Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Fruit Image Classification: A Model Comparison Study

Authors:
Konrad Pawlik - 897194 - k.pawlik@campus.unimib.it
Jan Fiszer - 897193 - j.fiszer@campus.unimib.it

January 23, 2023

Abstract

Convolutional networks are at the core of most state-of-the-art solutions for a wide variety of computer vision tasks. In recent years, deep convolutional networks started to become mainstream, achieving substantial gains in various benchmarks. Even though, increased model size tends to immediate quality gains (as long as enough labeled data is provided), it also results in increased learning difficulties. In order to avoid these problems, the Inception architecture [1] and residual connections [2] were introduced. Here we are exploring different approaches, starting from the Transfer Learning method [3] with different pre-trained models to the compression of models using the pruning strategy [4]. Our study also includes an attempt to create our own network using the aforementioned components and adapting a previously obtained model to the multi-label output task.

1 Introduction

The goal of this project is to investigate the effectiveness of the transfer learning technique in a fruit image classification and compare the performance of different models. The three pre-trained models, VGG16 [5], ResNet152 [2] and Xception [6], were utilized in this study and the performance of a custom CNN model based on ResNet and Inception architectures was also evaluated.

The hypotheses of this study are:

- Decreasing the number of parameters in the networks will not negatively impact their performance.
- A model trained on a single-label and simple dataset will be able to accurately classify images with multiple fruits.
- The custom CNN model will achieve results comparable to those of the pre-trained VGG16, Xception and Resnet152 models.

2 Datasets

The dataset [7] contains 67692 train and 22688 test 100x100 pixel images of fruits divided into 131 classes and each class includes photos from different angles of a given fruit (figure 1). Pictures have removed the background which makes classification easier. However, two different classes may represent fruits of the same species (figure 2), so they might not be straightforward to distinguish.

Figure 1: Different angles of the "Apple golden 3" image

Figure 2: Two different classes, the same species.

2.1 Data distribution

The mostly uniform distribution allows to use accuracy as the measure metric.

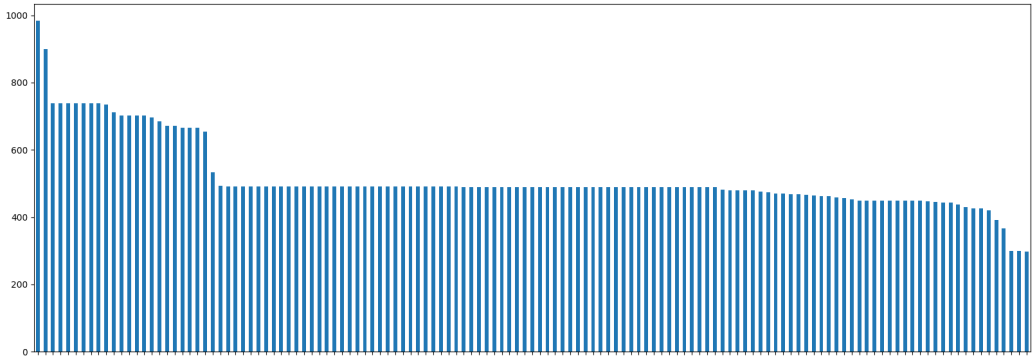


Figure 3: Distribution of all 131 classes

2.2 Data augmentation

Data augmentation using the ImageDataGenerator provided by TensorFlow [8] was applied, despite the size of the dataset. Since the objective was to classify fruits, it was important not to change the colors of the images. Moreover, it would not be right to modify the shape (small shear range). Considering all the above, these are the chosen parameters:

```
rotation_range=180,
shear_range=10,
horizontal_flip=True,
vertical_flip=True
```

3 The Methodological Approach

The main idea of this project is to explore different learning methods, evaluate and compare them. The selected techniques are the transfer learning, parameter number reduction on ResNet50 model strategy and development of its own model based on solutions from the Inception-ResNet network [9]. An additional challenge was to adapt one of the pre-trained models to the multi-label output task.

3.1 Transfer Learning

Transfer learning is a method where a saved network, that was previously trained on a large scale dataset, typically on a large-scale image-classification task, is either used as is or customized to a given task. The intuition behind transfer learning is that if a model is trained on a large and general enough dataset, it will effectively serve as a generic model for any related task. In this case, three networks were selected:

- VGG16
- Xception
- ResNet152

Each of them was loaded without the upper dense layers and then attached with a classification head adjusted to the correct dataset

```
x = Dense(256, activation='relu')(pretrained_model.output)
output = Dense(num_classes, activation='softmax')(x)
```

3.2 Parameter reduction on ResNet50

The study employs a reduced version of the ResNet50 architecture, as it is acknowledged that pre-trained models often possess an excessive number of parameters. The ResNet50 (figure 5) architecture was chosen as the experimental model, as it is a shorter variant of the ResNet152 architecture (the number in the name indicates the number of layers). Through a process of systematic layer trimming, the study ultimately arrived at the model depicted in figure 4, which was truncated after the second block.

An alternate approach to reducing the number of parameters in a neural network model is through the implementation of pruning techniques, which involve the identification and elimination of non-essential neurons and connections. The use of pruning techniques, specifically utilizing the TensorFlow Model Optimization library, was applied to a trimmed version of the

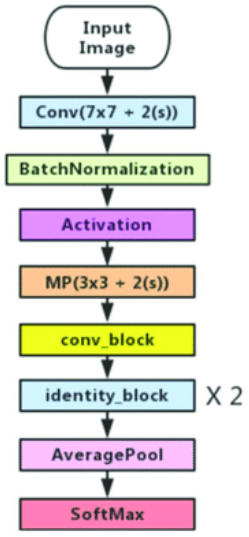


Figure 4: Trimmed after second block model

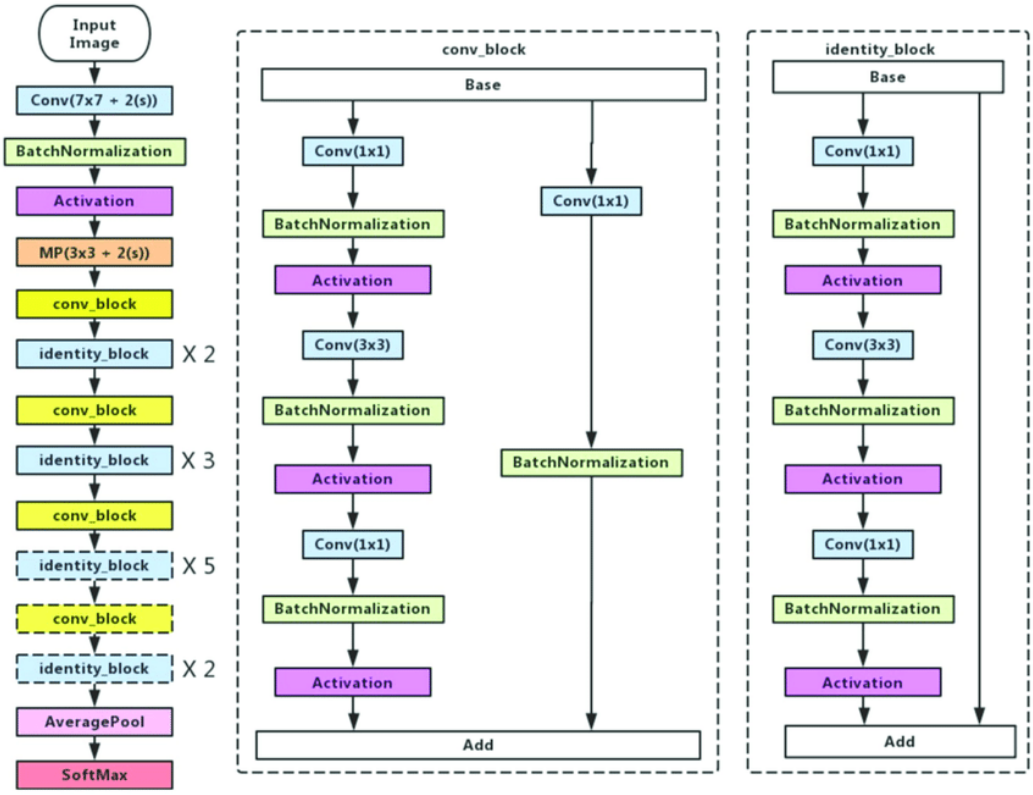


Figure 5: Full model architecture

ResNet50 model in order to reduce the number of parameters. The pruning schedule employed a Polynomial Decay function as provided by the tfmot.sparsity.keras module and was configured with hyperparameters:

```
initial_sparsity=0.6,
final_sparsity=0.9,
begin_step=0,
end_step=end_step
```

3.3 Custom Model

As both Inception and ResNet models achieve above-average results in various competitions, the idea was to combine the strengths of both networks and to develop the component including the residual connections (figure 6) and Inception blocks. Such a solution was applied in the creation of the Inception-ResNet network. In order to reduce complexity and learning time, proposed model was composed of two such components.

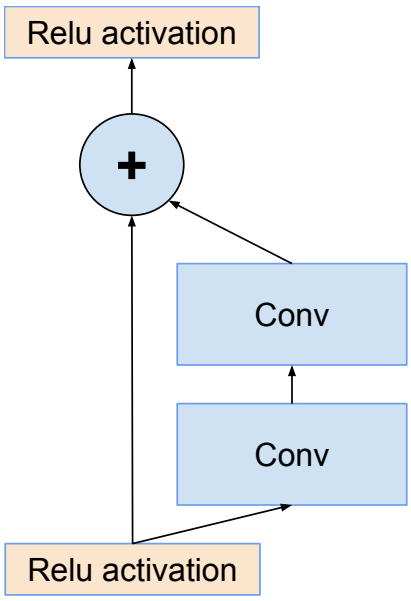


Figure 6: Residual connections as introduced in He et al. [2]

The input part of this networks consists of three convolutional and one max pooling layers.

```
x = Conv2D(32, kernel_size=3, strides=2, padding='valid', activation='relu', name='conv1')(x_input)
x = Conv2D(64, kernel_size=3, strides=1, padding='valid', activation='relu', name='conv2')(x)
x = Conv2D(128, kernel_size=3, strides=1, padding='valid', activation='relu', name='conv3')(x)
x = MaxPooling2D(pool_size=3, strides=2, padding='valid', name='maxpool1')(x)
```

The above-mentioned block has the following architecture (figure 7):

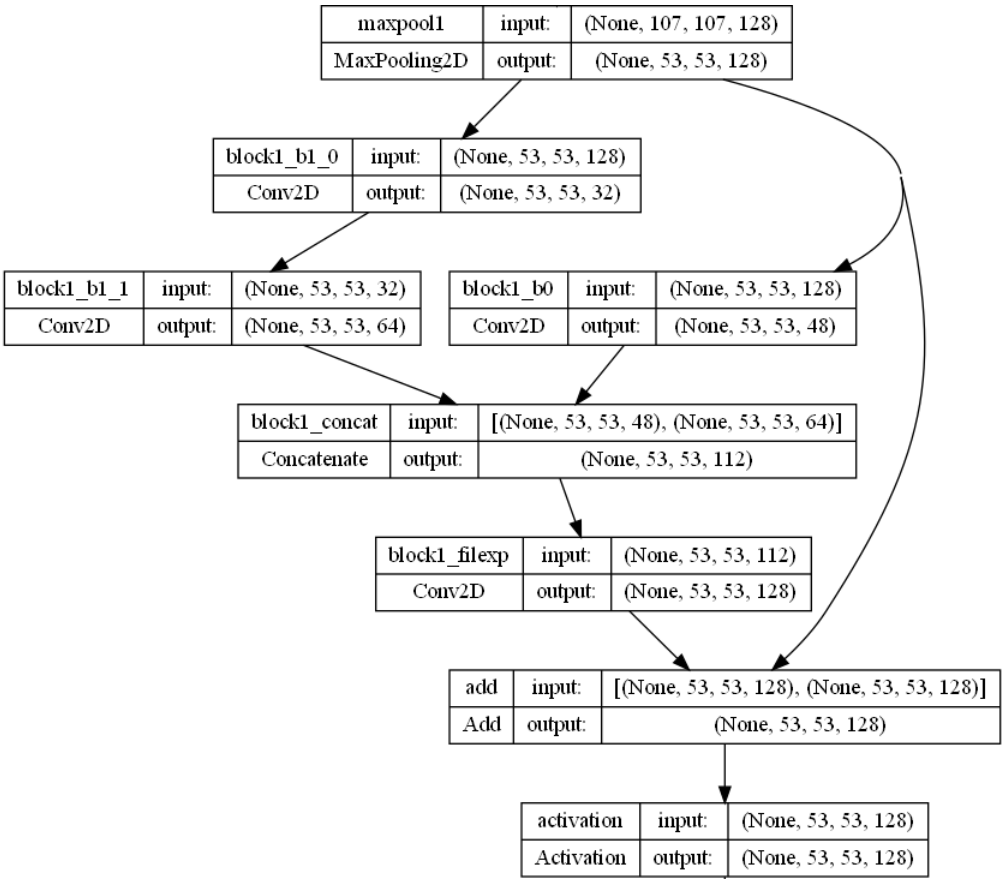


Figure 7: Custom block

Notable elements:

- Concatenate layer - Concatenates the results from two branches;
- Filter expansion layer - Scales up the dimensionality of the filter bank, followed by linear activation unit;
- Addition layer - Adds the results from stacked layers and shortcut connection.

Each block is followed by one convolutional layer and output part consists of the global average pooling, dropout and dense layers.

3.4 Classifying images with multiple fruits

In a real-world scenario, images of multiple fruits with various backgrounds are commonly encountered. To address this, except the trimmed ResNet50 model, a multi-label one was constructed utilizing the same architecture (figure 4). However, the output activation function was replaced with sigmoid, allowing the CNN to treat each class independently as if it were performing binary classification. Various, dedicated only for multi-label classification loss functions exist [10], but in this study, binary cross-entropy was employed. This is a reasonable choice, especially with the chosen loss function. The trained model produced probabilities for each individual class independently. The use of the same dataset allowed for the accuracy metric to be an appropriate measure for evaluating performance. Multiple fruit images were tested on both (single and multi label classification).

3.5 Training Methodology

In order to correctly assess the effectiveness of the individual models, specific parameters have been set for each of them:

- Optimizer - Adam [11]
- Loss function - Categorical cross entropy [12]
- Measure metrics - Accuracy

As the models selected for the transfer learning approach had only the output layer altered, the value for the epochs parameter was set to 2, as increasing it would not make a significant difference. For the model trimming, pruning and custom model learning the value of this parameter has been increased, as such action had a tangible effect on improving performance. The custom model was further extended by EarlyStopping [13] and ModelCheckpoint callbacks to limit the impact of an overfitting problem.

4 Results and Evaluation

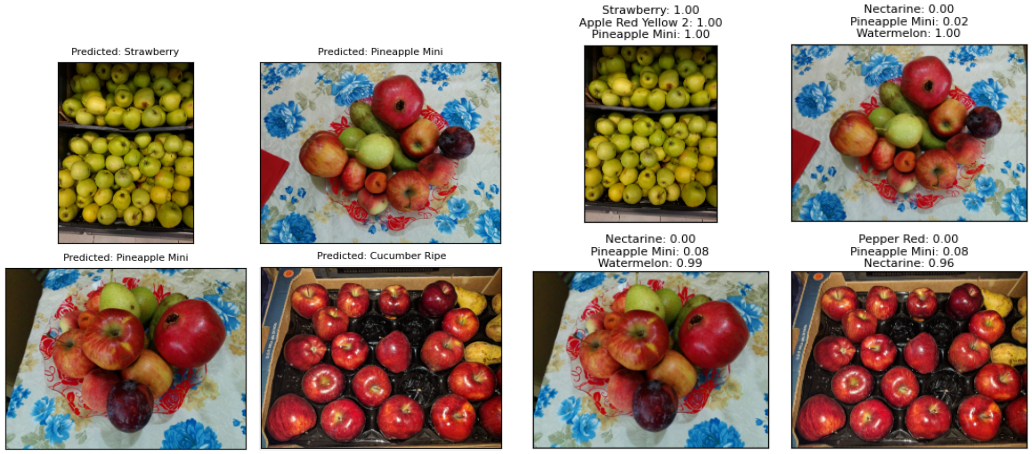
Table 4: Loss and Accuracy values for proposed models

Model	Loss	Accuracy
VGG16	0.317	0.935
Xception	0.385	0.894
ResNet152	0.122	0.965
Trimmed ResNet50	0.452	0.880
Pruned model	0.514	0.871
Custom model	0.143	0.972

All of the above mentioned models have achieved a satisfactory result, three of them managed to achieve a score higher than 0.9 in accuracy measure. Models associated with the attempt to reduce parameters, i.e. the trimmed ResNet50 and pruned model, received the lowest score, however, compared to others, notably Xception, sufficiently high. A similar relationship can be seen with the loss values.

4.1 Classifying images with multiple fruits

The performance of the proposed single-label and multi-label models was evaluated using a manually annotated dataset comprising 103 images. The results, as illustrated in figure 8, were found to be inadequate. The idea was the single-label may distinguish a individual fruit, but it demonstrated poor performance across all test cases. Similarly, the multi-label model also exhibited sub-optimal performance, with occasional overlap between predicted labels and the actual fruit names present in the images, but overall appearing random.



(a) Single-label model, expected to predict one of the fruits on image

(b) Multi-label model, with 3 most probable fruits

Figure 8: Manual predictions performed on images with multiple fruits

5 Discussion

The solutions presented coped with the fruit classification task without the slightest problem. Models with a significantly limited number of parameters were able to match much more powerful counterparts. The reason for this behaviour may be the simplicity of the task itself, or to be more precise, by working on idealised training and testing data set that do not necessarily reflect real-life cases. It is worth mentioning that the pre-trained networks were not able to show their full potential, because in their case the output part was reduced to only two dense layers.

For future multi-label improvement, the best would be to provide a dedicated dataset, including images with labels and their corresponding locations. Common for multi-label tasks CNN-RNN networks [14] would not perform effectively since the label-embedding may not work well, because the labels are for a specific group of words. Therefore, first detecting the objects from the photos and then applying the trained model will be a better solution. This approach may require the use of the object recognition techniques.

6 Conclusions

This study describes a comprehensive methodology for fruit recognition, including the following key steps:

- Transfer learning on pre-trained models VGG16, XeResNet152, and Xception
- Development of a custom CNN model, which managed to achieve the best result of all the models
- Reduction of the number of parameters in the ResNet50 model through pruning and trimming techniques, which successfully did not strongly impact the performance
- Evaluation of both single-label and multi-label models on a dataset of images containing multiple fruits, which revealed the unsuitability of the trimmed ResNet50 model in the real-case scenarios.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[3] D. Perkins and G. Salomon, "Transfer of learning," vol. 11, 07 1999.

[4] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and lt;0.5mb model size," 2016. [Online]. Available: https://arxiv.org/abs/1602.07360

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016. [Online]. Available: https://arxiv.org/abs/1610.02357

[7] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," 2017. [Online]. Available: https://arxiv.org/abs/1712.00580

[8] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané,

R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016. [Online]. Available: https://arxiv.org/abs/1602.07261

[10] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," 2013. [Online]. Available: https://arxiv.org/abs/1312.4894

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[12] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," 01 2005, pp. 561–568.

[13] L. Prechelt, "Early stopping - but when?" 03 2000.

[14] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," 2016. [Online]. Available: https://arxiv.org/abs/1604.04573