.

Classifying fruits and vegetables is a complex problem due to natural variations in shape, color, and texture, as well as differences in stages of ripeness. Additionally, factors such as lighting conditions and background diversity make the task more challenging. By addressing this problem, we aim to develop a model that can accurately distinguish between different types of products and assess their freshness. This has many potential applications in real life: food quality controls, automated inventory management, waste reduction systems, and so on.

II. Dataset Preparation

The Fresh and Rotten Classification dataset consists of two subsets: training and test, each containing images of fruits and vegetables labeled by category and freshness. The training set includes 23,619 samples, while the test set contains 6,738 samples. Each image is associated with two labels: a category label identifying the type of fruit or vegetable -composed by bananas, cucumbers, capsicums, oranges, tomatoes, bitter gourds, potatoes, apples, and okras- and a freshness label, where 0 indicates fresh and 1 indicates rotten. The images are taken in different environments, ensuring differences in the background conditions, and they are also resized and rotated, ensuring different viewpoints of the fruit or vegetable.

A. *Data Processing*

During the initial analysis of the dataset, we noticed that in many cases the fruits' and vegetables' names were wrongly written, and for this reason the first necessary step was to correct all these cases.

Then, we observed that the test set was missing two categories: capsicum and bittergourd. To address this issue, we reallocated 30% of the images from these categories in the training set to the test set, ensuring a more representative distribution of the classes in both sets. After this adjustment, the occurrences of these classes were lower than the others, but this has not led to major problems in the final performances. This is why we decided not to apply augmentation on them.

However, related to the general class distribution of the dataset, we noticed significant imbalance related to particular classes that were being overrepresented. To mitigate this issue, we applied a cutting strategy, limiting the number of images per class to 1,000 in the training set and 600 in the test set. This adjustment has been done working on random samples from the classes, and it made the dataset much more balanced, preventing the model from being biased toward the most frequent categories.

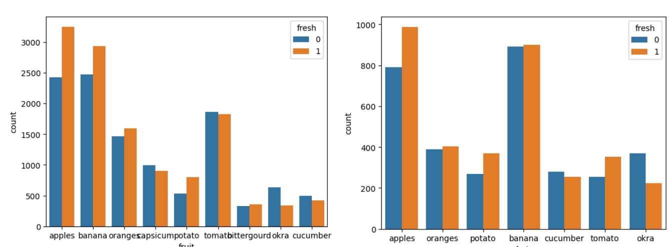

Fig. 1: Original distributions of the train and test sets.

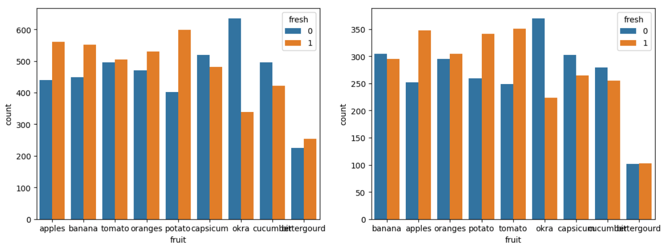

Fig. 2: Balanced train and test sets after the implementation of capsicum and bittergourd categories.

Regarding the balance between the freshness classes, the dataset was well distributed for all the types of fruits and vegetables, except for the okra's class in which fresh products (blue column) is significantly bigger than the rotten one. However, this imbalance did not leed to any particular problem.

After balancing the dataset, the training set was split into two parts: 70% for training and 30% for validation. This allowed us to properly evaluate the model's generalization capabilities during its learning in the training phase.

Finally, the creation of dataloaders for train, validation and test was done, with a Batch Size of 64 for all of them.

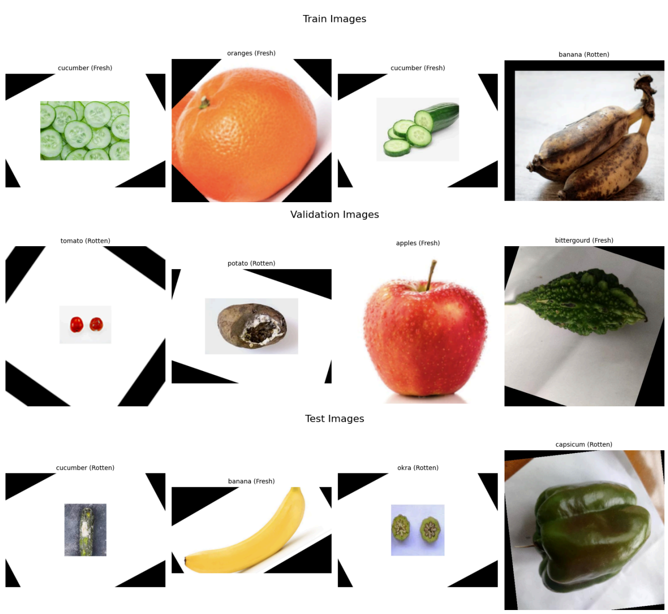

Fig. 3: Random sample images from the dataset split into training, validation and test sets.

The last necessary process performed before training the models, is the resizing of the images to a uniform size of 224x224 pixels. In this way the models will always have the same dimension of image as input. In addition to this, the intensity of the pixels was normalized to have a mean of 0 and a standard deviation of 1. This standardization helped stabilize the training process and improved model convergence.

III. Models Development

To tackle the two tasks, fruit/vegetable type classification and freshness detection, we experimented two different Convolutional Neural Network (CNN) architectures.

A. *Simple CNN Model*

The first model we developed is a simple, classical CNN structure, designed specifically for this task.

The architecture consists of three convolutional layers with max pooling, followed by an adaptive average pooling layer to reduce spatial dimensions. Then, the extracted features are passed through two fully connected layers, before branching into two separate classification heads: one for fruit/vegetable type classification, and the other for freshness detection. This first model is composed by a total of 1,638,315 trainable parameters.

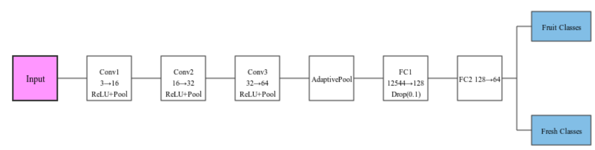

Fig. 4: Custom-made CNN structure.

B. *Fine-Tuned ResNet-18*

We also decided to try a second approach, leveraging transfer learning by fine-tuning a pre-trained ResNet-18 model [1].

We modified the final fully connected layers of ResNet-18 to extract feature representations, and then we added two separate branches: one for fruit/vegetable classification, and the other for freshness classification. Also here, similarly to the simple CNN case, each branch consisted of multiple fully connected layers with dropout regularization to prevent overfitting. Important to highlight that this model is significantly bigger than the previous one, having a total of 11,361,771 trainable parameters.

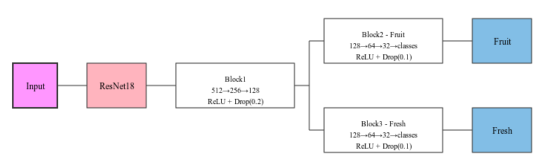

Fig. 5: Adapted ResNet-18 structure.

C. *Pruned Models*

Due to its dimensions, to further optimize the ResNet-18 based model and make it more efficient in computation and energy consumption, we implemented pruning techniques [2]. Here, the goal is to minimize the number of neurons used in the model maintaining its performances as good as possible. Specifically, we employed global L1 unstructured pruning, which identifies and removes the least important neurons in the model, based on their absolute values, that can be interpreted as utility for the final outcomes. This pruning method ensures that we reduce neurons across the entire network, rather than focusing on individual layers; in this way, the reduction in the model's size will be more efficient and won't be subject to particular constraints.

We chose to test three pruned versions of the ResNet-18 model, with 30%, 50%, and 70% of the neurons pruned. The objective is to explore how different levels of pruning can impact on the model's accuracy and efficiency, and then compare them.

IV. Training and Testing

The training and testing processes were done in a very similar way for all the models. In this way, the results can be directly compared.

A. *Simple CNN Training*

For the Simple CNN model, with randomly initialized weights, the classical Adam optimizer was used with a constant learning rate of 1e-3 and the CrossEntropyLoss as loss function for both classification types. Here, it is important to evidence that the two losses obtained were averaged and given back to the backpropagation process. This might not be the most optimal choice, but due to the simplicity of the model we decided to keep the most simple approach.

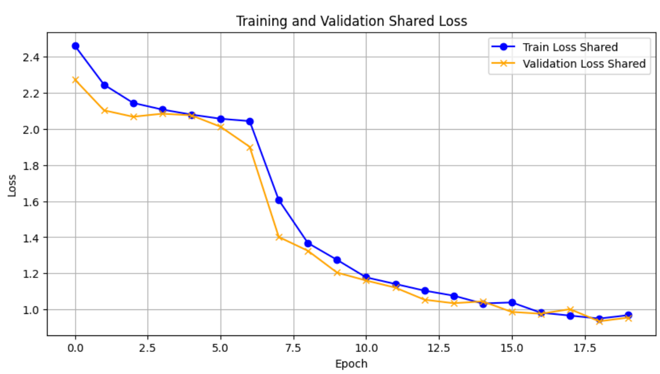

Fig. 6: Evolution of losses during Simple CNN's training and validation.

TABLE I: Simple CNN model's performances.

Ssimple CNN	Accuracy	Precision	Recall	F1-Score
Freshness metrics	76.27%	0.77	0.76	0.76
Fruit metrics	74.29%	0.78	0.76	0.75

Considering the simplicity of the model, and the double task required with a single loss given back to the whole model, we considered these results as satisfactory.

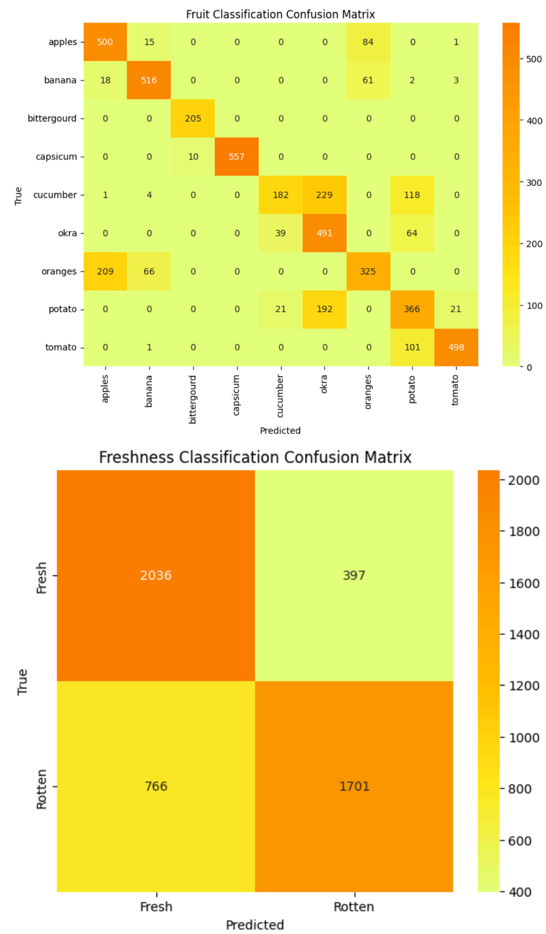

Fig. 7: Simple CNN Confusion Matrices.

However, it is also evident looking at Fig. 7 that some big issues in the knowledge of the model are present. We can notice how cucumber, oranges, and potato classes are frequently misclassified, and also how the rotten fruit/vegetable is often recognized as fresh, which could be a big problem in real use cases.

In conclusion, we can say that this first simple model is capable of learning many things, but it also highlights that a simple CNN model is not enough for a double task work with these requirements.

B. *ResNet-18 based Training*

The ResNet18-based model, on the other hand, leveraged transfer learning. This means that its weights were not randomly initialized, but were imported by a pre-trained version of the model. Only the final added layers were randomly initialized. The loss function is the CrossEntropyLoss as before, but due to this difference in the parameters initialization we decided to create three different Adam optimizers for the model's training: one for the pre-trained part, with a learning rate set at 1e-5 and taking the averaged losses, and two for the classifiers sections, both with a learning rate set at 1e-4 and each one taking the loss of its classification. In this way, the fine-tuning process should keep more knowledge from the previously trained part and adapt faster for the new classifier implemented. Here, also an early stopping technique was employed, to prevent overfitting.

It is evident that in this case the updates of the model are much more controlled, due to the much higher complexity of the model and to its starting point.

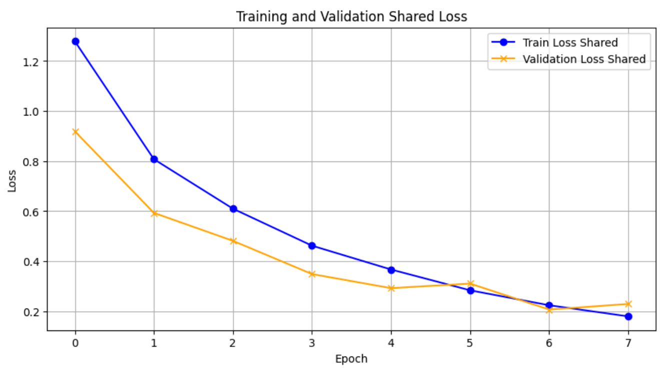

Fig. 8: Evolution of losses during ResNet-18 based model's training and validation.

TABLE II: Fine-tuned ResNet-18 based model's performances.

ResNet-18 Model	Accuracy	Precision	Recall	F1-Score
Freshness metrics	92.53%	0.93	0.92	0.93
Fruit metrics	92.24%	0.93	0.93	0.93

It is clear that this model, with its more complex structure, a previous knowledge and a much more controlled training, reached significantly better results than the previous one.

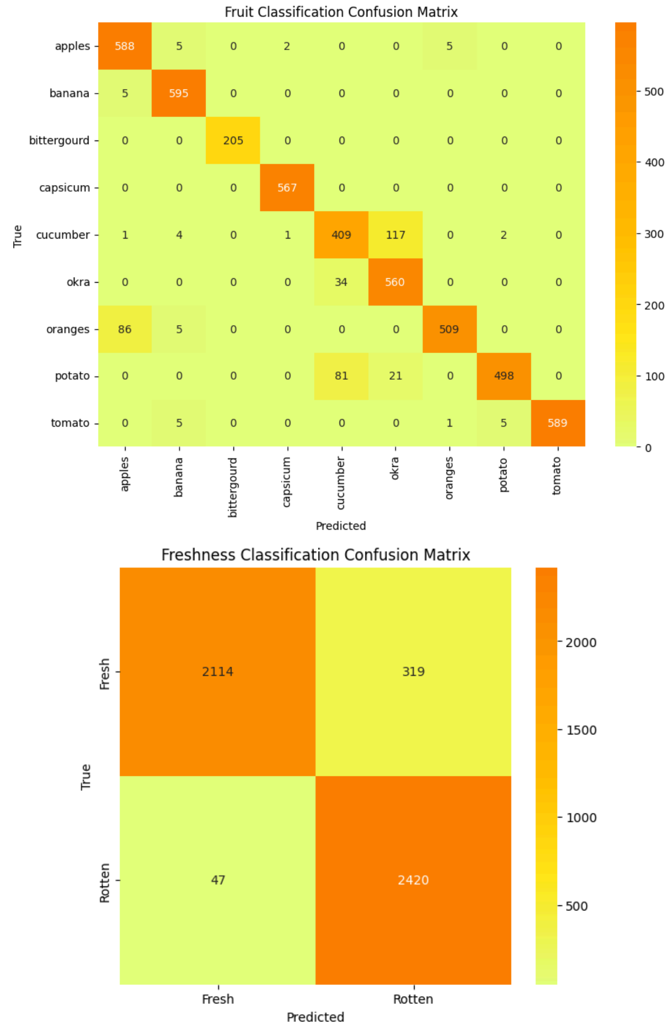

Fig. 9: ResNet-18 based model Confusion Matrices.

Fig. 9 highlights that both classifications are much better that the previous case, confirming the results obtained from training.

C. *ResNet-18 based pruned*

We decided to test pruned models both pre and post fine-tuning, to understand how much knowledge it would have lost and how much it can reacquire with respect to the original model. The following tables show the results of the three percentages of pruning applied to the ResNet-18 based model.

TABLE III: 30% pruned model's performances.

Pruned 30% Model	Accuracy	Precision	Recall	F1-Score
Freshness Pre-finetune	64.63%	0.79	0.65	0.60
Fruit Pre-finetune	51.51%	0.77	0.55	0.51
Freshness Post-finetune	96.18%	0.96	0.96	0.96
Fruit Post-finetune	98.41%	0.99	0.99	0.99

TABLE IV: 50% pruned model's performances.

Pruned 50% Model	Accuracy	Precision	Recall	F1-Score
Freshness Pre-finetune	56.98%	0.77	0.57	0.48
Fruit Pre-finetune	28.06%	0.26	0.33	0.25
Freshness Post-finetune	97.14%	0.97	0.97	0.97
Fruit Post-finetune	97.86%	0.98	0.98	0.98

TABLE V: 70% pruned model's performances.

Pruned 70% Model	Accuracy	Precision	Recall	F1-Score
Freshness Pre-finetune	56.86%	0.62	0.57	0.52
Fruit Pre-finetune	28.00%	0.24	0.33	0.25
Freshness Post-finetune	96.55%	0.97	0.97	0.97
Fruit Post-finetune	97.57%	0.98	0.98	0.98

It is clear that, even if the model was trained, the pruning techniques reduce a lot its performances if fine-tuning is not applied. But it is much more important to highlight that all the fined-tuned versions reach better performances than the originally trained model. This is probably due to the lightening, that not only makes the model faster and efficient, but it also makes its work simpler: for this reason, in the case of tasks where extreme precision and detail are not needed, the performance of simpler models can surpass that of overly complex ones.

V. Explainable AI (XAI)

Particularly in this context of fruits/vegetables and freshness identification, understanding the decision-making process of the model is essential, to understand if the system is reliable. In this work, we employed Class Activation Mapping (CAM) to visualize the regions in the input images that influenced more our models' classification outcomes [3].

We applied CAM to the last convolutional layer of both our Simple CNN and ResNet-18 based models, to analyze their decision-making processes and to compare their work. As expected, the general work is similar, but there are different approaches with pros and cons:

- Simple CNN: the model focuses on really small details in the image due to the small number of convolutional layers; however, due to this it focuses on high-contrast areas, often highlighting neglecting contextual information. For this reason, in cases where shadows or lighting variations altered the appearance of the fruit's surface, or the background contains particularly high contrasts, the model fails.

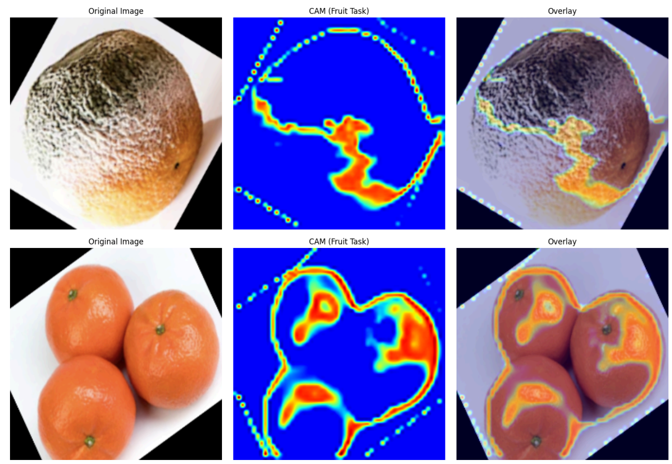

Fig. 10: CNN's Class Activation Maps on last convolutional layer.

From Fig. 10, it is evident in the upper images that the model is focused on the most appropriately lightened part of the fruit, almost ignoring the rest; in the lower images, instead, the shadow between the oranges shown is crucial for the model, that misclassifies the freshness because of it.

- ResNet-18 based model: this model demonstrates a broader focus, analyzing entire regions of the fruit rather than isolated patches. Thanks to its deeper architecture and analysis capabilities, it is able to choose often the more meaningful areas of the image to focus on, reducing misclassification errors thanks to global patterns identifications.

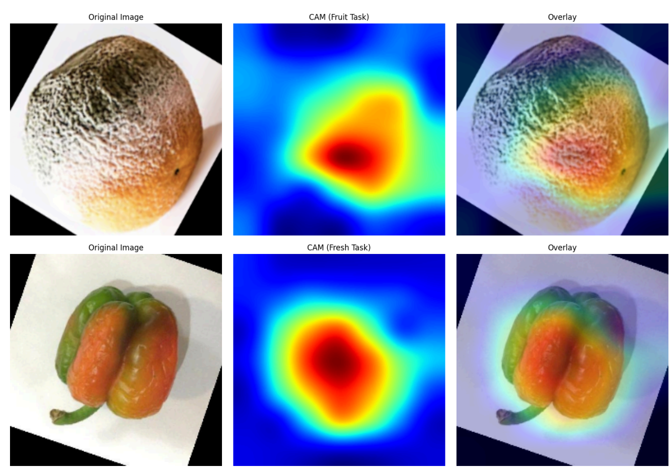

Fig. 11: ResNet-18 based model's Class Activation Maps on last convolutional layer.

From Fig. 11, it is clear that the model always focuses on the whole body of the fruit, independently by the conditions of the image and of the fruit.

VI. Conclusions

This study demonstrates the effectiveness of deep learning models in classifying both fruit types and freshness, offering valuable insights into model performance, optimization, and efficiency. The results confirm that ResNet-18 based model, when fine-tuned, outperforms a simple CNN architecture in both classification tasks. Furthermore, pruning techniques highlight the trade-off between model size and performance, revealing that pruned models can also achieve better accuracy than the not-pruned version, while significantly reducing computational complexity. These results suggest the importance of balancing model complexity and efficiency, particularly for real-time applications in food industry automation. Additionally, explainable AI techniques demonstrated their potential utility in providing insights into model decision-making process, enhancing confidence and interpretability - an important point, for food applications.

Additionally, many improvements can be done on the approach presented in this report. Improving the dataset by incorporating more diverse scenarios would enhance model robustness. Implementing an object detector could enable classification of multiple fruits within the same image. Extending the model to online applications would allow real-time analysis on live camera feeds. Additionally, integrating continual learning would enable the model to adapt to new fruit varieties and specific application conditions, ensuring long-term effectiveness.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[2] J. F. Renda Alex and M. Carbin., "Comparing rewinding and fine-tuning in neural network pruning." [Online]. Available: https://arxiv.org/abs/2003.02389

[3] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization." [Online]. Available: https://doi.org/10.1109/CVPR.2016.319