AgriEngineering

MDPI

Article

A Case Study toward Apple Cultivar Classification Using Deep Learning

Silvia Krug [1,2,*] and Tino Hutschenreuther [2]

[1] Department of Computer and Electrical Engineering, Mid Sweden University, Holmgatan 10, 851 70 Sundsvall, Sweden
[2] System Design Department, IMMS Institut für Mikroelektronik-und Mechatronik-Systeme Gemeinnützige GmbH (IMMS GmbH), Ehrenbergstraße 27, 98693 Ilmenau, Germany; tino.hutschenreuther@imms.de
* Correspondence: silvia.krug@imms.de

Abstract: Machine Learning (ML) has enabled many image-based object detection and recognition-based solutions in various fields and is the state-of-the-art method for these tasks currently. Therefore, it is of interest to apply this technique to different questions. In this paper, we explore whether it is possible to classify apple cultivars based on fruits using ML methods and images of the apple in question. The goal is to develop a tool that is able to classify the cultivar based on images that could be used in the field. This helps to draw attention to the variety and diversity in fruit growing and to contribute to its preservation. Classifying apple cultivars is a certain challenge in itself, as all apples are similar, while the variety within one class can be high. At the same time, there are potentially thousands of cultivars indicating that the task becomes more challenging when more cultivars are added to the dataset. Therefore, the first question is whether a ML approach can extract enough information to correctly classify the apples. In this paper, we focus on the technical requirements and prerequisites to verify whether ML approaches are able to fulfill this task with a limited number of cultivars as proof of concept. We apply transfer learning on popular image processing convolutional neural networks (CNNs) by retraining them on a custom apple dataset. Afterward, we analyze the classification results as well as possible problems. Our results show that apple cultivars can be classified correctly, but the system design requires some extra considerations.

Keywords: apple cultivar recognition; deep learning; challenges

Citation: Krug, S.; Hutschenreuther, T. A Case Study toward Apple Cultivar Classification Using Deep Learning. *AgriEngineering* **2023**, *5*, 814–828. https://doi.org/10.3390/agriengineering5020050

Academic Editor: Travis Esau

Received: 29 March 2023
Revised: 26 April 2023
Accepted: 27 April 2023
Published: 2 May 2023

1. Introduction

Apples are a fruit with a high genetic variation and seeds that produce new individuals with different gene combinations, leading to new varieties whenever a seed is allowed to grow and bear fruit. As a result, thousands of apple cultivars have been developed over time by selecting the most promising variants. Some of them are already forgotten or have been replaced by newer versions. Today, commercial apple farmers tend to grow a limited number of 10–25 cultivars only. Older variants still exist but are not as suitable for intensive farming. However, these cultivars are still important because they are typically well adapted to local environmental conditions and are robust without the need for pesticides. These features render it important to maintain genetic resources in order to be able to breed cultivars for the future [1,2].

In recent years, several collections of old cultivars (e.g., [3,4]) have been established all over Europe. The goal is to preserve the genetic pool for the future by collecting old cultivars and keeping them as live trees [5]. However, most of the traditional old cultivars are grown in traditional orchards with large trees. These habitats are very common all over Europe and are a vital element in biodiversity preservation [6,7]. Often, these orchards lose their purpose when fruit production shifts to specialized farming. As a result, many are abandoned or removed. In recent years, the attention of conservation of natural resources

has shifted toward these traditional orchards to revitalize the natural habitat. The apple cultivars planted there are vital to this, and thus, it is important to classify and add them to the collections. To achieve this, in-field classification by experts is frequently performed for the remaining orchards. However, the state of the trees and their potential age render the classification task difficult. Collecting probes for later comparison and analysis can be error-prone if batches become mixed or labels lost. An option for in-field assessment, similar to popular plant-classification apps, would help here.

To achieve this, it is important to classify the cultivar correctly based on images with limited equipment in the field. The challenges here are the similarities between different apples, the high number of cultivars, as well as a high variability of the apples from one cultivar. Therefore, apples are similar and variable at the same time, and small differences become very important.

Experts require 5–10 fruits at an optimal ripeness state in order to classify them mainly based on experience and so-called descriptors that represent the phenotypic characteristics of the cultivar [8]. The process involves collecting the fruit, documenting the tree in question and performing an expert analysis. At each step, mistakes and misclassifications can happen. A direct classification close to the tree would help to avoid some of them and save time.

Classifying cultivars based on fruit requires good observation skills, since only few features can separate one cultivar from another. This is another challenge because common apple cultivar descriptions are experience based and thus contain a certain amount of subjectivity and may vary from expert to expert based on their own experience. As a result, human experts can only classify cultivars they have had access to previously and were provided a proper reference. Even in collections, this is an important problem to date, as shown in [9].

Machine learning and computer vision object recognition and detection methods work in a very similar way by learning a set of features per object and by classifying it based on corresponding labels. Given enough images, the algorithm should be able to focus on relevant feature sets per class without any subjective description of the features. Thus, given the correct label and a large number of images, classifying cultivars in field based on images should be possible.

Thus far, cultivar classification has been studied by only a few authors. More often, the simpler task of classifying species is explored, and there are applications that accomplish this task. The few studies focusing on cultivar classification do not focus on expert classification as a base, and the results will be hard to generalize. The authors in [10] use only three cultivars that are easy to distinguish since these are green, red, and red with yellow variants. Using such a dataset leads to a model that is able to distinguish apples based on color but neglects that there are many green- or red-only cultivars that would be indistinguishable. Therefore, a model focusing on color only will not generalize well.

In this paper, we explore how to build a model using more stable descriptors following an expert approach by presenting a preliminary study based on transfer learning for a limited number of apple cultivars. To achieve this, we provide two main contributions in this paper: (1) the use of descriptor-based image selection for cultivar recognition and (2) the introduction of a specific preprocessing step to conserve the aspect ratio in the original image. Using these techniques, we are able to show a proof of concept for apple cultivar classification that should be able to scale with more cultivars given an appropriate number of images per class.

The remainder of the paper is organized as follows. In the next section, we review and discuss the state of the art. Afterward, we present the methodology we use in this paper and describe the preprocessing as well as the CNNs under test in detail. Section 4 highlights the results and discussion. Finally, we summarize our findings and highlight the next steps for future work.

2. Related Work

Genetic analyses have changed the taxonomy of plants and the relationship between species. However, morphological traits describing how species or cultivars look have been the major tools of botany and still remain important, e.g., to register new cultivars that differ from existing ones in one descriptor. The popularity of ML-based computer vision has introduced a new trend toward morphology or phenology [11,12] since a computer vision model based on convolutional neural networks (CNNs) depends on what it can see in the image that is similar to a classical trait analysis by experts. As a result, the approach to analyze image-based data is extended for different surveillance options [12] to monitor plants and animals.

Several ML-based tools exist for the classification of plant species (e.g., [13–15]) or other cognitive tasks such as bird call identification [16,17]. However, these approaches target species and not cultivars of one species, making the task somewhat easier, while at the same time showing the potential of ML-based approaches. The work on FloraIncognita [15] shows different challenges on how to enable a robust classification, which requires different views, of plant organs at different times of the day throughout the year. This applies to fruit cultivars as well, since we feature different weather conditions, different geographical and in-tree locations as well as ripeness degrees throughout one season. Therefore, the collected dataset needs to cover these aspects in order to allow a ML model to abstract from this kind of natural variance.

Approaches to classify cultivars have been presented recently. These include apples, tomatoes, cherries and grapes. Often, these approaches target other organs such as leaves (e.g., in [18]) rather than the actual fruit. Regarding apple differentiation, several approaches have been presented focusing on fruit or leaves in order to perform the classification. In [19], the authors present a study that is able to distinguish a limited amount of apples based on skin color. For this, the authors chose red and green variants. As a result, the classification is quite good, but it only learns two easy-to-distinguish features. Therefore, this study neglects the variance of different cultivars and does not generalize well.

The authors of [10,20,21] use a different approach. Depending on the fruit at hand, they selected a descriptive view of the cultivar and used that for their classification. In the case of apples [10], the longitudinal cut sections of apples were incorporated, in addition to texture from the outside. In the case of cherries [20] and tomatoes [21], images of the seed were used. The basic approach is the same for all three tasks. Images are prepared using a flatbed scanner and texture is extracted from the resulting images and used as a feature for different ML approaches, achieving results between 60 and 100% accuracy on the datasets. As such, the actual image is not used in the approach for classification. Furthermore, using a flatbed scanner for image acquisition is rather impractical for in-field assessments.

Other studies show promising approaches when using the leaves for classification [22,23]. Leaf images were used from different cultivars and were photographed from different angles. As such, individual leaves are represented several times in the datasets. This might lead to overfitting. However, both works give an impression on how many images are needed for similar cultivar discrimination. This and the study in [24] on apple quality assessment show that a limited amount of images can lead to good results.

In order to detect objects, other models can be used as well. In the case of fruit detection (FrD) [25] or flower detection (FlD) [26–28], special models are designed to identify and locate objects. While this involves classification, the models are not specifically designed for this task. However, these models are built to handle images from natural environments and thus provide an additional aspect needed toward a robust system for field work.

Table 1 summarizes the state-of-the-art comparison, showing that the area of cultivar classification (CC) and other ML-based questions such as various forms of quality assessments (QA) have gained high attention in the last years. When focusing on classification, typically digital cameras (DC) in studio environments (e.g., using a lightbox and tripod) [29], flatbed scanners (FS) are used for image capture to gain high-resolution images. For field work however, this is not feasible, and images captured by mobile phones (MP) would be

preferable, as shown by [27] for flowers in the natural environment. Others used public datasets (PD) or a web search to collect their data. Especially, using web-search-based data lacks proof of ground truth. This is however crucial for robust classification of the cultivar in order to use the system in conservation work later on. Related to this is the fact that there are a potentially large number of cultivars that require generalization capabilities of the model, if the model is to be extended for more cultivars over time. For this, we believe that it is important to use images that are descriptive regarding the characteristics of the apples identified by experts. What is missing are robust models that target cultivar classification, that are robust to the natural environment and that incorporate the expert approach as, e.g., described in [8], at least for a subset of plant organs. Our study is a step toward this.

Table 1. Literature comparison.

Paper	Year	Task	Object	ML Tools	Capture	Images	Cultivars	Plant Organ	Fruit Views	Expert Appr.
[19]	2012	CC	Apple	SVM	Web	90	2	Fruit	Outside	no
[14]	2018	SC	Plants	CNN	MP			Multiple	yes	
[30]	2019	CC	Apple	LDA	FS		25	Seeds		no
[18]	2020	CC	Grape	CNN	PD	300	5	Fruit	Outside	yes
[22]	2020	CC	Apple	CNN	DC	12,400	14	Leaf		no
[25]	2020	FrD	Tomato	YOLO	DC	966		Fruit	Outside	no
[26]	2020	FlD	Apple	YOLO	DC	2230	3	Flower		no
[31]	2021	CC	Apple	SVM	PD	13,000	6	Fruit	Outside	no
[10]	2021	CC	Apple	CNN	FS		3	Fruit	Outside, Cuts	no
[20]	2021	CC	Sour Cherry	CNN	FS	800	4	Seeds		no
[23]	2021	CC	Leaf	CNN	PD			Leaf		no
[32]	2022	CC	Apple	Custom DL	DC	14,400	30	Leaf		no
[29]	2022	CC	Apple	CNN	DC		9	Fruit	Outside	no
[33]	2022	QA	Mango	SVM		13,400	3	Fruit	Outside	no
[28]	2022	FlD	Kiwi	YOLO	DC			Flower		no
[34]	2022	QA	Plants	CNN		14,000		Fruit	Outside	no
[21]	2022	CC	Tomato	CNN	FS	200	5	Seeds		no
[27]	2023	FlD	Apple	YOLO	MP	3005		Flower		no
[35]	2023	CC	Apple	CNN		8538	13	Fruit	Outside	no
[36]	2023	QA	Apple	Custom DL	DC	9852	1	Fruit	Outside	no
ours	2023	CC	Apple	CNN	MP	600	5	Fruit	Inside	yes

In this paper, we will try to combine the approaches of [13,15] with state-of-the-art CNN models for image classification and focus on images acquired by mobile phones in the field using images of the apples. This approach follows a traditional expert assessment.

3. Method

3.1. Dataset Collection and Preparation

In order to evaluate the possibility to use machine learning models for apple cultivar classification, a suitable dataset is needed. Due to the high number of features that are visible in the longitudinal cut section of an apple (cf. [8]), we chose this view for our evaluation. Since there is no public dataset available featuring corresponding images, we had to collect our own. Unlike the authors of [10], we do not use apples from the supermarket but rather from trees in the surrounding areas. An expert classified each cultivar before using the images. All images of apples in this paper were captured by the

authors using mobile phone cameras to emphasize the later use of the model as part of a smartphone-based app.

The apples were collected and images acquired during the apple season 2020 from five different cultivars ("White Transparent", "Red Astrachan", "James Grieve", "Carola" and "Pinova") at different locations. In total, we collected over 100 apples per cultivar, but due to quality issues with the camera or damaged fruit, we obtained only 85 images per cultivar and 81 for "Carola" to use for the classification. This number is low, but should suffice for an initial proof. Figure 1 shows an example image from each class also highlighting the similarity of the apple cultivars form this view.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 1. Example images from each class: (**a**) White Transparent; (**b**) Red Astrachan; (**c**) James Grieve; (**d**) Carola; (**e**) Pinova.

Figure 2 shows selected images from one class only that highlight the intra-class variability as well as small defects (brown patches) on the apples. Both figures highlight the challenges very well, and the difficulty is expected to increase with the number of cultivars included in the study.

Figure 2. Example images from "Red Astrachan" class showing intra-class variability.

The images were taken with the built-in camera of different mobile phones by placing the apple in the center of the image and ensuring that it fills the screen well. While this results in different distances between the camera and object, it maximizes the number of pixels per apple. We chose this over a fixed setup with constant distance since we assume that size is a rather variable descriptor of the apple and thus is not the most important classification feature. We use the default resolution of each mobile phone, which results again in different resolutions for the base images. In order to derive the final square pixel image according to Table 2 as input to the models, we add classical image pre-processing to the dataset preparation. The following pre-processing steps are applied for each image in the dataset:

1. Apple detection and segmentation;
2. Derive bounding box around the apple;
3. Crop to square image based on the bounding box;
4. Rescale cropped image to final resolution.

The important point in this chain is to crop the image to a square bounding box, which preserves the original proportions in the image. While size is a highly variable feature of apples, shape is not. This not only applies to the apple itself but also to its organs. Therefore, we chose to buffer the image with white pixels at the edges in order to have a square image with good proportions, rather than a somewhat distorted version of it, if the rectangular bounding box becomes rescaled into a squared image. Figure 3 shows the

difference between the two options. We added this step after initial evaluations with VGG that showed a bad performance of 20 to 50%. Due to this pre-processing step, we limited the model choice to traditional classifiers in this study. Other models, such as the YOLO family, which focus on detection and classification, would remove the need for detection and segmentation, but the resulting bounding box would only be a square shape if the object is perfectly square. As a result, we would expect similar problems with distorted shape features in the YOLO classification and thus did not consider these models at this stage. However, we plan to evaluate YOLO for detection and segmentation in the future.

Table 2. Model setup.

Model	Image Size (px, px)	Weights Memory (MByte)	Parameters	Depth
VGG16	224, 224	154.2	138.4 M	16
Resnet50	224, 224	90.2	25.6 M	107
Inception	299, 299	83.9	23.9 M	189
InceptionResnetV3	299, 299	214.9	55.9 M	449
Xception	299, 299	81.8	22.9 M	81
EfficientNetB3	300, 300	47.6	12.3 M	210

(**a**) (**b**) (**c**)

Figure 3. Difference between proportion conserving cropping and rectangular bounding box rescale: (**a**) original image; (**b**) square image based on bounding box; (**c**) square crop with buffer to preserve apple proportion.

After preprocessing, the 85 images were split into 50 for training, 20 for validation, and 15 for testing. We chose two versions of building the dataset. The first one was handcrafted, where we made sure to have all combinations (ripeness state, damages, different origin) represented in each class. We chose this approach to cover as much of the variability in the dataset (see [13] for problems with variability) in order to allow the model to cover these aspects. The second variant features a classical 80–20 percent split for training and validation with k-fold random selection of the images in each set, still reserving 15 images for testing not used in the training process. In addition to this dataset, we used another extended version that provides more images per class. This dataset is used to verify what happens if more images are available.

3.2. Machine Learning Models

Several CNN models have been developed over the last 10 years and have been applied to the ImageNet dataset [37] for benchmarking. Since the focus of this study is to evaluate how these models perform for apple cultivar recognition as a proof of concept, we used existing models and transfer learning rather than designing a model from scratch. As a base, we used keras [38] with tensorflow backend [39]. This setup provides several popular CNN models with pre-trained weights from the ImageNet dataset. Among those, we selected the following models and will give a brief introduction to each one.

VGG [40] was one of the first deep convolutional models. It is known to generalize well on new datasets and is therefore a good choice for transfer learning cases. However,

VGG is a rather large model with 138.4 M parameters and thus requires a lot of memory. We used VGG16 with 16 convolutional layers in this study.

ResNet [41] enhanced the architecture of CNNs by introducing residual connections. This allows for easier training of deeper models with more layers. As a result of increased depth, the models show improved performance. Several variants of the ResNet architecture exist. We chose ResNet50 as the smallest variant, with a future mobile implementation in mind. The other variants are deeper and provide better performance but at the cost of high memory requirements.

The next model under test is InceptionV3 [42]. Inception introduces factorized convolutions and regularization to use the available resources more efficiently. The model is again deeper but has a lower number of parameters compared to ResNet and VGG.

Inception-ResNetV2 [43] combines both previous architectures. The main design goal was to speed up the training process with comparable results to Inception. This architecture is the deepest under test and shows a bigger size than the base models but is lower than VGG. We included Inception-ResNet to evaluate whether the speedup and slight performance gains are worth the additional cost in terms of memory in our case.

Xception [44] features a similar approach as Inception but replaces the inception block with depthwise separable convolutions. It has the same number of parameters as Inception but is less deep. Performance-wise, Xception is reported to slightly outperform Inception.

Finally, we consider the EfficientNet family[45] as a candidate. This architecture is newer and was built with efficiency in mind. As a result, the models are smaller, but they give good performance and have the potential to outperform ResNet and MobileNet, another model targeting mobile deployments. EfficentNet models are specifically scaled to the inputs, so that different variants exist with different input sizes and the resulting number of parameters. To stay comparable to the other models under test, we chose EfficientNetB3 with a similar input size. It is the smallest model in our study and shows similar results as Inception-ResNet.

Further models are available. This includes MobileNet, with a target on mobile applications and very low memory requirement. However, this model shows a lower performance on ImageNet and thus has been omitted as a test candidate here, since we expect the task to be rather challenging. Another option would be newer models such as Vision Transformers (ViTs). However, such models require a large amount of images for the training, which is not given in this case and will be generated only over time. Therefore, we limit our choice to the described models for now but plan to reevaluate the model selection as the dataset grows and the models become suitable for smaller datasets.

The described models focus on classification of images. Furthermore, there are models that perform object detection and classification at the same time. The YOLO family of models [46] is a popular example for this, with the original model described in [47]. Since we focus on a proof that classifying apple cultivars mimicking the expert approach is possible, we did not use these models for now. The main reason is to be able to control the detection and preprocessing to avoid errors. In addition, the YOLO family is focused on video detection, and only the tiny [48] or nano [49] models are suitable for use with mobile devices. The performance reported in [50] indicates that these models have a lower accuracy than their bigger counterparts. These model variants should however be included in future comparisons.

3.3. Model Setup

All models we considered in this study are loaded with pre-trained weights but without the original top layer. Instead, we defined a custom classifier for the given cultivars and initially trained the new classification layers before fine tuning the base network. This is the classical transfer learning approach.

We built the classifier using an AveragePooling Layer after the CNN followed by a dense layer with Relu activation with 1024 neurons, a dropout layer and the final dense layer with softMax activation as the classifier. This classifier was used for all models.

Each model uses its default image size as denoted in Table 2. The table also indicates the size of the weights and the depth and the number of parameters of the corresponding models. Depth and parameter values where taken from the Keras website (https://keras.io/api/applications, accessed on 26 March 2023).

According to this, EfficientNet shows the lowest memory requirement, while InceptionResnet and VGG show the biggest. This is relevant for future implementation within a mobile device that should be able to run the trained model for inference.

Each model is trained for up to 60 epochs with early stopping and data augmentation to mitigate the limited amount of images. We used Adam optimizer and optimize the accuracy. A checkpoint callback was used during training to save the weights of the best model variant. Later, we used the best model per training run for evaluation with the unseen test dataset.

During the training, we used data augmentation to increase the available data. Due to the importance of the apple shape, we chose only those options that preserve the aspect ratio, e.g., rotation, zooming, and shifting. To perform these augmentation steps, we used the default API provided by keras. Further augmentation options such as feature standardization or shearing are available, but they were not used in this study, to stay close to the original object. The resulting variations keep the nature of the cultivar class and are thus good options to enhance data availability.

Using our small dataset (<100 images per class) provides an additional challenge to the training process. However, collecting sufficiently large amounts of images is time consuming and might take years, especially for rather rare cultivars where only few trees are known as ground truth. Handling 100 apples is however something that is possible in one season and thus would help to extend the dataset faster and study the impact of more images over time. A similar minimum number of images is used for some cultivars in [29]. Therefore, one aspect of our study is to evaluate if it is possible to obtain initial results with such a low number of images.

3.4. Experiments

In order to evaluate if the models are able to correctly classify the apple cultivars and to explore the difficulty of the problem, we performed the following experiments:

1. Performance comparison of models. Finally, we evaluate the performance of the models and analyze differences in the predictions.
2. Impact of more images in the dataset. We performed this step to evaluate whether additional data are beneficial to the task.
3. Impact of manually constructed vs. random (k-fold validation) dataset preparation. We performed this step to investigate whether a manually constructed dataset covering all sources of variance is worth the effort.

For each experiment, we repeated the required training process five times and averaged the results. This also applied to the k-fold validation variants.

4. Results and Discussion

Regarding experiment one, Figure 4 shows the comparison of the trained CNN architectures. The boxplot is based on the best variant of the five training runs for each model on the smallest 85 image dataset. All models are able to achieve a classification accuracy of 70 % and above. This is promising for the transfer learning case, given the dataset similarity using only the longitudinal section of the apple.

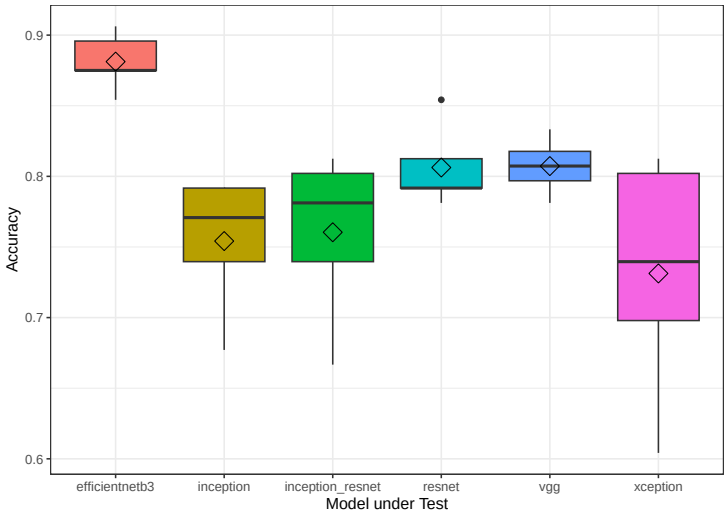

Figure 4. Statistics of the accuracy score over five training runs per model under test on the unseen test data.

Table 3 lists the top accuracy of all five runs as well as the training and evaluation time for the models. The training was performed on an Lenovo T14s notebook equipped with a NVIDIA T500 GPU. Tensorflow and keras were configured to use the GPU.

Table 3. Model performance.

Model	**Test**		**Train**		**Top Test Performance**	
	(s/epoch)	**(ms/step)**	**(s/epoch)**	**(ms/step)**	**Loss**	**Accuracy**
EfficientNetB3	7	48	7	715	0.3179	0.9062
ResNet	21	204	4	456	0.3757	0.8542
VGG	28	250	24	2000	1.4768	0.8333
Inception-ResNet	60	371	7	736	0.5154	0.8125
Xception	9	77	7	623	0.6351	0.8125
Inception	6	47	5	552	0.5592	0.7917

When analyzing the results in detail, the choice of VGG16 as a model for transfer learning can be confirmed, as expected. However, it comes at the highest cost in terms of memory usage and is the slowest model under test, resulting from the high number of calculations. The high variability of Xception results makes the transferability of performance reported in ImageNet to new tasks somewhat questionable. This came somewhat as a surprise, as it still outperforms Inception but fails to outperform ResNet or Inception-ResNet. Regrading Inception-ResNet, the improved speedup for training does not pay off in this study since the result does not outperform the other variants except for Inception and Xception, while having a high cost in terms of memory usage. According to Table 3, ResNet shows the best top result of the traditional CNNs. When observing the box plot for ResNet in Figure 4, this value is however far away from the average score of all training runs and thus has to be considered as an outlier.

The best overall performance is provided by EfficientNetB3. It shows an increased accuracy of on average 88% compared to the other models under test, for all five runs. This is important because with a future mobile application running the inference at the device, this model also has the lowest memory footprint and is among the faster models for evaluation. Furthermore, all runs show a good convergence, as can been seen for one example run in Figure 5. As a result, we chose EfficientNetB3 as the CNN architecture for the remaining experiments. In the future, we plan to compare this choice to YOLO-based variants to assess if it outperforms the combination of classical image segmentation, manual preprocessing and a relatively small EfficentNet classifier. This will involve a test of whether

the described contribution to add a shape-preserving preprocessing when preparing the images as input will be needed in that case and how this could be integrated into the YOLO architecture. For this, we plan to consider a case similar to [51], where YOLO is used for detection and segmentation only. In that setup, the resulting image can be buffered, as described here, and fed into another classifier.

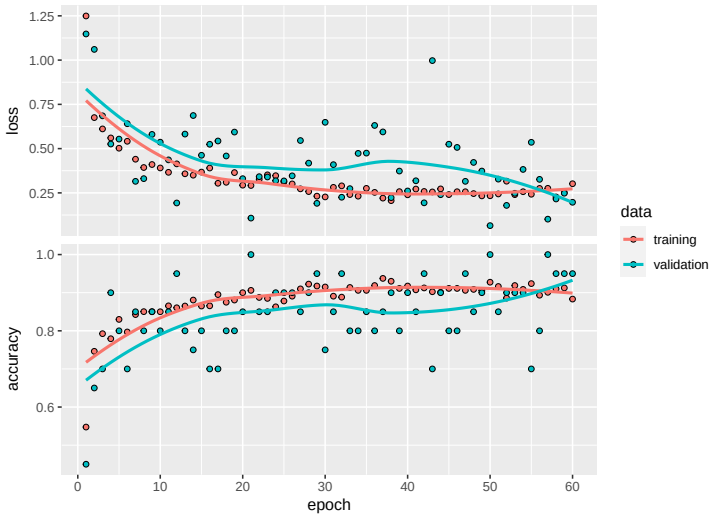

Figure 5. Example training results for EfficientNetB3 over 60 epochs.

One would expect that additional images per class render a better accuracy of the model. We attempted to verify this by adding new images from 2021 to our original 85-image dataset. However, the number of available images per class was different, and especially for the Red Astrachan, we had only a few additional images. Thus, the increase in the number of images led to a somewhat imbalanced dataset. Experiments two and three were conducted using different variants of this base amount of images. Table 4 lists the different variants and images for training, testing and validation. The per class numbers in the table represent the target. Due to imbalances, the mentioned number can be lower. This is especially true for the Red Astrachan.

Table 4. Datasets evaluating the impact of more images per class.

Variant	**Number of Images**						
		Overall			**Per Class**		
	Total	**Train**	**Valid**	**Test**	**Train**	**Valid**	**Test**
Original 85 Constructed	650	250	75	96	50	15	20
Constructed Increased Train and Test	745	554	75	116	>100	15	25
Constructed Increased Train, Valid and Test	745	529	100	116	>100	20	25
Random k-fold	744	542	135	67	>100	>26	>17

When initially increasing the number of images, we randomly selected a few images for the test data and put all other images in the training data to enhance the training process. However, the result was not as expected. Instead of an improved performance, the accuracy dropped by 2% to 86%. When looking at the resulting predictions, it became obvious that the imbalance and the increased intra-class variability caused this, as long as the validation set is untouched. Adding a subset of the new images from training to validation, the average accuracy stayed at 88%. However, the variance in the results between the five runs remains high. This variant shows the overall best model with an accuracy of 92% but also a rather low accuracy variant with 84.5%.

To evaluate this effect further, we performed a classical five-fold validation, where 20% of the images for training from each class are used for validation. The respective 20% were shifted five times. The test images were the same and were kept constant as unseen images. As a result, the accuracy dropped again and even down to 84%, which is lower than the constructed case without the increased validation set. Figure 6 shows the corresponding statistics.

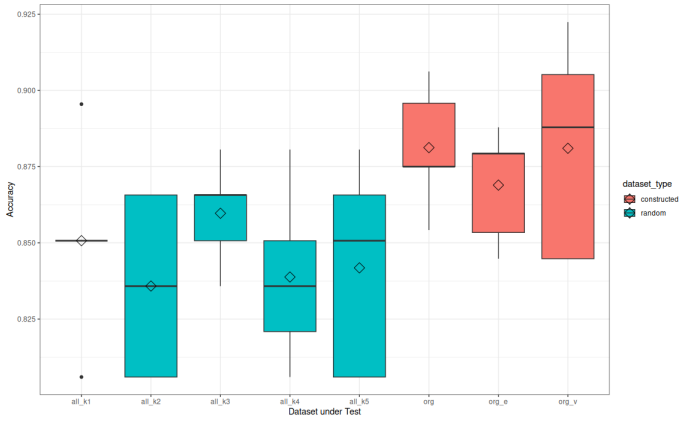

Figure 6. EfficientNetB3 performance on different dataset variants.

The random split data show worse performance compared to the constructed dataset. This shows that for the limited amount of data in this study, it is crucial to cover the complete variation within the data in all three parts of the dataset. In this case, this results in different origins of the apples, different ripeness states and especially the quality of the cut apple sections. Images where the crucial features are not visible, e.g., due to a unfocused lens or a cut that is somewhat off, are more difficult to classify, as expected.

The confusion matrix in Figure 7 for the best model highlights this. Out of the 116 images in the unseen test data, only 10 were misclassified. The most difficult classes are Carola and James Grieve, accounting over 50% of the errors. When looking at the corresponding images, it becomes clear that these are somewhat extreme cases of the respective class closer to the other classes. Thus here, we observe the impact of low inter-class separation.

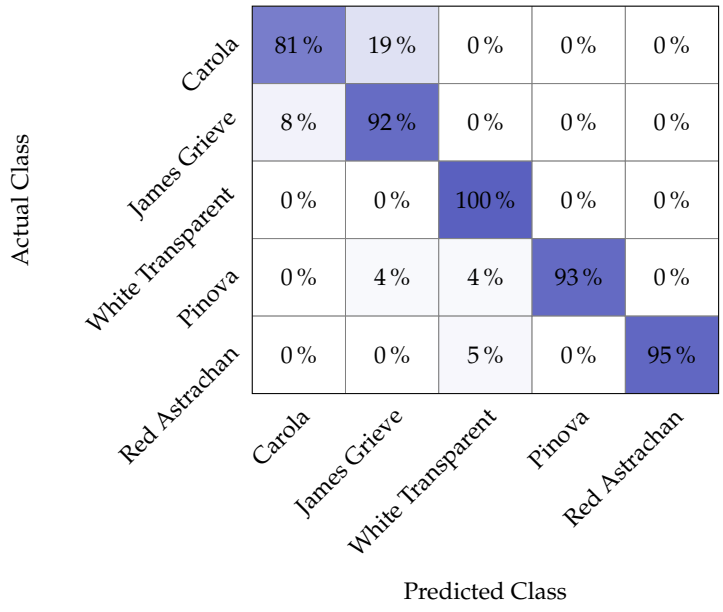

Figure 7. Confusion matrix of the best EfficientNetB3 model under test.

Figure 8 highlights this with examples of classification errors for Carola. When compared to the typical images in Figure 1, the similarity to the other classes is obvious. This is also the major challenge for apple recognition that has been the focus of research thus far. Especially toward generalizability and the addition of further classes, the observed similarity has to be handled properly. Our results show as well that models are able to obtain good prediction results even with very few images available. Toward a robust system, the goal should however be to extend the dataset with more images per cultivar.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 8. Misclassified Carola images. Images (**a**) to (**c**) were classified as James Grieve, image (**d**) as Red Astrachan and image (**e**) as Pinova

As a result, if only a limited tiny dataset is available, the system should be built with good-quality images whenever possible. Image preparation is also crucial in this case because cropping images without keeping the aspect ratio of the depicted object leads to decreased classification accuracy. This results from added variability since suddenly atypical shapes are possible, and thus, the classes become less separable.

To further increase accuracy using the given model, one could think about a majority voting where the predictions for more than one apple are combined to a final score. This would be closer to the approach of human experts. In addition, this study focused on the longitudinal cut of an apple only. While this cut shows many features of an apple, there are a number of additional features only visible from the outside of the apple [8]. Therefore, we plan to add two additional views (calxy and stem side) to the dataset to enhance the accuracy further.

Regarding the generalizability of the presented approach to classify apple cultivars, it is important to have enough good-quality images that cover the whole spectrum of expected inputs. If this is fulfilled, the approach should scale well to additional apple cultivars. This is an enhancement compared to the work of, e.g., [10,19], where the choice of image limits the generalizablilty.

5. Conclusions and Future Work

In this paper, we compared a number of well-known CNN architectures to classify apple cultivars. We used images from the longitudinal section of a cut apple only to highlight the challenges of distinguishing very similar classes with a high intra-class variability. In order to do that, we had to add a manually preprocessing step, which ensured the aspect ratio of the given apple image to preserve the crucial shape of the apple. Otherwise, the intra-class variability increases, and the task becomes unnecessarily difficult. Using the resulting images, EfficentNetB3 showed the best performance and is the smallest model under test. It is therefore a promising choice to build a mobile application for the task to classify apples in the field.

However, when extending the dataset, the results dropped somewhat due to new variability within a class and increased similarity between classes. Only the constructed datasets were able to perform well under these conditions. This shows the need to carefully construct the datasets, especially with few images per class. This is especially important for the collection and establishment of datasets for rare cultivars and can provide a guideline of, first, how many apples need to be collected and what properties to include and, second, how to build the corresponding datasets for maximum performance.

Despite the need to build a carefully constructed dataset, our findings provide promising preliminary results toward an app-based apple classification system. We were able to show that longitudinal cuts, as used by the experts, perform well even on tiny datasets and that including images according to expert knowledge is beneficial for the system. Therefore, we plan to extend the current work by evaluating how to integrate images of further apple descriptors and by studying the impact of additional cultivars as a next step. Further analyses will include YOLO for segmentation and YOLO as a benchmark for the current model. Finally, the goal is to actually provide a classification system to the conservation workers.

Author Contributions: Conceptualization, S.K. and T.H.; methodology, S.K.; software, S.K.; validation, S.K. and T.H.; formal analysis, S.K.; investigation, S.K.; resources, S.K.; data curation, S.K.; writing—original draft preparation, S.K.; writing—review and editing, S.K. and T.H.; visualization, S.K.; supervision, S.K.; project administration, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Szot, I.; Goncharovska, I.; Klymenko, S.; Bulakh, P. Importance of Old and Local Apple Cultivars. *Agrobiodivers. Improv. Nutr. Health Life Qual.* **2022**, *6*, 156–170 [CrossRef]
2. Tripodi, P.; Cocozza, A. Harnessing Crop Diversity through Genetics, Genomics and Phenomics Approaches. *Plants* **2023**, *12*, 1685. [CrossRef] [PubMed]
3. Kellerhals, M.; Szalatnay, D.; Hunziker, K.; Duffy, B.; Nybom, H.; Ahmadi-Afzadi, M.; Höfer, M.; Richter, K.; Lateur, M. European pome fruit genetic resources evaluated for disease resistance. *Trees* **2012**, *26*, 179–189. [CrossRef]
4. Flachowsky, H.; Höfer, M. Die Deutsche Genbank Obst, ein dezentrales Netzwerk zur nachhaltigen Erhaltung genetischer Ressourcen bei Obst. *J. Kult.-J. Cultiv. Plants* **2010**, *62*, 9.
5. Hanke, M.; Höfer, M.; Flachowsky, H.; Peil, A. Fruit genetic resources management: Collection, conservation, evaluation and utilization in Germany. In Proceedings of the I International Symposium on Fruit Culture and Its Traditional Knowledge along Silk Road Countries (Acta Horticulturae 1032), Tbilisi, Georgia, Yerevan, Armenia, 4–8 November 2013; pp. 231–234.
6. Zerbe, S. A Century of Practice and Experiences of the Restoration of Land-Use Types and Ecosystems. In *Restoration of Multifunctional Cultural Landscapes: Merging Tradition and Innovation for a Sustainable Future*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 313–380.
7. Zerbe, S. Traditional Agroforestry Systems. In *Restoration of Ecosystems–Bridging Nature and Humans: A Transdisciplinary Approach*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 409–418.
8. Höfer, M.; Eldin Ali, M.A.M.S.; Sellmann, J.; Peil, A. Phenotypic evaluation and characterization of a collection of Malus species. *Genet. Resour. Crop Evol.* **2014**, *61*, 943–964. [CrossRef]
9. Reim, S.; Schiffler, J.; Braun-Lüllemann, A.; Schuster, M.; Flachowsky, H.; Höfer, M. Genetic and Pomological Determination of the Trueness-to-Type of Sweet Cherry Cultivars in the German National Fruit Genebank. *Plants* **2023**, *12*, 205. [CrossRef]
10. Ropelewska, E. The application of image processing for cultivar discrimination of apples based on texture features of the skin, longitudinal section and cross-section. *Eur. Food Res. Technol.* **2021**, *247*, 1319–1331. [CrossRef]
11. Christodoulou, M.D.; Clark, J.Y.; Culham, A. The Cinderella discipline: Morphometrics and their use in botanical classification. *Bot. J. Linn. Soc.* **2020**, *194*, 385–396. [CrossRef]
12. Katal, N.; Rzanny, M.; Mäder, P.; Wäldchen, J. Deep learning in plant phenological research: A systematic literature review. *Front. Plant Sci.* **2022**, *13*, 805738. [CrossRef]
13. Wäldchen, J.; Rzanny, M.; Seeland, M.; Mäder, P. Automated plant species identification—Trends and future directions. *PLoS Comput. Biol.* **2018**, *14*, e1005993. [CrossRef]
14. Wäldchen, J.; Mäder, P. Machine learning for image based species identification. *Methods Ecol. Evol.* **2018**, *9*, 2216–2225. [CrossRef]
15. Mäder, P.; Boho, D.; Rzanny, M.; Seeland, M.; Wittich, H.C.; Deggelmann, A.; Wäldchen, J. The Flora Incognita app–interactive plant species identification. *Methods Ecol. Evol.* **2021**, *12*, 1335–1342. [CrossRef]
16. Kahl, S.; Wilhelm-Stein, T.; Klinck, H.; Kowerko, D.; Eibl, M. Recognizing birds from sound-the 2018 BirdCLEF baseline system. *arXiv* **2018**, arXiv:1804.07177.
17. Kahl, S.; Wood, C.M.; Eibl, M.; Klinck, H. BirdNET: A deep learning solution for avian diversity monitoring. *Ecol. Inform.* **2021**, *61*, 101236. [CrossRef]
18. Franczyk, B.; Hernes, M.; Kozierkiewicz, A.; Kozina, A.; Pietranik, M.; Roemer, I.; Schieck, M. Deep learning for grape variety recognition. *Procedia Comput. Sci.* **2020**, *176*, 1211–1220. [CrossRef]

19. Suresha, M.; Shilpa, N.; Soumya, B. Apples grading based on SVM classifier. *Int. J. Comput. Appl.* **2012**, *975*, 8878.
20. Ropelewska, E.; Sabanci, K.; Aslan, M.F. Discriminative power of geometric parameters of different cultivars of sour cherry pits determined using machine learning. *Agriculture* **2021**, *11*, 1212. [CrossRef]
21. Ropelewska, E.; Piecko, J. Discrimination of tomato seeds belonging to different cultivars using machine learning. *Eur. Food Res. Technol.* **2022**, *248*, 685–705. [CrossRef]
22. Liu, C.; Han, J.; Chen, B.; Mao, J.; Xue, Z.; Li, S. A novel identification method for apple (Malus domestica Borkh.) cultivars based on a deep convolutional neural network with leaf image input. *Symmetry* **2020**, *12*, 217. [CrossRef]
23. Zhang, Y. Improved Leaf Image Classification Using Topological Features and CNN With Attention Module. In Proceedings of the 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 22–24 October 2021; pp. 311–315.
24. Sun, L.; Liang, K.; Song, Y.; Wang, Y. An Improved CNN-Based Apple Appearance Quality Classification Method With Small Samples. *IEEE Access* **2021**, *9*, 68054–68065. [CrossRef]
25. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* **2020**, *20*, 2145. [CrossRef] [PubMed]
26. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [CrossRef]
27. Shang, Y.; Xu, X.; Jiao, Y.; Wang, Z.; Hua, Z.; Song, H. Using lightweight deep learning algorithm for real-time detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2023**, *207*, 107765. [CrossRef]
28. Li, G.; Suo, R.; Zhao, G.; Gao, C.; Fu, L.; Shi, F.; Dhupia, J.; Li, R.; Cui, Y. Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. *Comput. Electron. Agric.* **2022**, *193*, 106641. [CrossRef]
29. García Cortés, S.; Menéndez Díaz, A.; Oliveira Prendes, J.A.; Bello García, A. Transfer Learning with Convolutional Neural Networks for Cider Apple Varieties Classification. *Agronomy* **2022**, *12*, 2856. [CrossRef]
30. Sau, S.; Ucchesu, M.; D'hallewin, G.; Bacchetta, G. Potential use of seed morpho-colourimetric analysis for Sardinian apple cultivar characterisation. *Comput. Electron. Agric.* **2019**, *162*, 373–379. [CrossRef]
31. Bhargava, A.; Bansal, A. Classification and grading of multiple varieties of apple fruit. *Food Anal. Methods* **2021**, *14*, 1359–1368. [CrossRef]
32. Chen, J.; Han, J.; Liu, C.; Wang, Y.; Shen, H.; Li, L. A Deep-Learning Method for the Classification of Apple Varieties via Leaf Images from Different Growth Periods in Natural Environment. *Symmetry* **2022**, *14*, 1671. [CrossRef]
33. Gururaj, N.; Vinod, V.; Vijayakumar, K. Deep grading of mangoes using Convolutional Neural Network and Computer Vision. *Multimed. Tools Appl.* **2022**, 1–26. [CrossRef]
34. Parashar, N.; Mishra, A.; Mishra, Y. Fruits Classification and Grading Using VGG-16 Approach. In Proceedings of the International Conference on Communication and Artificial Intelligence: ICCAI 2021, Mathura, India, 13–16 April 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 379–387.
35. Yu, F.; Lu, T.; Xue, C. Deep Learning-Based Intelligent Apple Variety Classification System and Model Interpretability Analysis. *Foods* **2023**, *12*, 885. [CrossRef]
36. Zhang, L.; Hao, Q.; Cao, J. Attention-Based Fine-Grained Lightweight Architecture for Fuji Apple Maturity Classification in an Open-World Orchard Environment. *Agriculture* **2023**, *13*, 228. [CrossRef]
37. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
38. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 26 March 2023).
39. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 26 March 2023).
40. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
43. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
44. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
45. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
46. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
47. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA 27–30 June 2016; pp. 779–788.

48. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694. [CrossRef]
49. Wong, A.; Famuori, M.; Shafiee, M.J.; Li, F.; Chwyl, B.; Chung, J. YOLO nano: A highly compact you only look once convolutional neural network for object detection. In Proceedings of the 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS), Vancouver, BC, Canada, 13 December 2019; pp. 22–25.
50. Sharma, A. *Training the YOLOv5 Object Detector on a Custom Dataset*; Chakraborty, D., Chugh, P., Gosthipaty, A.R., Huot, S., Kidriavsteva, K., Raha, R., Thanki, A., Eds.; PyImageSearch: Philadelphia, PA, USA, 2022. Available online: https://pyimg.co/fq0a3 (accessed on 26 March 2023).
51. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef]