Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Fruit Recongnition - Adversarial Approach

Authors:
Cavallini Francesco (920835) - f.cavallini8@campus.unimib.it
Perfetti Luca (919835) - l.perfetti2@campus.unimib.it
Villa Fabio (907506) - f.villa93@campus.unimib.it

February 21, 2025

Abstract

In this work, the performance of different deep learning architectures for fruit and vegetable classification was explored, considering two distinct labels: the type of fruit/vegetable and its quality. The dataset used consisted of images organized into labeled folders.

As a starting point, a custom CNN model was developed and used as a baseline, against which three more advanced approaches were compared: (1) a custom Siamese model trained with triplet loss, (2) a transfer learning model based on ResNet101 pre-trained on ImageNet, and (3) a transfer learning model based on VGG19 pre-trained on ImageNet then fine tuned with a custom loss.

The models were evaluated on their ability to predict both labels using accuracy and F1-score as evaluation metrics. Finally, the three advanced approaches were compared to determine the most effective strategy for multi-label classification in the food domain, highlighting the strengths and limitations of transfer learning compared to similarity-based metric learning.

1 Introduction

Multitask learning was employed as a learning paradigm to enable a model to optimize multiple objectives simultaneously, improving generalization ability compared to networks trained separately for each task. In this work, a hierarchical classification problem was addressed, applied to a dataset of fruit and vegetable images, where each instance was labeled with two classification levels:

- **Label 1**: general category (*es. "Apple"*);
- **Label 2**: specific quality (*es. "Apple Golden"*).

The objective was to develop and compare four different neural network architectures. Initially, a custom CNN model was created and served as the baseline for comparison with three more advanced approaches: (1) a custom Siamese network trained with triplet loss, (2) a transfer learning model utilizing ResNet101 pre-trained on ImageNet, and (3) a transfer learning model based on VGG19 pre-trained on ImageNet, combined with a custom loss function. All models employed an architecture consisting of a convolutional backbone followed by a fully connected network, with two distinct output

layers for predicting the two labels (execpt for siamese network, this network's output will be detailed later).
This research was motivated by the growing interest in multitask models capable of leveraging relationships between related tasks to improve the learned representation.
To this end, the four proposed models were trained and evaluated, comparing their performance in terms of classification accuracy and generalization ability. This comparison made it possible to determine which architecture was best suited for solving multitask learning problems with hierarchical classification, using the case study of fruit (and vegetables) as a representative example. Additionally, to test generalization capability on data with a different distribution from the original set, the models were also evaluated on fruit and vegetable images randomly collected from Google, to analyze their performance in a more realistic and less controlled context.

2 Dataset

The dataset available at the following link [1] was chosen, consisting of 94,110 images (100x100x3) divided into 141 classes of fruits and vegetables. Each class's samples consists of images extracted from a video in which a single object is rotated 360° along an axis, transforming each video frame into a distinct image. This process allowed capturing each instance from multiple angles, providing a detailed representation of each fruit or vegetable. However, as will be discussed later, this acquisition method impacted the models' generalization capabilities.
The data preparation steps were as follows:

1. **In-depth dataset analysis**

2. **Construction of label2** (subclass): Initially, some subclasses that did not identify a specific variety but only a generic categorization (e.g., "Apple 6" or "Pear 2") were removed, unlike classes such as "Apple Golden," which represented an actual subclass of "Apple." Additionally, to obtain a more balanced dataset, redundant subclasses were removed, keeping only a single representation for each variety (e.g., among "Apple Golden 2" and "Apple Golden 3," only "Apple Golden 1" was retained).

However, early experiments highlighted generalization difficulties in the models, suggesting that this subdivision might reduce image diversity and limit the networks' learning ability. For this reason, the dataset was reorganized by grouping all classes with the same name but different numbering into a single category (e.g., "Apple Golden 1" and "Apple Golden 2" were merged into "Apple Golden"). Additionally, classes without a specific subclass were retained and renamed with the suffix "undefined" (e.g., "Apple 6" became "Apple Undefined"). This choice preserved greater image diversity within each class, potentially improving the models' generalization ability.

$$\text{number of classes } 141 \rightarrow 121$$

3. **Construction of label1** (primary class): All 121 classes were then grouped based on their common category (e.g., "Apple Golden," "Apple Hit," and "Apple Undefined" were unified under the class "Apple"). After this operation, the final dataset consisted of 70 main classes (label1 $\rightarrow$ category) and 121 subclasses (label2 $\rightarrow$ quality). This subdivision enabled training models in a multitask learning context, allowing them to simultaneously learn both category and quality classification.

Final number of classes

Label1 (Category)	**Label2 (Quality)**
70	121

4. **Creation of sub-sets**:

Train	**Validation**	**Test**
56,440 samples	23,619 samples	14,051 samples

5. **Data Preprocessing**: Each pre-trained model required its own distinct normalization process. To avoid incorrectly combining these different preprocessing steps on the same variables, custom models had to implement their own preprocessing, while VGG and ResNet used their respective Keras functions for preprocessing.

6. **One-hot encoding of labels y1 and y2.**

7. **Addition of external test data**: A limited set of external images, collected from Google, was later added to the original dataset and organized in a new folder. These images will be used to evaluate the models' generalization ability, providing an additional measure of their robustness in real-world scenarios.

Note: Several model learning tests were conducted on various dataset manipulations, and the reported version achieved the most satisfactory results.

3 The Methodological Approach

As anticipated, the process of model development and comparison was structured into the following phases:

- **model 0**: Baseline model. A custom CNN was initially manually defined. This was adopted as a benchmark to evaluate the performance of subsequent models, allowing for the measurement of potential improvements or deteriorations.

  The model architecture consists of an input layer of size 100×100 with three channels, followed by three convolutional layers. The first convolutional layer uses a larger kernel size (9×9) and a reduced number of filters, without padding, to capture broader patterns in the images. The next two layers adopt 3×3 kernels, a higher number of filters (128 and 256 respectively), and *same* padding, thus preserving the feature map dimensions and ensuring better feature extraction. Each convolutional layer is followed by a max pooling operation with a 2×2 window, progressively reducing the feature map dimensions to enhance computational efficiency.

  At the end of the convolutional part, a global average pooling layer transforms the extracted features into a one-dimensional array, reducing the number of parameters compared to a traditional Flatten operation and improving the model's generalization. The fully connected network consists of a single dense layer with 128 units and ReLU activation, followed by a dropout layer with a probability of 0.3 to reduce the risk of overfitting.

The architecture concludes with two separate classification heads: one for predicting the fruit category and one for quality, both using a softmax activation function for multi-class classification. The Adam optimization algorithm was used with a learning rate of 0.001.

The resulting model had a total of 442,175 trainable parameters. This configuration maintained a contained complexity compared to more advanced transfer learning models while offering a good balance between representation capacity and generalization.

This model was designed to achieve good performance while maintaining relatively short training times, thanks to its compact architecture. It is expected that its performance will be surpassed by Transfer Learning-based models, which leverage more complex and pre-trained networks on large-scale datasets.

- **model 1**: A Siamese network was implemented, using a new instance of model 0 as the backbone encoder. It was trained with triplet loss, Adam optimizer, a learning rate of 0.0005 (a low learning rate to prevent exploding gradients), and 30 training epochs. This model was chosen because it is believed that the classification problem at hand may be somewhat analogous to facial feature analysis (which is the primary use case for this type of network). By retrieving a dataset of faces, it is possible to confirm (through heatmaps generations) some similarities:
  - In both datasets, a "standard" and regular shape can be reconstructed (for faces, this corresponds to the shape of a facial mask, while for fruits/vegetables, it is a circle, given that vast majority of fruits are circular).
  - Both datasets contain large areas of high variance that help distinguish the image (in the facial dataset, these are mainly the mouth and eyes, while in the fruit dataset, this corresponds to the lower region of the fruit/vegetable).

  Given that the two problems are easily comparable, it is hypothesized that this network could be an effective tool for learning this classification problem, especially since it involves not only fruit category classification but also fruit quality classification. Since many examples of different fruit qualities resemble each other a lot, it is hypothesized

that this model will be particularly effective for learning on the second label. However, as this is a multi-label classification problem, a modified version of the triplet loss function will need to be developed. Specifically, instead of taking a triplet of images as input, this loss function takes a set of five images:

- anchor: the image used for comparison
- positive1: an image with the same label1 as the anchor image
- negative1: an image with a different label1 than the anchor image
- positive2: an image with the same label2 as the anchor image
- negative2: an image with a different label2 than the anchor image

An example of this quintuple of images is shown below:

where:

- positive1 and negative1 are very similar to each other, yet anchor and positive may be very different → the min-max problem is complex → triplet loss should work well.
- anchor, positive2 and negative2 are very similar to each other → the min-max problem is complex → triplet loss should work even better.

The final loss is calculated using the triplet-loss formula on:

$$L_1 = triplet\_loss[anchor, positive1, negative1]$$

and on:

$$L_2 = triplet\_loss[anchor, positive2, negative2]$$

Therefore, the final loss calculation, to address both min-max distance problems between positive and negative images, is given by:

$$L = L_1 + L_2$$

Since this model does not produce any output predictions, the encoder part of the network is used as a feature extractor. Then two SVMs are trained on the extracted features from the training dataset. Those will provide the actual predictions.

- **model 2**: The next step involved the use of Transfer Learning, implementing a model with ResNet101 as the convolutional backbone. The weights of the pre-trained network were frozen, while, due to computational limitations, an input size of 100x100 was chosen instead of the standard 224x224 of ResNet, accepting a possible loss in weight optimization.

  The classification part was structured similarly to the baseline model, using Global Average Pooling to obtain a one-dimensional output. However, due to the higher dimensionality of the ResNet101 output, a Dense layer with 256 neurons and Dropout at 0.3 was added, followed by a second Dense layer with 128 neurons without dropout. The two classification heads and the various activation functions remained unchanged from the baseline model.

  The resulting model has a total of 44,404,415 parameters, of which 582,079 are trainable. The strategy involved training the model for 100 epochs, using the Adam optimizer with a learning rate of 0.0001. This value was chosen to ensure stable learning, reducing fluctuations during training. Additionally, early stopping was implemented to prevent overfitting and optimize model performance.

  The expectation is that, thanks to Transfer Learning from ImageNet, the model will achieve better performance than both the baseline and the Siamese network, with improved generalization capability on the external dataset. However, this advantage comes with a longer training time due to the larger number of trainable parameters compared to the reference model.

- **model 3**: As the final implementation, a model based on VGG19 was developed with a custom loss function, defined with the aim of reducing classification inconsistencies between label1 and label2. This choice was motivated by the goal of replicating the performance obtained from the ResNet101 model. While aware of the inherent limitations of VGG19 compared to the previous model, the introduction of this custom loss

function aims to bridge that performance gap. The network architecture was implemented using a Global Average Pooling to obtain a one-dimensional output, followed by three Dense layers with 512, 256, and 128 neurons, respectively. To improve the model's generalization, two regularization techniques were applied: the insertion of dropout layers between the Dense layers and L2 regularization (with a parameter of 0.001) on all fully connected layers. The Adam optimizer was used with a learning rate of 0.0001. This architecture presents 451,519 trainable parameters out of a total of 20,024,384.

The loss function, in addition to calculating the standard loss for each label, introduces a penalty term that monitors the consistency between the predictions of the two labels. This mechanism prevents illogical situations based on the nature of the dataset, such as classifying an image as 'Apple' in label1 and 'Pear Red' in label2.

The expectation is that this implementation will provide performance comparable to or better than the ResNet101 model, while improving the consistency of classifications.

Note: Due to the large number of images (train, test, and validation totaling 90k images) and limited computational resources, a value for epochs was set as follows:

$$\frac{150\% \text{ of the entire dataset}}{\text{number of epochs}}$$

In this way, for each individual epoch, only a fraction of the dataset was visited, rather than the entire dataset being explored. This resulted in a total number of images being visited during all epochs equal to the length of the entire dataset multiplied by 1.5. This approach reduced the risk that some images would not be explored, as each batch was randomly sampled. Although adding the factor of 1.5 did not eliminate this risk, it significantly mitigated it. However, this strategy led to the possibility of fluctuations in the loss, which could increase or decrease when samples not previously seen in earlier epochs were explored for the first time. This choice helped significantly improve training times.

4 Results and Evaluation

4.1 Model 0 - CNN Custom

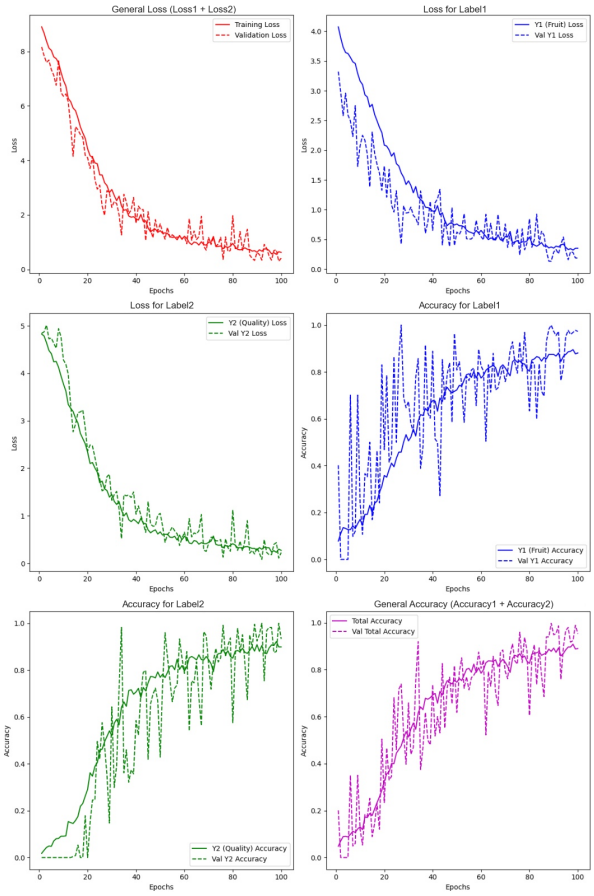

Figure 1: Graphs of the loss and overall accuracy of the base model, along with the specific metrics for each of the two labels (category and fruit quality) during training and validation.

	Accuracy (label1)	**F1-score (label1)**	**Accuracy (label2)**	**F1-score (label2)**
Custom CNN:	95.54%	was 95.16%	was 95.77%	was 95.52%

Table 1: Baseline model results

4.2 Model 1 - Siamese Network with Triplett Loss

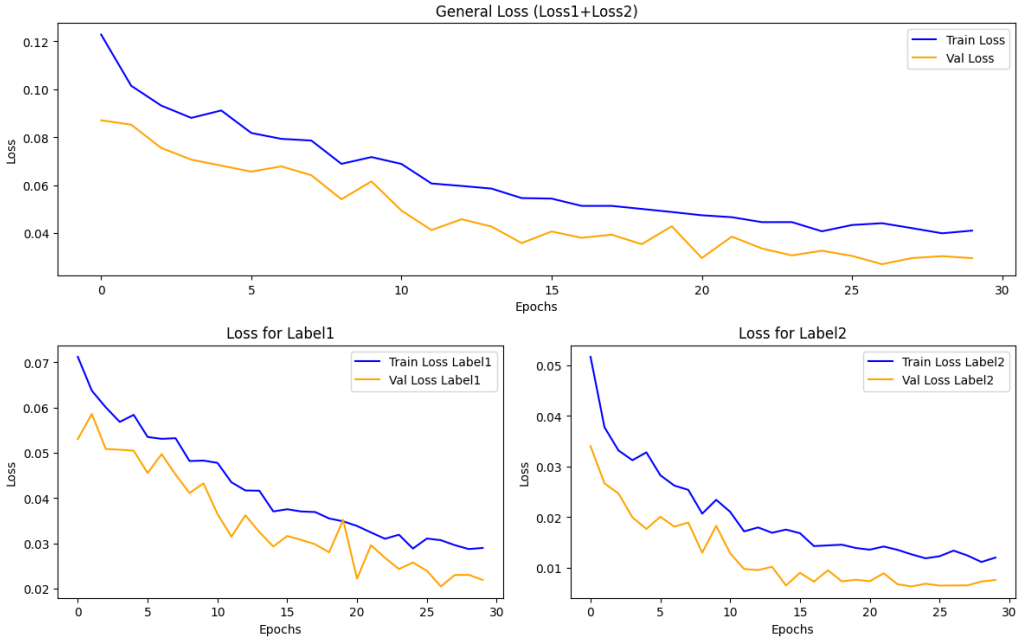

Figure 2: Total model loss.

Once the model is trained, its features are extracted, and two SVMs are trained (one for each output). This way, the *model compression* strategy is applied (using the **knowledge distillation** method), allowing for each input image to be assigned an actual prediction label.
Below are the classification results produced by this new training session.

	Accuracy (label1)	**F1-score (label1)**	**Accuracy (label2)**	**F1-score (label2)**
Modello siamese:	97.56%	97.98%	97.54%	97.57%
Baseline:	+2.02% (was 95.54%)	+2.80% (was 95.16%)	+1.77% (was 95.77%)	+2.05% (was 95.52%)

Table 2: Results of the Siamese model compared to the baseline.

4.3 Model 2 - ResNet101

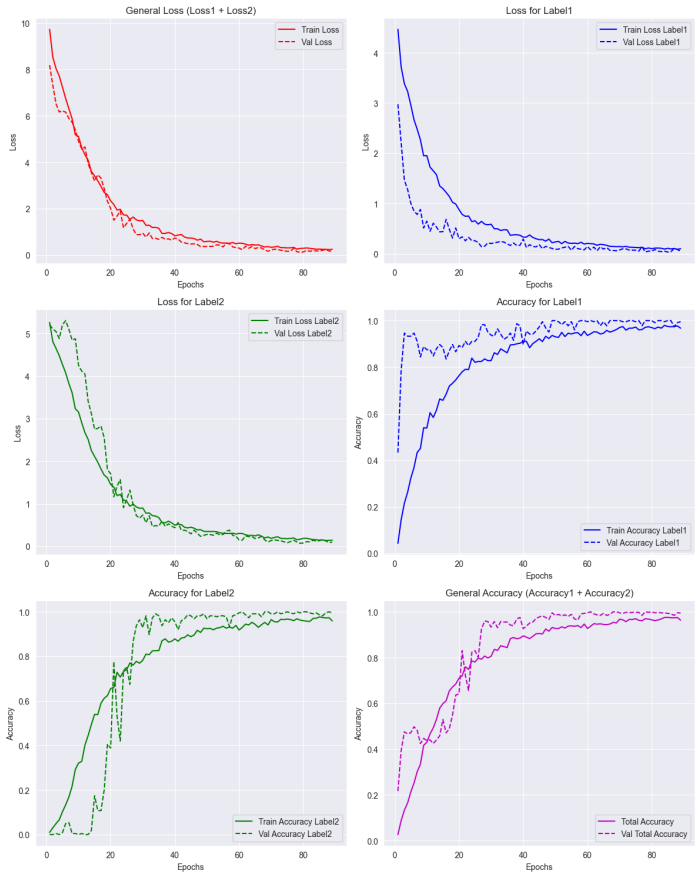

Figure 3: Graphs of the loss and overall accuracy of the ResnNet101 model based, along with the specific metrics for each of the two labels (category and fruit quality) during training and validation.

	Accuracy (label1)	**F1-score (label1)**	**Accuracy (label2)**	**F1-score (label2)**
Modello ResNet101:	99.23%	99.23%	99.18%	99.18%
Siamese:	+1.67% (was 97.56%)	+1.25% (was 97.98%)	+1.67% (was 97.54%)	+1.61% (was 97.57%)

Table 3: Results of the model based on ResNet101 compared to the previous model (Siamese Network).

4.4 Model 3 - VGG19 con Custom Loss

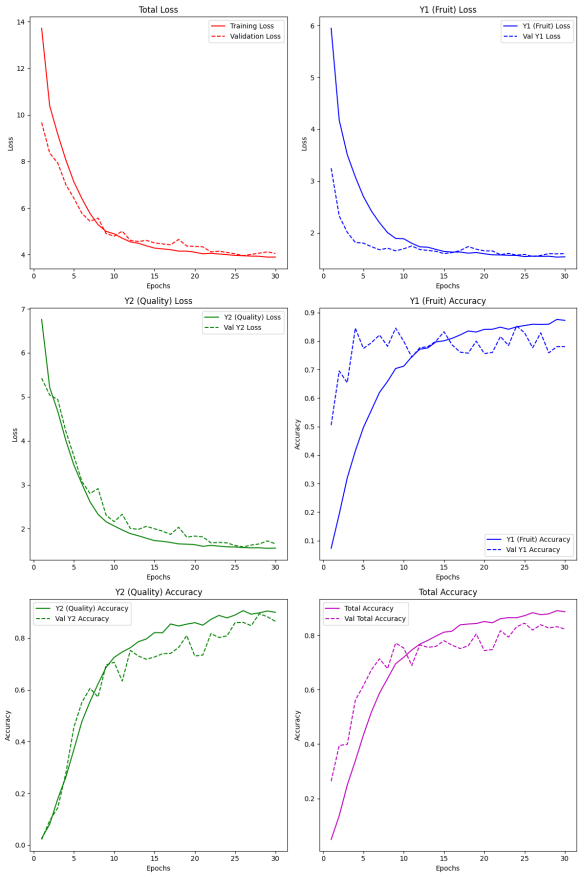

Figure 4: Graphs of the loss and overall accuracy of the VGG19 model based, along with the specific metrics for each of the two labels (category and fruit quality) during training and validation.

	Accuracy (label1)	**F1-score (label1)**	**Accuracy (label2)**	**F1-score (label2)**
Modello VGG19:	92.88%	92.51%	96.15%	95.96%
ResNet101:	-6.35% (was 99.23%)	-6.72% (was 99.23%)	-3.03% (was 99.18%)	-3.22% (was 99.18%)

Table 4: Results of the model based on VGG19 with Custom Loss compared to the previous model (ResNet101).

5 Discussion

The analysis of the learning curves for the different models showed a varied picture in terms of generalization ability and overfitting. The model based on multi-label triplet loss demonstrated good behavior during training, with a progressive reduction in loss for both labels (fruit category and quality) as well as the overall loss; it performed particularly well in training the second label, confirming the previously developed hypothesis of this model functioning well on label2. However, while effectively learning the distinctions between varieties of fruits inside the dataset, its generalization ability proved limited, with a drastic reduction in performance on the external test set (less than 20% accuracy). This suggested that the model tended to overfit the training data, reducing its effectiveness on unseen images (although this phenomenon, as mentioned before, was heavily influenced by dataset composition of multiple frames of same fruit's video).

A different behavior was observed in the ResNet101-based model, which showed high accuracy and F1-score (99%) on the test set, although this result was influenced by the nature of the data, where the training, validation, and test sets contained very similar images. However, this model significantly improved generalization compared to the previous one, achieving about 50% accuracy on an external dataset. The errors were also more understandable, with the model tending to confuse visually similar fruits rather than failing randomly. This model proved to be the most performant, in line with the initial hypotheses. Thanks to transfer learning and its deep architecture, it achieved superior results compared to the other models.

The architecture based on VGG19 with a custom loss function demonstrated decent generalization ability but suffered from overfitting, particularly in classifying the general fruit category (label1), where training accuracy continued to increase while validation accuracy stabilized. Label2, which distinguished between the quality of the same fruit, showed better learning capability with fewer signs of overfitting. Despite good final accuracies (0.92 for label1 and 0.96 for label2), the loss remained high, suggesting that the model made predictions with some degree of uncertainty. Furthermore, the comparison with ResNet101 showed that VGG19 failed to reach the same performance, likely due to architectural limitations not fully compensated for by the custom loss function.

Another point of comparison was the training time. The baseline model took about 20 minutes, while the Siamese approach with SVM required an

hour and a half. ResNet101, thanks to early stopping, completed in 36 minutes, while VGG19 took 43 minutes. It was important to note that these times were not directly comparable, as the models were trained on machines with different hardware. However, the comparison provided an indication of the relative computational costs of the different architectures, with ResNet101 maintaining a good balance between performance and training time, suggesting a good balance between efficiency and performance, while the Siamese+SVM approach proved particularly computationally expensive.

A critical issue that emerged was the limited variety of the dataset. To address this, data augmentation was applied, introducing random transformations such as flipping, brightness adjustments, slight color changes, and image off-centering. However, this strategy did not lead to significant improvements and, in the case of the VGG model, even worsened the performance. Consequently, this aspect was not explored further in the report. These results highlighted the need to expand the dataset by introducing images of different fruits rather than limiting it to variations of the same fruit rotated.

6 Conclusions

In this work, various deep learning models were developed and compared for multi-label classification of fruits and vegetables, evaluating their performance in terms of accuracy, F1-score, and generalization ability on an external test. The most significant results are presented in Table 5.

	Execution times	Accuracy	F1-score	External test accuracy	External test + Data Augmentation
Baseline model	20m	∼94%	∼95%	<20%	Slight improvement
Siamese Network	1:30h	∼97%	∼97%	<20%	Slight improvement
ResNet101	36m	∼99%	∼99%	∼50%	Same
VGG19	43m	∼95%	∼95%	∼20%	Deterioration

Table 5: Summary of the experimental results obtained.

Analysis of the results shows that the fine-tuned ResNet101 model with traditional loss was the most effective for hierarchical multi-label classification, achieving the best performance in terms of accuracy, generalization ability, and training time (using early stopping). It outperforms the custom networks by leveraging pre-trained weights from ImageNet through transfer learning, and surpasses VGG due to its deeper architecture and residual connections.

References

[1] M. Oltean, "Fruits-360 dataset," https://www.kaggle.com/moltean/fruits, 2017, accessed: 2025-02-21.