BAVARIAN JOURNAL OF APPLIED SCIENCES®

A Comparison of Convolutional Neural Networks and Feature-Based Machine Learning Methods for the Ripeness Classification of Strawberries

Leon Binder* **Michael Scholz*** **Roman-David Kulko***

ABSTRACT

A variety of machine learning methods are often used for ripeness detection of fruits and vegetables using image data. Existing research in this area often focuses only on training feature-based classifiers or on using raw images with convolutional neural networks. The purpose of this paper is to compare both approaches in terms of their classification accuracy. To answer our research question, we analyze the performance of convolutional neural networks and different feature-based classifiers on a balanced dataset consisting of three strawberry ripeness classes: unripe, ripe, and overripe. Our investigation shows that convolutional neural networks outperform almost all feature-based classifier. However, the penalized multinomial regression achieves the best accuracy of 86.27 % without any hyper-parameter tuning. Another insight is that different methods lead to the best sensitivity for different ripeness classes. Convolutional neural networks most accurately classify unripe strawberries, while ripe strawberries are best classified by penalized discriminant analysis and overripe berries are best classified by penalized multinomial regression.

Für die Reifedetektion von Obst und Gemüse anhand von Bilddaten werden häufig verschiedene Methoden des maschinellen Lernens eingesetzt. Bestehende Forschung in diesem Bereich konzentriert sich oft nur auf das Training von merkmalsbasierten Klassifikatoren oder auf die Verwendung von Rohbildern mit Convolutional Neural Networks. Ziel dieser Publikation ist es, beide Ansätze in Bezug auf ihre Klassifikationsgenauigkeit zu vergleichen. Um unsere Forschungsfrage zu beantworten, analysieren wir die Leistung von Convolutional Neural Networks und verschiedenen merkmalsbasierten Klassifikatoren auf einem balancierten Datensatz, der aus drei Reifeklassen von Erdbeeren besteht: unreif, reif und überreif. Unsere Untersuchung zeigt, dass Convolutional Neural Networks fast alle merkmalsbasierten Klassifikatoren übertreffen. Die penalisierte multinomiale Regression erreicht jedoch die beste Genauigkeit von 86,27 % ohne jegliches Hyper-Parameter-Tuning. Eine weitere Erkenntnis ist, dass unterschiedliche Methoden zur besten Genauigkeit für unterschiedliche Reifeklassen führen. Convolutional Neural Networks klassifizieren unreife Erdbeeren am genauesten, während reife Erdbeeren am besten durch die penalisierte Diskriminanzanalyse und überreife Erdbeeren am besten durch die penalisierte multinomiale Regression klassifiziert werden.

KEYWORDS

Ripeness classification, Computer Vision, Machine Learning

Reifegradklassifizierung, Computer Vision, Maschinelles Lernen

* Technology Campus Grafenau, Deggendorf Institute of Technology

1. Introduction

One of the many areas where computer vision has become increasingly important in recent years is agriculture. Several applications such as crop health monitoring [1], growth stage detection [2], and automatic crop harvesting (e.g., [3, 4]) are only possible due to tremendous progresses in computer vision. Traditional machine learning methods, such as linear regression, have often proven unable to adapt to an ever-changing complex environment. For example, automatic fruit harvesting requires computer vision methods to locate the fruits, to identify the ripeness level of the fruits and to decide whether the ripe fruits are accessible to the harvesting robot [3].

Each of these tasks aroused a number of studies to develop and identify appropriate computer vision and machine learning methods. For example, the problem of identifying the ripeness level of fruits is a classification problem that can be addressed with traditional supervised learning algorithms such as k-nearest neighbors [5], decision trees [6], support vector machines [7], naïve Bayes classifier [8], but also with modern computer vision methods such as convolutional neural networks (e.g., [9, 10]). However, due to the no-free-lunch theorem [11], none of the methods dominates the other methods in terms of accuracy. Recent research shows that the dimensionality and size of the training dataset, as well as the characteristics of the data, determine which method is appropriate for a classification task [12]. A major advantage of convolutional neural networks is their ability to integrate classification and feature extraction. Ripeness of fruits can be detected visually based on features such as color, shape or texture. Identifying features that represent these characteristics and are able to discern between different ripeness levels is a very difficult task. For example, fruit color can be represented by several statistical features such as mean, median, standard deviation, kurtosis, skewness, minimum, maximum or mode that can be computed also in different color spaces such as RGB, HSV or Lab. Convolutional neural networks, in contrast, only take the pixel matrix of an image and extract the information that is necessary for a good classification from this matrix through different layers.

As the number of characteristics that change at different stages of fruit ripeness is rather small and some simple statistical features can often describe these characteristics, it is questionable whether convolutional neural networks can really use the integration of feature extraction and classification to their advantage. Past research primarily focused exclusively on the usage of traditional methods or convolutional neural networks. We contribute to existing literature on automatic fruit ripeness prediction by comparing several methods from these two approaches for strawberry classification based on image data to find out if CNNs lead to performance improvements over traditional methods.

The remainder of this paper is organized as follows. In Section 2, we review relevant literature on ripeness classification and classification methods. In Section 3, we present the data and methodology used to compare the different classification methods. The results of this comparison are presented in Section 4. We conclude this paper in Section 5 with a discussion of the implications of our results and limitations of our study.

2. Related Work

Fruit ripeness classification based on image data has been addressed in many studies (e.g., [8, 13, 10]). These studies differ in the fruits studied, the classifiers and the ripeness levels that are investigated. However, most studies focus only on traditional machine learning methods such as Naïve Bayes (NB), decision trees (DT), k-nearest neighbors (KNN) or support vector machines (SVM).

Mazen et al. collected 300 images of green, yellowish green, mid-ripe and overripe bananas and converted them to HSV color space [8]. After removing the background, a ripeness factor was defined as the ratio of brown pixels out of all banana pixels. Texture features to quantify contrast, coarseness and direction were also extracted. SVM, NB, KNN, DT, discriminant analysis (DA) and fully connected neural networks (FNN) were trained as classifiers. With an accuracy of 97.75 % FNN was found to be the most accurate classifier.

Indrabayu et al. developed a prototype for strawberry grading and sorting [14]. Strawberries were placed on a conveyor belt and continuously photographed. For each of the three

ripeness classes unripe, partially ripe, and ripe 100 images were collected and converted from RGB to HSV color space. After detecting and cropping the images to the relevant strawberry area, the mean values for all three color channels were extracted. An SVM with a radial basis kernel was trained as classifier and achieved an accuracy of 85.64 %. Interestingly, their classifier reached a sensitivity of over 96 % for ripe and unripe strawberries, but a sensitivity of only 62.94 % for partially ripe strawberries. This might be due to leaves covering the strawberries or background pixels still being present in the cropped images.

Also Castro et al. used traditional machine learning methods to automatically classify the ripeness stage of fruits [13]. They trained four classifiers (FNN, SVM, DT, and KNN) on three different color spaces (RGB, HSV, LAB) to classify cape gooseberries into seven ripeness stages. After collecting 925 images, the authors extracted the mean values of each color dimension in the color spaces. FNN, SVM, KNN perform better in the LAB color space, while DT achieves the highest classification accuracy in the HSV color space. The authors demonstrate that dimension reduction with principal component analysis increases the accuracy of all classifiers. Overall, SVM proved to be the best classifier with an accuracy of 93.02 %.

A few studies investigated the performance of convolutional neural networks (CNN) for fruit ripeness classification.

Zhang et al., for example, trained a CNN to classify bananas into seven predefined ripeness stages [10]. They collected 17,312 images over a 14-day period and sorted them into seven and twelve ripening stages based on their date of imaging. The proposed CNN consisted of three convolutional and max-pooling layers, followed by two fully connected layers. The authors reached a classification accuracy of 94.4 % and 92.4 % with respect to seven and twelve ripeness stages, respectively. Several misclassifications were due to severe defects such as too many black spots. However, the model still achieved precision and recall above 90 %. The trained CNN achieved a higher accuracy in fine-grained ripeness classification than state-of-the-art machine learning methods based on SVM.

Sustika et al. explored different CNN architectures (AlexNet, MobileNet, GoogLeNet, VGGNet, Xception) for strawberry quality classification [9]. In total, the authors collected 1,870 images of strawberries. The images were manually assigned to four quality levels. The first three levels were distinctions of strawberries with good quality and the fourth level described strawberries with bad quality (i.e., overripe, damaged, or rotten strawberries). Models using different CNN architectures were trained and evaluated for their performance in a binary classification of strawberries (i.e., good and bad quality) and a classification into the four quality levels. The model trained with the VGGNet architecture gave the highest performance in both binary classification and rank classification with an accuracy of 96.49 % and 89.12 %, respectively.

We contribute to these studies by comparing traditional machine learning methods that require explicit feature extraction to convolutional neural networks that implicitly extract features. The next section describes the experimental setting used for this comparison.

3. Experimental Setting

We compare traditional machine learning methods to convolutional neural networks for classifying strawberries into one of the following three degrees of ripeness:

- Unripe: Firm strawberries with a greenish or whitish coloration in some parts on the visible side.
- Ripe: Slightly soft strawberries with an even deep red coloration on the visible side.
- Overripe: Strawberries with bruises and very soft spots on the visible side.

In Section 3.1, we describe the data collection and ground-truth labelling process. Section 3.2 summarizes the applied training algorithms and Section 3.3 describes the steps applied to the images to prepare them for classification. Section 3.4 presents the workflow and setup for training and evaluating the different classifiers.

3.1. Data

For our experiments we use unripe, ripe and overripe strawberries of the same cultivar. Photos of unripe and ripe strawberries are taken immediately after buying the strawberries. Some of the ripe strawberries are stored at 8°C for three days to produce overripe strawberries. All strawberries are photographed under controlled conditions. Each strawberry is put on a black painted toothpick and photographed from at least four sides. We use a black background to facilitate extraction of the strawberries from the background during data processing. We collect a total of 666 images.

All strawberry images are classified independently by three persons into one of the three classes – unripe, ripe and overripe. We finally form the ground truth as the majority vote of the three individual votes for each strawberry. Fleiss's Kappa between the three raters is 0.649 indicating substantial reliability of agreement according to Landis and Koch [15]. Furthermore, Cohen's Kappa shows a moderate reliability between raters A1 and A2 (0.5934) and between A2 and A3 (0.5501). Inter-rater reliability is with a Cohen's Kappa of 0.6490 substantial. The resulting ripeness level distribution is shown in Table 1.

	Unripe	Ripe	Overripe	Total
Number of Images	184	243	239	666
Percentage	27.63 %	36.49 %	35.89 %	100.00 %

Table 1: Ripeness level distribution

3.2. Classification Methods

Table *2* lists the classification methods with their abbreviations that we use in the following sections. We also note the input format for each method. Except for CNN, which uses raw pixel data, the classifiers are based on statistical features such as the mean of color dimensions.

Abbreviation	Method	Input Format
KNN	k-Nearest Neighbors	Statistical Figures
C5	C5.0 Decision Tree	Statistical Figures
NB	Naïve Bayes	Statistical Figures
PDA	Penalized Discriminant Analysis	Statistical Figures
PMR	Penalized Multinomial Regression	Statistical Figures
FNN	Fully Connected Neural Network	Statistical Figures
CNN	Convolutional Neural Network	Pixels

Table 2: Methods

3.3. Data Preparation

First, we prepare the raw images to i) improve the classification accuracy and ii) make the images usable for convolutional neural networks. Images are taken in raw format (CR2) with a resolution of 5,472 x 3,648 pixels. Images at this size are too large for convolutional neural networks even when trained on dedicated GPUs. The original images also contain several data related to the background, which is meaningless for the classification task. With data preparation, we aim to remove background information and downscale the images. Specifically, data preparation consists of the following four steps:

1. Loading: We load an image from the raw image format.

2. Removing the background: Although we use a black background and the strawberries are put on a black toothpick, the pixels that do not represent the strawberry do not represent pure black. This is due to some light reflections as well as strawberry juice running down the toothpick. We thus remove background data to avoid the background and especially the strawberry juice on the toothpick from affecting the classification. Specifically, we first convert the image from RGB to LAB color space. We use the B-dimension (blue-yellow) to identify background pixels. Morphological transformations and median blur are necessary to avoid losing dark pixels within the strawberry itself. The resulting threshold

mask is a 2-dimensional matrix with Boolean values that indicates for each pixel in the image whether it is part of the background or foreground. We then apply this mask on the image so that all background pixels in the resulting image are transformed to transparent pixel.

3. Cropping and resizing: The next step is to crop the image so that the strawberry fills the whole image. This results in different image sizes as the strawberries are of different sizes. Since neural networks require a fixed structure, we resize the cropped images consistently to an image resolution of 512 x 512 pixel.

4. Saving: At the end of the data preparation pipeline, each image is saved in PNG format.

Figure 1 displays the effect of the preprocessing pipeline for an example image. Furthermore, Figure 2 shows one example image for each of the three ripeness classes.

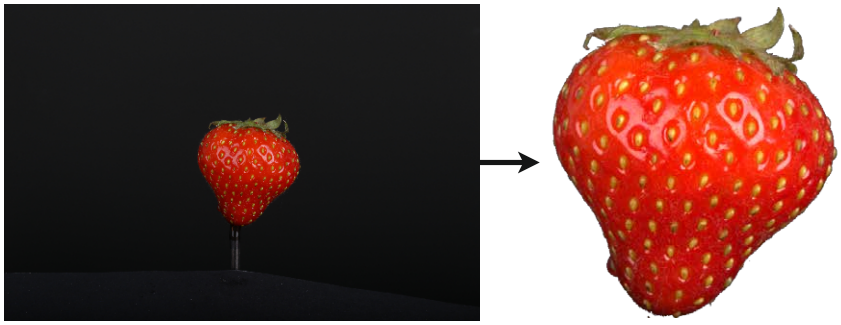

Figure 1: Example of an original and preprocessed image

unripe ripe overripe

Figure 2: Examples for unripe, ripe, and overripe strawberries after preprocessing

Traditional machine learning methods and fully connected neural networks operate with manual features. We thus extract five statistical features from each image: the mean, standard deviation, median, kurtosis and skewness of the three corresponding channels in RGB, HSV and LAB color spaces. Using statistical features instead of pixel data involves a loss of information. However, as the number of statistical features is rather marginal compared to the number of pixel-based features, the use of statistical features is accompanied by a significant reduction of model complexity and hence a substantial reduction in training time. Table 3 shows the extracted features for the three example images from Figure 2.

		RGB			HSV			LAB		
Class	**Statistic**	**R**	**G**	**B**	**H**	**S**	**V**	**L**	**A**	**B**
Unripe	mean	0.73	0.31	0.13	22.85	0.81	0.73	149.82	52.37	15.20
	sd	0.19	0.15	0.12	26.81	0.17	0.19	50.64	16.19	7.41
	median	0.74	0.30	0.10	15.54	0.86	0.74	144.93	51.52	17.28
	kurtosis	-0.31	-0.66	0.99	93.98	-0.43	-0.31	0.08	-0.16	-1.04
	skewness	-0.38	0.40	1.17	7.93	-0.83	-0.37	0.55	0.15	-0.48
Ripe	mean	0.78	0.24	0.21	215.72	0.78	0.79	149.36	34.93	18.40
	sd	0.16	0.22	0.16	164.35	0.19	0.16	70.39	14.96	7.61
	median	0.80	0.16	0.16	351.27	0.84	0.80	127.97	31.58	19.97
	kurtosis	0.90	-0.64	1.10	-1.85	-0.47	0.90	-0.25	1.24	-0.58
	skewness	-0.83	0.78	1.25	-0.35	-0.82	-0.83	0.84	0.96	-0.58
Overripe	mean	0.84	0.30	0.33	256.63	0.69	0.84	183.03	24.86	17.73
	sd	0.16	0.24	0.19	151.24	0.21	0.16	83.03	13.08	6.65
	median	0.87	0.24	0.28	349.90	0.72	0.87	164.41	23.01	18.66
	kurtosis	1.56	0.18	1.49	-1.03	0.23	1.56	0.29	2.99	-0.43
	skewness	-1.13	0.92	1.24	-0.97	-0.73	-1.13	0.87	1.13	-0.51

Table 3: Extracted features from the three example images in Figure 2

One insight regarding the three example strawberries from Table 3 is that with increasing ripeness, the statistics of the B dimension in the RGB color space also increase. The increase in mean and median can be associated with the increasingly darker shade of red. The tendency can also be seen in the mean, median and standard deviations of the hue (H), with a big increase from the unripe to the ripe strawberry image. While the unripe strawberry has a more warm-reddish hue, the ripe and overripe strawberry become increasingly magenta in color.

3.4. Model Building & Evaluation

3.4.1. Traditional Methods

The models based on the traditional methods are trained with the caret package in the R programming language. We experiment with different data preprocessing techniques and finally use mean-std-standardization for NB, PDA and PMR, min-max-normalization for KNN, and no preprocessing for DT.

We divide the data into 80 % for training and validation and 20 % for estimating the test accuracy of the traditional classification methods (KNN, C5, NB, PDA, PMR). We perform a 10-fold cross validation for computing the training, validation and test accuracy. For the traditional methods, we decide not to tune hyperparameters in order to keep the training effort as low as possible.

3.4.2. Fully Connected & Convolutional Neural Networks

Neural networks have the ability to learn complex relationships between inputs and the corresponding output. However, they have the disadvantage that many hyperparameters need to be tuned to achieve at least moderate results. We used Python 3.6 with Tensorflow 1.14 for hyperparameter tuning, training, validation and testing of the fully connected neural networks and the convolutional neural networks.

Our CNNs require feature-wise centering and normalization of the standard deviation of the images. We therefore extracted the mean and standard deviations of the color channels in the training data and used these values to normalize the images in the training, the validation and testing sets.

It is also noticeable that CNNs tend to overfit easily. Therefore, we artificially increased the amount of training data with data augmentation. We applied the following four augmentation steps to the images: First, we rotated images between 0 and 30 degrees. Second, we shifted the images horizontally and vertically between 0 and 30 %. Third, we randomly zoomed into the images with a zoom factor between 0 and 10 %. And fourth, we randomly flipped images horizontally. Figure 3 shows some examples for augmented images. The augmentation methods do not change the ground-truth of the images but ensure that the classification models are trained on a greater variety of data.

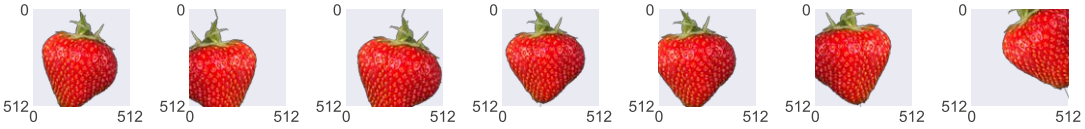

Figure 3: Examples of the transformations in the data augmentation step

FNNs mostly consist of Dense layers, with the possibility of intermediate Dropout layers to prevent overfitting. CNNs, on the other hand, consist of a juxtaposition of Convolution and Pooling layers, that enable automatic feature extraction. The feature extraction layers are followed by one Flatten layer and at least one Dense layer. A selection of layer-specific hyper-parameters is shown in Table 4.

Layer type	**Hyper-Parameters**
Dense	Number of units, activation function, bias, initializer / regularizer / constraint
Convolution	Number of filters, kernel size, activation function, strides, padding, dilation, bias, initializer / regularizer / constraint
Pooling	Pooling size, padding size, strides
Flatten	
Dropout	Percentage

Table 4: Layer-specific hyper-parameters

For the FNN and CNN we also split the data in 80 % for training and validation and 20 % for hyperparameter tuning and model accuracy testing. The parameters are tuned based on the same data splits. We computed the test accuracy of the five sets of parameters for FNN and CNN leading to the highest validation accuracy. The test accuracy is estimated based on a 10-fold cross validation.

4. Results

In this section, we summarize the results of our experiments. We first present the results for the classification accuracy of all investigated methods. Thereafter, we focus on the FNN and CNN and discuss the effect of training epochs and filter layers on the classification accuracy.

	Accuracy					
	Training		**Validation**		**Test**	
	Mean	**Sd**	**Mean**	**Sd**	**Mean**	**SD**
KNN	0.7977	0.0090	0.8194	0.0842	0.8351	0.0206
C5	0.7769	0.0073	0.7633	0.0539	0.7828	0.0206
NB	0.8048	0.0066	0.8084	0.0647	0.7746	0.0172
PDA	0.8381	0.0060	0.8441	0.0476	0.8216	0.0042
PMR	0.8594	0.0094	0.8875	0.0477	**0.8627**	0.0072
FNN	0.8897	0.0166	0.8760	0.0469	0.8239	0.0146
CNN	0.8477	0.0193	0.8704	0.0460	0.8373	0.0227

Table 5: Accuracy of the different methods based on cross-validation

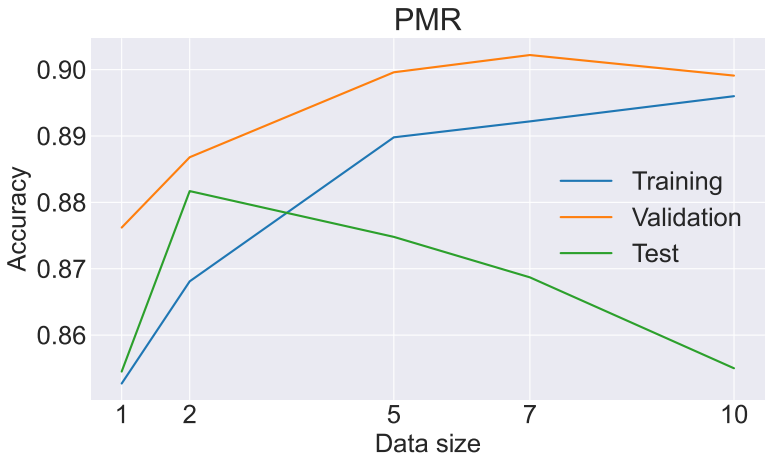

Figure 4: Accuracy of the PMR with increasing data size via data augmentation

We also investigated the effect of data augmentation on the performance of the traditional methods. Figure 4 illustrates, as an example, the training, validation and test accuracy for the Penalized Multinomial Regression (PMR) and different augmentation factors that lead to a 2-, 5-, 7- and 10-times larger data size. It is apparent that data augmentation likely causes the problem of overfitting. This pattern has been found for all methods except C5, where the data augmentation leads to a performance improvement of over 7%.

Table 5 indicates that CNN does not lead to the best performance regarding the strawberry images and that researchers should particularly consider using PMR for this classification task due to a higher accuracy and a lower computational effort. However, the results in Table 5 refer to the total classification accuracy in a balanced dataset. We also computed the accuracy and number of misclassifications for each class separately in order to get some further insights about the performance of the tested classification methods. Therefore, we computed the mean classification sensitivities as well as the mean number of correct and incorrect predictions for each class using a 10 cross-validation. The results are depicted in Table 6.

Method	Class Unripe				Class Ripe				Class Overripe			
	Prediction				Prediction				Prediction			
	Recall	U	R	O	Recall	U	R	O	Recall	U	R	O
KNN	0.8027	29.7	6.5	0.8	0.8633	1.5	42.3	5.2	0.8312	1.9	6.2	39.9
C5.0	0.7757	28.7	7.3	1.0	0.7735	2.5	37.9	8.6	0.7979	1.4	8.3	38.3
NB	0.8054	29.8	7.1	0.1	0.7959	2.4	39.0	7.6	0.7292	2.0	11.0	35.0
PDA	0.7324	27.1	9.9	0.0	**0.9184**	0.0	45.0	4.0	0.7917	0.0	10.0	38.0
PMR	0.8541	31.6	4.2	1.2	0.8837	1.0	43.3	4.7	**0.8479**	1.0	6.3	40.7
FNN	0.8459	31.3	5.7	0.0	0.8000	6.5	39.2	3.3	0.8312	0.0	8.1	39.9
CNN	**0.8595**	31.8	5.2	0.0	0.8796	2.5	43.1	3.4	0.7771	0.1	10.6	37.3

Table 6: Mean sensitivity and classification count for classes of the models in Table 5

The unripe strawberries were very rarely or never falsely classified as overripe by the classification methods. The same applies to the overripe strawberries. This indicates that unripe and overripe strawberries can be well separated from each other. Table 6 also demonstrates that different classification methods perform best in predicting strawberries of the three ripeness classes. CNN most correctly classified unripe strawberries, whereas ripe strawberries were best identified by a PDA and overripe berries were most correctly classified by a PMR. This implies that our proposed CNN should be used in situations where most strawberries are unripe, and ripe and overripe berries need to be sorted out.

4.1. Effect of Training Epochs

In the following, we present the effect of the number of training epochs on classification accuracy for one of the FNN and one of the CNN models trained during cross validation. We trained an FNN consisting of 5 dense layers with a total of 8,418 weights. The first four hidden layers have 45 neurons each and use sigmoid activation functions. The last layer includes three output neurons corresponding to the three ripeness classes. The model was trained for 200 epochs using the Adam optimizing algorithm, categorical cross entropy as loss function and a training batch size of 16. Before training, we applied a min-max-normalization to the input features.

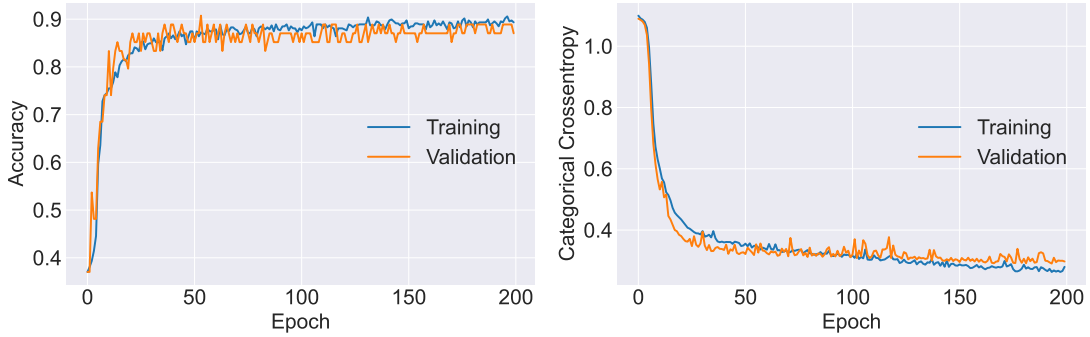

Figure 5: Progression of the accuracy and categorical cross entropy of the fully connected neural network

Figure 5 shows that the fully connected neural network reached training and validation accuracy of over 80 % already after 14 epochs. The additional training only marginally improved the validation accuracy and the loss, respectively. Furthermore, the model starts to overfit the training data from epoch 88 as indicated by a training loss that is lower than the validation loss.

The CNN we are considering here has the following architecture: 6 * (Conv2D → MaxPooling2D → Dropout) → Flatten → 3 * Dense. It consists of 22 layers and 52,419 weights and was trained for 100 epochs using the RMSprop optimizing algorithm, categorical cross entropy as loss function and a train batch size of 16. All Convolutional layers consist of 32 filters with kernel size of (3, 3) and relu activation functions. We defined a pooling size of (2, 2) in the MaxPooling layer and set the dropout rate in the Dropout layers to 0.2. The first two Dense layers after the Flatten layer each include 32 neurons with a relu activation functions. Like the fully connected neural network, the last Dense layer had 3 output neurons with softmax activation functions.

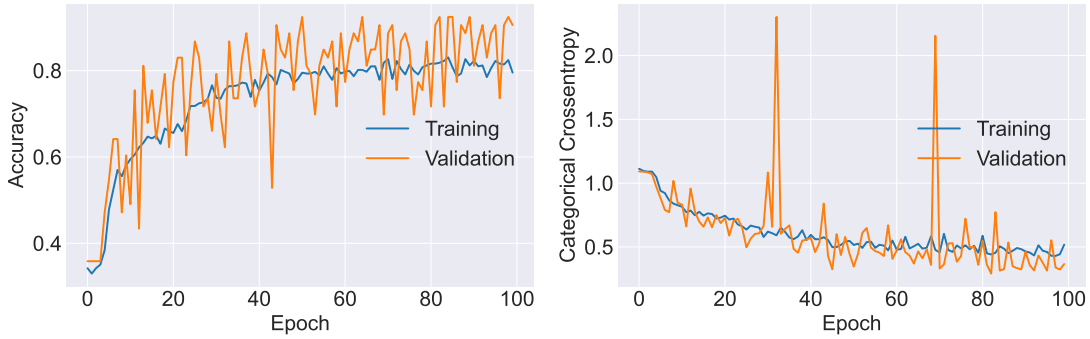

Figure 6: Progression of the accuracy and categorical cross entropy of the convolutional neural network

The CNN required 45 epochs to reach a training accuracy of 80 %. Figure 6 also shows that the validation accuracy and loss fluctuate more than they did for the fully connected neural network. This means that small changes to the weights, which improve the training accuracy, can have a very strong impact on the validation accuracy.

4.2. Visualization of Filter and Activation maps

We extracted the 32 filters in the first layer of the convolutional network presented in the previous subsection to get a better picture of how this classifier works and why the validation accuracy might be prone to such high variance. The 3 x 3-pixel filters displayed in Figure 7 are difficult to interpret. The identified patterns are not as clearly describable as it is the case in sophisticated models such as VGG16 (e. g., edge filters, blob filters).

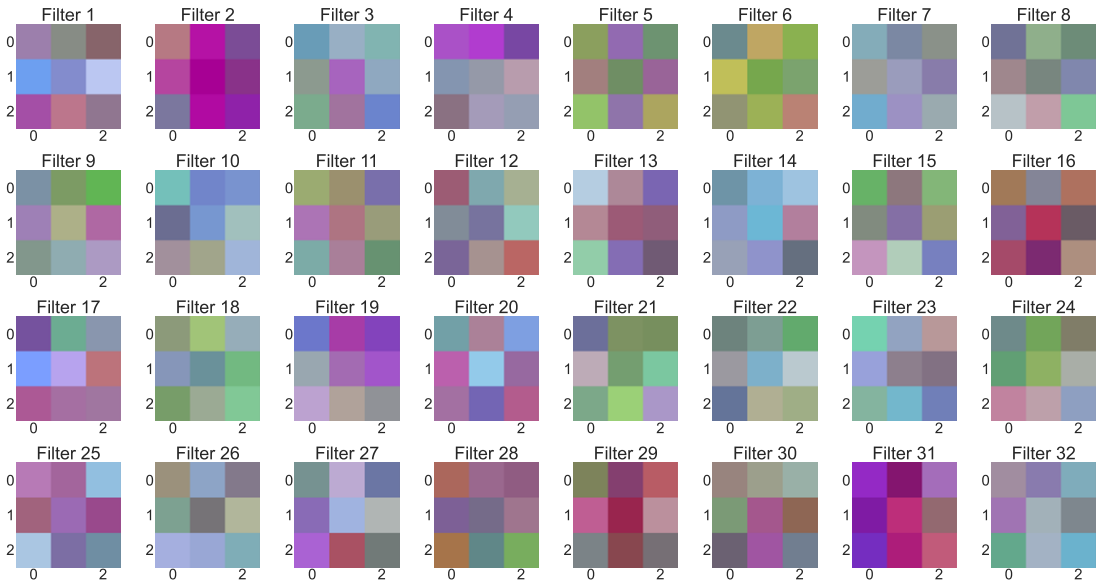

Figure 7: Trained filters in the first convolutional layer

Figure 8 visualizes the activation maps after applying the 32 filters from Figure 7 on the unripe strawberry from Figure 2. The activation maps show to what extent and in which image regions the patterns of the filters were found in the image. Red areas indicate high and blue areas a low activation regarding a specific filter. Figure 8 demonstrates that some filters are trained to detect edges (e.g., filters 11, 24, 30 and 31) whereas others are trained to identify rather greenish (e.g., filter 18) or reddish areas (e.g., filters 3 and 32) of the strawberries. Filters 16 and 28 detect the transparent background of the images. Only a few of these filters correspond to patterns that are obviously meaningful for classifying strawberries according to their ripeness. Some of the filters might detect patterns that are randomly correlated with the ripeness level in the training data. For example, if most unripe strawberries in the training set are prone to chromatic aberration, some filters might be trained to detect this image error rather than to separate unripe from ripe and overripe strawberries. Giving too much weight to such filters leads to rather low validation accuracy, which is a possible explanation for the high variance of the validation accuracy of our trained convolutional neural network.

Some of the more relevant filters could be extracted from the CNN and used to generate additional features for the traditional methods. Filter 13, for example, seems to differentiate between areas with high and low red intensity. The activation map helps to identify interesting additional statistical features that might improve the prediction accuracy.

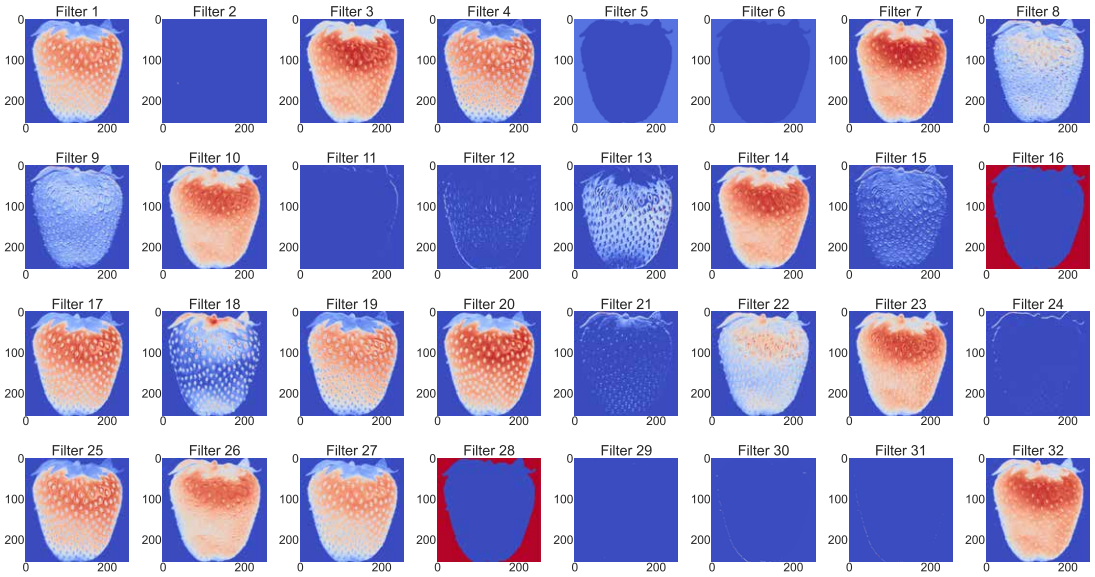

Figure 8: Activation maps after the filter in the first convolutional layer are applied

5. Conclusion

In this paper, we compared convolutional neural networks with some traditional machine learning methods for classifying strawberries into three ripeness classes.

Our results show that convolutional neural networks do not generally lead to a higher performance in case of image-based classification. With an accuracy of 83.73 % convolutional neural networks performed worse than penalized multinomial regression (86.27 %) in our study. The accuracy of most other traditional feature-based machine learning methods is comparable to convolutional neural networks, with the exception of Naïve Bayes and decision trees. This indicates that visual differences between ripeness levels of strawberries can be explained using fairly simple color-based statistics. Convolutional neural networks, as well as fully connected neural networks also have the disadvantage that it is usually impossible to explain why a strawberry is assigned to a certain class. In addition, there is no specific rule for determining the structure of a network. As a result, an extensive search for convincing hyperparameters is necessary. Although traditional methods also consist of some hyperparameters (e.g., penalty weight), the number of hyperparameters in these methods is much lower and the accuracy is often quite high even without hyperparameter tuning. In this study, we only tuned hyperparameters of fully connected neural networks and convolutional neural networks.

Our results also show that the classifier with the best overall test accuracy does not necessarily best classify strawberries of all three ripeness levels. Unripe strawberries are best identified by convolutional neural networks (accuracy: 85.95 %), whereas ripe strawberries are best classified by penalized discriminant analysis (accuracy: 91.84 %) and overripe strawberries are best classified by penalized multinomial regression (accuracy: 84.79 %). Therefore, researchers and practitioners should consider their specific classification situation when selecting an appropriate classifier. If most strawberries are ripe and the goal is to sort out unripe as well as overripe strawberries, our results recommend using a penalized discriminant analysis. These results furthermore demonstrate that there is no single classifier that performs best in all situations. Implementing ensemble classifiers is hence worthwhile in many scenarios ([16, 17]). Especially selective ensembles that statically or dynamically select classifiers for an ensemble could be useful for classifying strawberries according to their ripeness. These ensemble models are suitable for selecting ensembles based on some prior information about the distribution of the ripeness levels. Our study is subject to three

limitations. First, we used a balanced training and test set. Real datasets are usually unbalanced, which could affect the performance of the classifiers compared in our study. Recent research has shown that the performance is lower when an unbalanced dataset is used for training [12]. Future research should thus also compare convolutional neural networks with traditional machine learning methods on unbalanced datasets. Second, we used only one dataset in this study. Investigations on other datasets are necessary to test the robustness of our results. And third, we restricted our comparison to individual classifiers. Several recent studies have provided evidence that ensembles outperform individual classifiers in several classification tasks [12, 16]. Thus, comparing convolutional neural networks with ensemble classifiers for identifying the ripeness level of fruits provides a further avenue for future research.

6. Acknowledgement

This research was funded by the Federal Ministry for Economic Affairs and Energy within the project “Fresh Analytics – Plattform zur KI-Optimierung der Lebensmittellieferkette von Produzent bis Konsument” (Grant No. 01MD19009E).

References

[1] Puig Garcia, Eduard; Gonzalez, Felipe; Hamilton, Grant; Grundy, Paul (2015): Assessment of crop insect damage using unmanned aerial systems: A machine learning approach. In: *Weber, T, McPhee, M J, & Anderssen, R S (Eds.) Proceedings of MODSIM2015, 21st International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand Inc. (MSSANZ)*, S. 1420–1426.

[2] Sadeghi-Tehran, Pouria; Sabermanesh, Kasra; Virlet, Nicolas; Hawkesford, Malcolm J. (2017): Automated Method to Determine Two Critical Growth Stages of Wheat: Heading and Flowering. In: *Frontiers in plant science* 8, S. 252.

[3] Ge, Yuanyue; Xiong, Ya; Tenorio, Gabriel Lins; From, Pal Johan (2019): Fruit Localization and Environment Perception for Strawberry Harvesting Robots. In: *IEEE Access* 7, S. 147642–147652.

[4] Zhang, Qirong; Chen, Shanxiong; Yu, Tingzhong; Wang, Yan (2017): Cherry recognition in natural environment based on the vision of picking robot. In: *IOP Conference Series: Earth and Environmental Science* 61, S. 12021.

[5] Abbaszadeh, Rouzbeh; Moosavian, Ashkan; Rajabipour, Ali; Najafi, Gholamhassan (2015): An intelligent procedure for watermelon ripeness detection based on vibration signals. In: *Journal of food science and technology* 52 (2), S. 1075–1081.

[6] Goel, Nidhi; Sehgal, Priti (2015): Fuzzy classification of pre-harvest tomatoes for ripeness estimation – An approach based on automatic rule learning using decision tree. In: *Applied Soft Computing* 36, S. 45–56.

[7] Nandi, Chandra Sekhar; Tudu, Bipan; Koley, Chiranjib (2014): A Machine Vision-Based Maturity Prediction System for Sorting of Harvested Mangoes. In: *IEEE Transactions on Instrumentation and Measurement* 63 (7), S. 1722–1730.

[8] Mazen, Fatma M. A.; Nashat, Ahmed A. (2019): Ripeness Classification of Bananas Using an Artificial Neural Network. In: *Arabian Journal for Science and Engineering* 44 (8), S. 6901–6910.

[9] Sustika, R.; Subekti, Agus; Pardede, Hilman; Suryawati, E.; Mahendra, O.; Yuwana, S. (2018): Evaluation of deep convolutional neural network architectures for strawberry quality inspection. In: *International Journal of Engineering and Technology (UAE)* 7, S. 75–80.

[10] Zhang, Yan; Lian, Jian; Fan, Mingqu; Zheng, Yuanjie (2018): Deep indicator for fine-grained classification of banana’s ripening stages. In: *EURASIP Journal on Image and Video Processing* 2018 (1).

[11] Wolpert, David (1996): The Lack of A Priori Distinctions Between Learning Algorithms. In: *Neural Computation* 8, S. 1341–1390.

[12] Scholz, Michael; Wimmer, Tristan (2020): A comparison of classification methods across different data complexity scenarios and datasets. In: *Expert Systems with Applications*, S. 114217.

[13] Castro, Wilson; Oblitas, Jimy; De-La-Torre, Miguel; Cotrina, Carlos; Bazan, Karen; Avila-George, Himer (2019): Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces. In: *IEEE Access* 7, S. 27389–27400.

[14] Indrabayu, Indrabayu; Arifin, Nurhikma; Areni, Intan Sari (2019): Strawberry Ripeness Classification System Based On Skin Tone Color using Multi-Class Support Vector Machine. In: *International Conference on Information and Communications Technology (ICOIACT)*, S. 191–195.

[15] Landis, J.; Koch, G. (1977): The measurement of observer agreement for categorical data. In: *Biometrics* 33 1, S. 159–174.

[16] Lessmann, Stefan; Baesens, Bart; Seow, Hsin-Vonn; Thomas, Lyn C. (2015): Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. In: *European Journal of Operational Research* 247 (1), S. 124–136.

[17] Bashir, Saba; Qamar, Usman; Khan, Farhan Hassan (2016): IntelliHealth: A medical decision support application using a novel weighted multi-layer classifier ensemble framework. In: *Journal of Biomedical Informatics* 59, S. 185–200.

Leon Binder (M.Sc.)

Leon Binder studied business informatics at the Deggendorf Institute of Technology (DIT). Since 2019, he has been working as a researcher in the team "Business Data Analytics & Optimization" at the Technology Campus (TC) Grafenau, focusing on the areas data analytics, machine learning and computer vision.

Leon Binder studierte Wirtschaftsinformatik an der Technischen Hochschule Deggendorf (THD). Seit 2019 ist er als wissenschaftlicher Mitarbeiter im Team „Business Data Analytics & Optimization" am Technologie Campus (TC) Grafenau im Bereich Datenanalyse, Maschinelles Lernen und Computer Vision tätig.

Contact / *Kontakt*
✉ **leon.binder@th-deg.de**

Dr. Michael Scholz, Dipl.-Wirt.-Inf. (Univ.)

Michael Scholz is the head of the research team “Business Data Analytics and Optimization” at the TC Grafenau (DIT). His research is focused on business data analytics and the economic effects of e-commerce applications. He is the author of several papers in journals, such as the European Journal of Operational Research, Decision Support Systems, Journal of Statistical Software, Electronic Markets, and Business & Information Systems Engineering.

Michael Scholz leitet das Forschungsteam „Business Data Analytics and Optimization“ am TC Grafenau der THD. In seiner Forschung untersucht er Methoden zur Analyse von Unternehmensdaten und ökonomische Effekte insbesondere von E-Commerce-Anwendungen. Er ist Autor einer Vielzahl von Veröffentlichungen in wissenschaftlichen Zeitschriften wie dem European Journal of Operational Research, Decision Support Systems, dem Journal of Statistical Software, Electronic Markets und Business & Information Systems Engineering.

Contact / *Kontakt*
✉ **michael.scholz@th-deg.de**

Dr. rer. nat. Roman-David Kulko, Dipl. Chem. (Univ.)

Roman-David Kulko is a research associate in the “Applied Artificial Intelligence” research group at the TC Grafenau. His research interests lie in interdisciplinary applied spectroscopy and machine learning.

Roman-David Kulko ist wissenschaftlicher Mitarbeiter in der Forschungsgruppe "Applied Artificial Intelligence" des TC Grafenau. Seine Forschungsinteressen sind die interdisziplinäre angewandte Spektroskopie und maschinelles Lernen.

Contact / *Kontakt*
✉ **roman-david.kulko@th-deg.de**