Computers and Electronics in Agriculture 211 (2023) 107944

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Comparison of deep learning methods for grapevine growth stage recognition

Martin Schieck [a,*], Philippe Krajsic [a], Felix Loos [a], Abdulbaree Hussein [a], Bogdan Franczyk [a], Adrianna Kozierkiewicz [b], Marcin Pietranik [b]

[a] Leipzig University, Grimmaische Straße 12, 04109 Leipzig, Germany
[b] Wroclaw University of Science and Technology, Wybrzeze Wyspianskiego Street 27, 50-370 Wroclaw, Poland

ARTICLE INFO

Dataset link: https://www.kaggle.com/datasets/schieckmartin/grapesdevelopmentstages

Keywords:
Deep learning
Viticulture
Computer vision
Grapes growth stages

ABSTRACT

Monitoring the phenological development stages of grapes represents a challenge in viticulture. It includes the phenological distinction of the growth stages of grapevines and the continuous technological developments, especially in computer vision, enabling a detailed classification of economically relevant development stages of grapes. In the present work, we show that based on a cascading computer vision approach, the development stages of grapes can be classified and distinguished at the micro level. In a comparative experiment (ResNet, DenseNet, InceptionV3), it could be shown that a ResNet architecture provides the best classification results with an average accuracy of 88.1%.

1. Introduction

Viticulture is a domain full of background knowledge. Besides changing long-term climate conditions, it is also full of short-term uncertainty and fuzziness, like differing local weather and soil conditions and a need for qualified employees. However, modern viticulture is increasingly digitized, so actual data supplemented by artificial intelligence (AI)-based functions can help farmers make decisions that are impossible for humans to accomplish. Essential decisions such as applying pesticides are based on recognizing the development stages of grapevines. For example, some plant protection products can only be used at certain stages of development. Due to this necessity, a descriptive scheme was developed (BBCH scale) (Meier et al., 2009), which precisely describes the vegetation state of grapevines. Two numbers describe the development stages: the macro stage and the micro stage. For example, BBCH code 73 describes macro stage 7 (fruit development) and micro stage 3 (berries are grain sized). This way, rough and detailed distinctions can be made in developing grapevine.

Current research on the applications of machine learning techniques focused on the recognition of growth or maturity stages based on distinguishing colors of grapes to determine the right time for harvest (Reis et al., 2012; Pereira et al., 2018; Nayak et al., 2019; Aguiar et al., 2021). Also, in the field of special crops, different alternatively computer vision methods like Googlenet, Alexnet (Muhammad et al., 2018), YOLO (Sozzi et al., 2022), VGGNet (Yu et al., 2019), DetectNet (Yu et al., 2019), ResNet (Cecotti et al., 2020; Gonzalez et al., 2019) and FCN (Grimm et al., 2019) were applied to the fruit detection and recognizing fruit types. The latest developments are increasingly used in applications such as harvest field estimation, grape disease detection, grape phenology, vineyard management and monitoring, quality evaluation, and grape phenology (Seng et al., 2018). Our previous work (Franczyk et al., 2020) focused on automating grape recognition and variety identification. Our developed model for grape identification at a vineyard reaches an accuracy of 99% of correctly recognized grape varieties. Some of the results mentioned above were promising but insignificant because most proposed solutions have been verified using a minimal test dataset. A more extensive and annotated dataset must be generated to build reliable models. Furthermore, these approaches are only suitable after the stage relevant for plant protection, as they can only be used at a stage when the berries begin to change color. However, the berries are already closed at this stage, and the pesticides cannot exert their effect. Aguiar et al. (2021) provided a novel dataset with 1929 images, and respective annotations were different stages of grape bunches obtained by visiting a vineyard four different times to record the data. This time-lapse is too big to recognize the micro level development. BBCH develops within a few days, depending on the weather's influence, and decisions must be made from one day to the next.

To detect bunches of grapevines, the authors of Sozzi et al. (2022) compared six versions of the YOLO object detection algorithm (YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny, YOLOv5x, and YOLOv5s). The best combination of accuracy and speed was achieved by YOLOv4-tiny. Therefore, in the present work, also a YOLOv4 was evaluated for the segmentation of the grape bunches. To the best of our knowledge, just a few papers focused on classifying micro development

* Corresponding author.
E-mail address: schieck@wifa.uni-leipzig.de (M. Schieck).

https://doi.org/10.1016/j.compag.2023.107944
Received 10 November 2022; Received in revised form 9 March 2023; Accepted 18 May 2023
Available online 7 June 2023

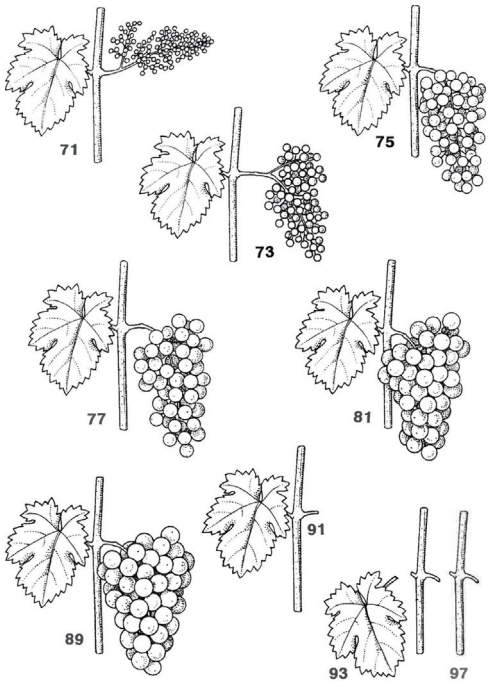

Fig. 1. BBCH 71-79 for grapevines.
Source: Adapted from Meier (2018).

stages. In Shen et al. (2022), a method combining deep learning and image analysis to identify development stages in colored wine grapes under natural field growing conditions has been proposed. The results showed that the Mask R-CNN with ResNet50-FPN structure as the backbone feature extraction network performed well with average precision equal to 81.53%. However, the research refers to growth stage 8: the ripening or maturity of fruit and seed, which was not considered in our research.

This research addresses the described issue by comparing different deep learning methods for the grapevine growth stage recognition problem. Additionally, we propose a method for multi-class grape bunch detection, which allows a detailed distinction on the micro development stage, especially between stages 71–79, since it is within these stages that decisions are made on the application of economically relevant plant protection products. Fig. 1 gives a visual intuition on these stages, where the appearance of BBCH 79 (“Majority of berries touching”) could be considered equal to 81 in order to distinguish from 77 (“Berries beginning to touch”.) (Meier, 2018).

To evaluate the suitability of deep learning methods, a cascading approach was applied. First, an image segmentation based on a YOLOv4 neural network was performed. The resulting bounding boxes were passed to an image cropping process. The obtained grape bunches’ images were then classified using different neural networks (DenseNet, ResNet, InceptionV3) according to their development stage. To train the segmentation and classification algorithms, we generated an annotated dataset with information on the BBCH stage of the respective grapevines. Therefore our work contributes to the research field in the following way:

- A new annotated dataset.
- A new algorithm for grape bunch detection at different BBCH micro level.
- A new algorithm for classification of phenological development stages of grape bunches at the BBCH micro level.

The remainder of this paper is structured as follows: In Section 2, the data preparation procedure and the model development are presented. Section 3 describes the obtained results from the conducted experiments. Section 5 concludes the research findings and gives an outlook on future work.

2. Materials and methods

We designed the data acquisition process to create an image dataset that can be used for many grapevine-related topics. From this more general dataset, we created a labeled dataset that we used to train our grape bunch segmentation and BBCH classification algorithm.

2.1. Data acquisition

To gather the images for our dataset, we installed 100 RGB cameras of type “Blink Outdoor” in different commercially used vineyards around Meissen, Saxony, Germany. One camera produces a picture with the size of 1 280 pixel-width and 720 pixel-height. The cameras are monitoring ten different varieties of grapevines (Spätburgunder, Schwarzrießling, Frühburgunder, Elbling, Grauburgunder, Goldrießling, Regent, Weißburgunder, Dornfelder, Scheurebe) from four different perspectives as shown in Fig. 2. To capture the BBCH macro stages 6, 7, and 8, we started capturing at the beginning of June and stopped capturing at the end of August. All cameras take images every hour and upload the images to cloud storage. Due to the technical limitations of the cameras, an image could only be uploaded if the camera had a connection to the internet. Because of geographical obstacles and the growing leaf density in the vineyard during the vegetation period, the bandwidth of an outdoor wireless sensor network could decrease, and some cameras cannot upload images all the time (Krug et al., 2021). Therefore, the number of connected cameras varied between 20 and 50 per hour. At the end of the relevant part of the growing season, we collected 61 381 images.

2.2. Data sampling and labeling

Due to limited human expert resources, labeling all the collected image data was impossible for us, so we created a subset of the initial dataset. We used only images from every camera captured at 6:00, 8:00 am and 12:00, 2:00, and 6:00 pm.

We labeled the grape bunches with rectangular bounding boxes on these images and added the BBCH code to every bounding box. It is possible that within one image, some bunches with different BBCH stages are visible. Because of these circumstances, we decided to approach a cascading process described in detail in the following Section 2.3. For labeling, we used the online tool cvat.ai.[1] This tool provides chronological sorting of images to significantly reduce labeling time by propagating bounding boxes over chronological images and adjusting the propagated bounding boxes. During the labeling period, the labeling team could not estimate the grapevine development stage on every image of the initial dataset. Among the circumstances that caused this inability to estimate the development stage are bad lightning conditions of the image and leaves in front of the camera covering the lens. Also, the rainy weather conditions caused a blurring of the images. Furthermore, during the course of the year, camera angles have changed due to mechanical impacts from working in the vineyards and we have lost some good usable focus on grapes. After labeling the images, the dataset consists of 1 212 images containing bounding boxes. We use these images for the segmentation training in Section 2.4, performed during the vegetation period. At the end of the vegetation period, we produced 2 099 labeled images of sufficient quality for training.

Furthermore, for the classification task in Section 2.5, we excluded all bounding boxes smaller than 5 000 px because even for human experts, it was not possible to distinguish between the growth stages

[1] https://www.cvat.ai/

Fig. 2. Top left: Bottom up perspective, Top right: Side to side perspective, Bottom left: Top down perspective, Bottom right: Middle perspective.

Table 1
Size of the dataset for segmentation and classification.

Dataset	Total number of photos with bounding boxes	Total number of bounding boxes within the photos
v0.1	1212	3993
v0.2	2099	6641

accurately. Furthermore, for the problem discussed in the paper, only BBCH stages from the macro class 7 are relevant, so we excluded all images that contain other stages. After all, we had 3 993 bounding boxes to train the grape bunch segmentation and BBCH classification algorithm. At the end of the growing season, we generated 6 641 bounding boxes with an area greater than 5 000 px. Table 1 summarizes the size of the two dataset versions.

The dataset was uploaded to kaggle.[2] While the "v0.1" contains the images used for training the models in this work, "v0.2" includes all the images of the vegetation period as described in this section. "v0.2" thus includes all images from version "v0.1" supplemented by the images from the rest of the year.

At the end of the data preparation step, we have two different datasets for the following tasks — segmentation and classification:

- A dataset with 2 099 labeled images with bounding boxes of size 1 280 × 720 px for the segmentation task (1 212 images used for training)
- A dataset with 6 641 images with cropped grape bunches of different sizes depending on the labeled bounding box (3 993 used for training)

2.3. Model development

In the following part, we propose a cascading approach to the considerations above. The general overview of the process we developed can be found in Fig. 3. It consists of three stages:

1. image segmentation
2. image cropping
3. image classification.

The input is a single image on which the framework performs the segmentation, returning a set of bounding boxes that point out areas containing grape bunches. These bounding boxes are then used to crop selected areas, providing images containing only the extracted segments with a single grape bunch. Finally, such partial images are fed into the neural network, which performs the final classification. The output is a vector of probabilities of classifying a given grape bunch from the cropped input image into selected development stages assigned with the corresponding BBCH codes.

2.4. Segmentation

The segmentation stage is built by adapting the YOLOv4 neural network (Wang et al., 2022). It is trained on 1 212 labeled images mentioned in Section 2.2. The training procedure was performed using the hardware setup described in Section 3. The data is split into 70% training, 15% validation, and 15% test. The training process comprises 300 epochs.

2.5. Classification

The dataset described in Section 2.2 consists of images, which human experts have annotated with 3 993 bounding boxes pointing separate grape bunches, each with an area greater than 5 000 px. This allowed us to categorize the bunches further using BBCH classification into five different classes: 71, 73, 75, 77, and 79. Due to the initial imbalance between the first two classes and the latter three, we have decided to cluster classes 71 and 73 into a single class 71_73 representing the early grape development stage. Images belonging to each class have been divided into three subsets: training set (containing 70% of images), validation set (containing 10% of images), and test set (containing 20% of images). The detailed distribution of the described set of images can be found in Table 2.

The first two sets have been used to train three state-of-the-art neural network models, namely ResNet (Wu et al., 2019), DenseNet (Iandola et al., 2014) and InceptionV3 (Szegedy et al., 2017; Xia et al., 2017), which are explained below. The last of the described datasets

[2] https://www.kaggle.com/datasets/schieckmartin/grapesdevelopmentstages

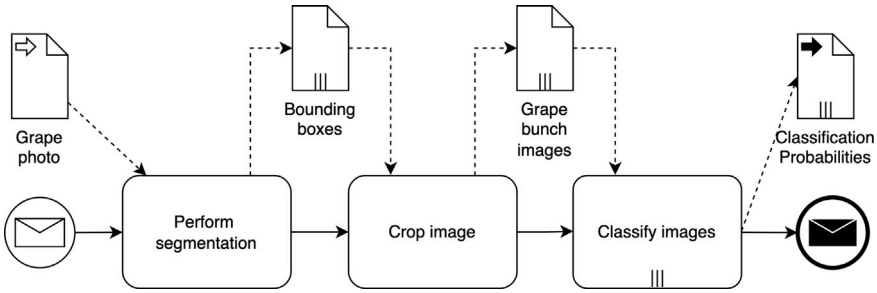

Fig. 3. Grape growth stage detection process overview.

Table 2
Distribution of training photos.

Class label	Total number of grape bunch photos	Number of grape bunch photos in the training set	Number of grape bunch photos in the validation set	Number of grape bunch photos in the test set
71_73	975	713	81	181
75	726	532	66	128
77	1200	876	97	227
79	1092	804	93	195
Total	**3993**	**2925**	**337**	**731**

Table 3
Hyperparameter of neural network training.

	InceptionV3	DenseNet	ResNet
optimizer	SGD	SGD	SGD
weight decay	0	1e−4	1e−4
learning rate	0.045	0.1	0.1
lr decay rate	0.94	none	none
lr step size	2	none	none
num epochs	100	100	100
momentum	0.9	0.9	0.9
batch size	32	32	32

Table 4
Summary of neural network training.

	InceptionV3	DenseNet	ResNet
Loss	0.3941	0.6272	0.5166
Accuracy	0.8420	0.7992	0.7909
Value Loss	0.8050	0.9316	0.8050
Value Accuracy	0.7281	0.7000	0.6844

Table 5
Comparison of training duration.

	InceptionV3	DenseNet	ResNet
num epochs	**52**	59	78
duration best model (s)	**9956**	25 000	25 065
duration 100 epochs (s)	**21 741**	47 529	34 832

have been extracted solely for experimental comparison of the aforementioned neural networks, which is covered in the next section.

Residual nets (ResNets) learn residual functions by reference to the layer inputs, rather than learning unreferenced functions. Instead of hoping that each of the few stacked layers directly corresponds to a desired underlying mapping, residual nets let those layers correspond to a residual mapping. They stack residual blocks on top of each other to form a net. ResNet has several implementations with different depths. For example, a ResNet-50 has fifty layers using these blocks. In this study, a ResNet-50 has been used.

The second neural net used in this paper is DenseNet. This type of convolutional network uses dense connections between layers, through dense blocks, where all layers (with matching feature map sizes) are directly connected. To maintain the feed-forward character, each layer receives additional input from all previous layers and passes its own feature maps to all subsequent layers. Furthermore, DenseNet strengthens feature propagation and promotes feature reuse.

A third type of neural nets is Inception-v3, which is a convolutional network architecture from the Inception family that has several improvements, including the use of label smoothing, factorized 7 x 7 convolutions, and the use of an auxiliary classifier to pass label information further down the network. It breaks down a large-scale convolution kernel into smaller convolution kernels to further reduce network parameters and make it faster to run without lowering overall performance.

The training of all three models has been performed using ImageAI (Moses, 2018)[3] python-library on the hardware described in Section 3. The setting of the hyperparameters for the training of the three networks are summarized in Table 3.

Because the model's performances in identifying the correct class differ just a little between the three compared models (Table 4), we additionally compared the training performance of the model. The number of epochs (num epochs) and the duration of the training process for each of the models are summarized in Table 5. We did not set an early stopping for the training, so the table differentiates between the duration for the epoch in which the best model performance was achieved (duration best model) and the duration for the whole training process (100 epochs). The results show that the duration for the training of the InceptionV3 model took less than half of the time as the DenseNet and ResNet models whereby a similar prediction performance was achieved.

3. Experimental results

As a result of the experiments, we show a workflow combining a grape bunch object detector with a classification of grape development stages. Because we have proposed a two-stage architecture, the experiment has been divided into two parts. We want to verify the model developed for image segmentation in the first part. The second one will indicate the best architecture for image classification.

All training and validation of the object detection and classification have been performed on a Tensorbook laptop with Intel Core i7-11800H, NVIDIA® RTX™ 3080 Max-Q GPU, and 64 GB of RAM.

3.1. Image segmentation

The Precision (Eq. (2)), Recall (Eq. (3)), and F1-Score (Eq. (4)) measures are calculated to measure the object detector's performance. While train and validation datasets are used for the training process, the calculation of Precision and Recall are based on true positives (TP), false positives (FP), and false negatives (FN) of the predicted bounding boxes of the test data. In this regard, true and false predictions depend on the threshold value "intersection over union" (IoU) (Eq. (1)) of predicted bounding boxes ($bbox_{prediction}$) compared to the

[3] http://imageai.org/

Table 6
Summary of object detection.

Confidence	AP@.5 = 75.96%			AP@.75 = 27.04%		
	Precision	Recall	F1	Precision	Recall	F1
0.00	0.37	0.88	0.52	0.20	0.47	0.28
0.10	0.68	0.83	0.75	0.38	0.46	0.42
0.20	0.74	0.81	0.77	0.41	0.46	0.43
0.30	0.76	0.79	0.78	0.43	0.45	0.44
0.40	0.79	0.78	0.79	0.45	0.44	0.45
0.50	0.81	0.75	0.78	0.47	0.44	0.45
0.60	0.84	0.71	0.77	0.50	0.42	0.46
0.70	0.87	0.67	0.76	0.53	0.41	0.46
0.80	0.90	0.61	0.72	0.57	0.39	0.46
0.90	0.92	0.47	0.62	0.59	0.30	0.40
0.99	0.95	0.04	0.08	0.73	0.03	0.06

ground truth bounding boxes ($bbox_{ground\_truth}$). That is measured by the Jaccard-similarity (Jaccard, 1912) of the two areas $bbox_{prediction}$ and $bbox_{ground\_truth}$ in

$$IoU = \frac{area(bbox_{prediction} \cap bbox_{ground\_truth})}{area(bbox_{prediction} \cup bbox_{ground\_truth})} \quad (1)$$

where Union is the sum of prediction and Intersection is the overlap of these two areas. Therefore a higher IoU-value requires more overlapping of the predicted bounding box to the labeled ground truth, which leads to more restrictive consideration of true predictions. The most used IoU-threshold-values are 50% and 75% (Padilla et al., 2021).

The Precision, Recall, and F1 metrics are then defined as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 * \frac{TP}{TP + FP + FN} \quad (4)$$

An object detector with high precision performs well in selecting only relevant objects. In contrast, a high recall determines a good performance in finding all relevant objects. There is a trade-off between precision and recall for one specific detector, which can be plotted as a precision–recall-curve (PRC). The PRC of our proposed YOLOv4-detector is shown in Fig. 5. The average precision (AP) is an accepted metric to compare the performance of an object detector and is defined as the area under the PRC. This choice is dictated to handle the trade-off between precision and recall.

The AP in this work is calculated by performing an all-point-interpolation over the confidence levels shown in Table 6. We used this all-point-interpolation procedure for calculating the AP as it is also the accepted metric for Pascal-challenge (Everingham et al., 2015). The AP@0.5 mentioned above and AP@.75 are calculated over different confidence levels at IoU-threshold 0.5 and 0.75 on the validation data. The results are shown in Table 6. Fig. 4 shows the object detection result at different growth stages from one of the cameras applied in the vineyard, ranging from BBCH-Code 71, 73, 75, 77 to 79.

While previous works on detecting grape bunches at different development stages achieved an AP of 66.96% (Aguiar et al., 2021), our model realizes an AP of 75,96% on different stages. The YOLOv4 configured with an IoU of 50% performs best in our experiments.

3.2. Image classification

We compared three architectures for the micro BBCH level classification task. The expert interview and experiment with the images of the test dataset were performed with a human expert to prove the reliability of the model output. That expert worked in leading positions in vinegrowing companies and has over 39 years of experience. In our experiments, all three classification models outperform the human expert.

Table 7
Comparison of neural network architectures' performances.

		BBCH code			
		71_73	75	77	79
InceptionV3	Accuracy	0.897	0.855	0.833	**0.914**
	Precision	0.770	0.596	0.738	**0.830**
	Recall	0.834	0.531	0.718	0.851
	F	0.801	0.562	0.728	**0.841**
DenseNet	Accuracy	0.874	0.855	0.814	0.891
	Precision	0.705	0.631	0.710	0.770
	Recall	**0.845**	0.414	0.678	0.841
	F	0.769	0.500	0.694	0.804
ResNet	Accuracy	**0.908**	**0.870**	**0.843**	0.903
	Precision	**0.806**	**0.649**	**0.750**	0.795
	Recall	0.829	**0.563**	**0.740**	**0.856**
	F	**0.817**	**0.603**	**0.745**	0.825
Human expert	Accuracy	0.861	0.735	0.689	0.801
	Precision	0.633	0.485	0.368	0.660
	Recall	0.655	0.410	0.378	0.717
	F	0.644	0.444	0.373	0.688

The results of conducted experiments have been collected and presented as confusion matrices. The confusion matrix shows the combination of actual and predicted classes. It is a good measure of whether models can account for overlapping class properties and understand which classes are most easily confused (see Fig. 6).

The helpful extension of the confusion matrix, which visualizes the misclassified classes, is class prediction error as a stacked bar Fig. 7. Interesting conclusions could be drawn for classes 75 and 77. It is easy to see that ResNet architecture is the least likely to be wrong for the mentioned classes. It is essential in the process of pesticide dose management. Two time points are relevant to control and prevent fungal diseases on bunches: Spraying into the outgoing flower (I) and the final treatment (II). With the transition from BBCH 69 to 71, spraying into the outgoing flower (I) is necessary to cover the young and unprotected berries with a protective film and to protect them against peronospora and oidium. Immediately before the end of berry growth at BBCH 77, a final treatment (II) is required to protect the closing grapes during the ripening phase against the now increased risk of a botrytis infection (Müller, 2019). Therefore, accurate predictions of the transition BBCH 75 to 77 and BBCH 77 to 79 are crucial.

Based on confusion matrices, we present the classification report that provides the main classification metrics on a per-class basis, like accuracy, precision, and F-1 measure. All of them have been defined in Section 3.1. In the case of multi-class problems, micro-averaging and macro-averaging are used to extract a single number for each precision, recall, and other metrics across multiple classes. Although the micro-average is usually used in unbalanced data sets (our dataset is slightly unbalanced) for a broader analysis, we present both micro- and macro-values. Tables 7 and 8 present calculated measures separately for each class and cumulatively, respectively.

It is clear that the models can differentiate sufficiently between the four relevant stages, with slightly better performance detecting the two outer stages (71_73, 79) and uncertainty predicting stages in the middle of the range (75, 77). This is probably caused by minor differences in pictures presenting grape bunches in growth stages 75 and 77. Except in isolated cases, the best performance was almost always shown by the ResNet architecture and the worst by the human expert (Table 7). The non-uniform quality of images causes a lower value of recall. In many test cases, the differences in grape growth stages are imperceptible and not noticeable, so it is not easy to detect them. This leads to incomplete classification results.

With an accuracy of 0.881, precision, recall, and F-1 of 0.762 in the experiment run (Table 8), the ResNet model shows the best performance in classifying the micro stages of BBCH for grapes, which is quite a good result. Our models are more than 10% more accurate than human-made classification by an expert on the same dataset.

Fig. 4. Images acquired from one camera perspective at different growth stages.

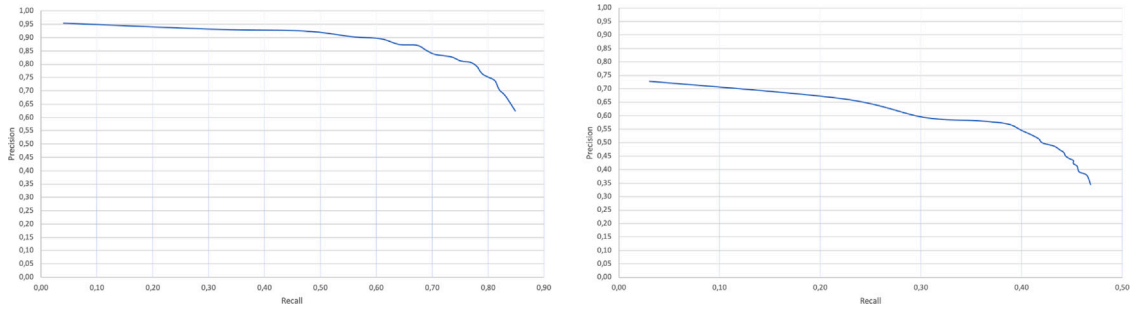

Fig. 5. Precision–recall-curve for IoU-thresholds 50% and 75%.

Table 8
Comparison of micro and macro average performances.

	Micro average				Macro average			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
InceptionV3	0.875	0.750	0.750	0.750	0.875	0.734	0.734	0.733
DesneNet	0.858	0.717	0.717	0.717	0.858	0.704	0.695	0.702
ResNet	**0.881**	**0.762**	**0.762**	**0.762**	**0.881**	**0.750**	**0.747**	**0.747**
Human expert	0.772	0.543	0.543	0.543	0.772	0.537	0.54	0.537

Accuracy and other classification metrics can be misleading because they do not consider class imbalance. Thus to draw more reliable conclusions, we have chosen Cohen's Kappa score (Cohen, 1960). This statistic allows for measuring the inherent uncertainty that comes with generating predictions. Table 9 presents the calculated Cohen's Kappa coefficients. All the obtained results are statistically relevant for an

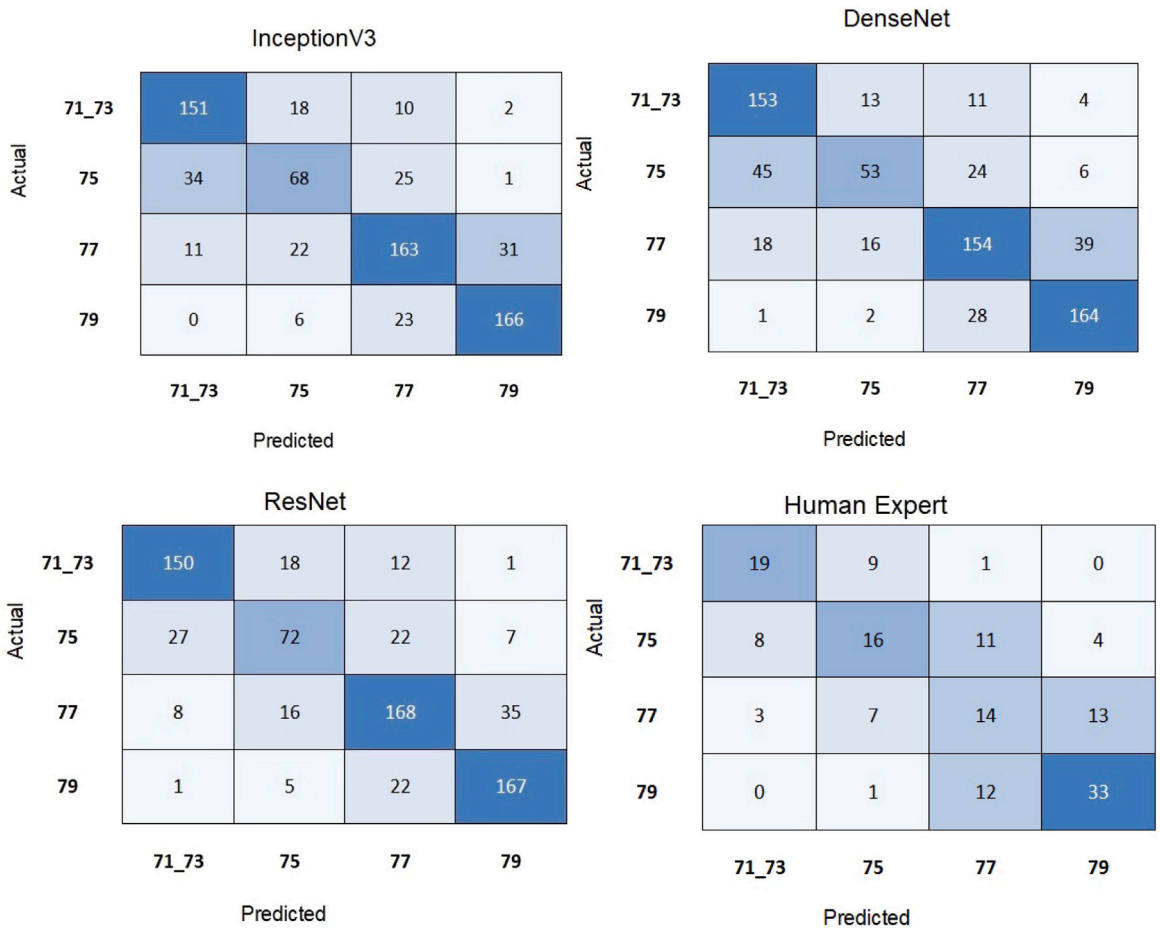

Fig. 6. Confusion matrixes.

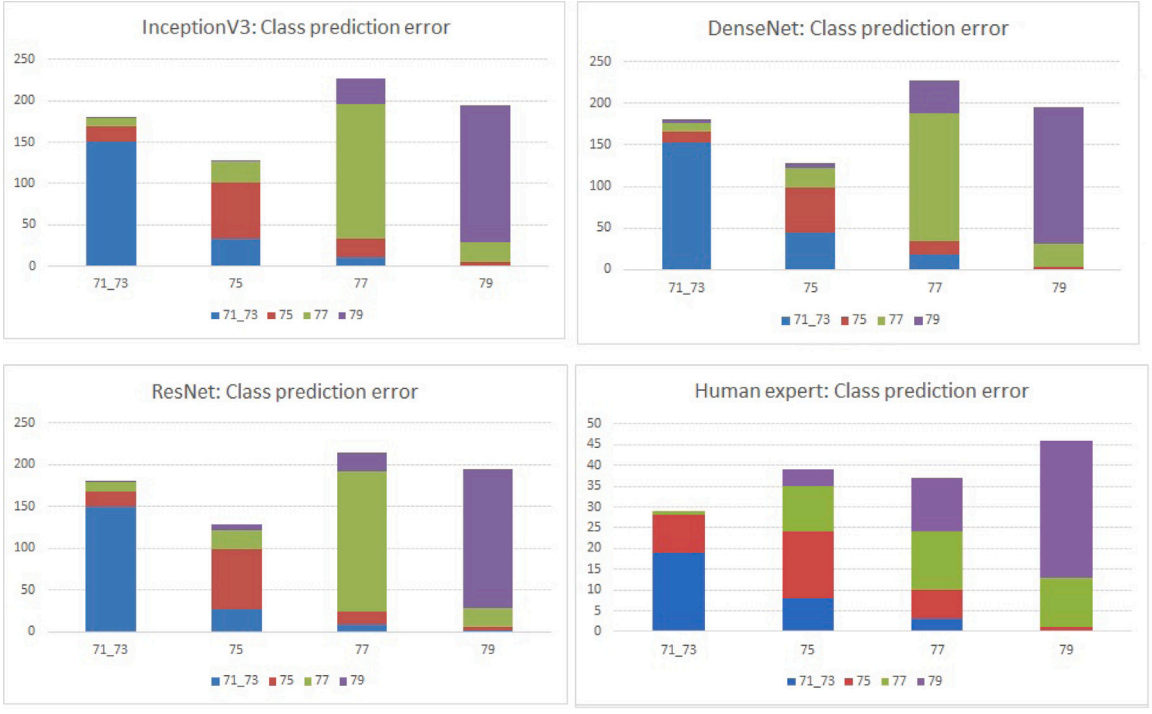

Fig. 7. Class prediction error.

Table 9
Comparison of Cohen's Kappa scores.

	InceptionV3	DenseNet	ResNet	Human expert
Kappa	0.6614	0.6155	**0.6777**	0.3849
statistics value	30.4816	28.2838	31.1972	8.1315
p-value	0.000001	0.000001	0.000001	0.000001

assumed significance level equal to 0.05 (p-value equal to 0.000001). The calculated Cohen's Kappa score equals 0.3894, which indicates a fair agreement between the predictions made by a human expert and the ground truth (Landis and Koch, 1977). All neural network models can get substantial agreement. The ResNet architecture turned out to be the best in this ranking.

4. Discussion

The generation and use of a temporally high-resolution image dataset have supported the development of the presented models. However, relevant results could only be obtained from image size and quality of more than 5000px for a single bunch. To evaluate the year-round support of management tasks by in-field cameras, generating images in further phases of the vegetation period is necessary. The results of this work are based on the detection of grapevines. Based on this, however, it is also possible to expand the potential for detecting development stages on other special crops such as apples. Furthermore, integrating the models into running farm-management-information-systems was not the focus of this work. From an application and transfer perspective, winegrowers' handling of the provided functions still needs to be answered so far. Therefore, future work should include uncertainty quantification of the model output in the classification of grapevines. These investigations support the classification process in considering uncertainties, such as out-of-distribution conflicts.

5. Conclusions

This paper has shown that the development stages of grapes can be distinguished not only at the macro level, which extends current research in the field. With the proposed deep learning model, about 9 out of 10 images of grape bunches can be correctly identified. It was shown that the permanent installation of in-field cameras could help to determine the BBCH stage over the course of the year for the application of crop protection. This can contribute to a more targeted and demand-oriented application of crop protection and, thus to a reduction in the use of pesticides. The performance of the computer vision approaches considered here (ResNet, InceptionV3, DenseNet) for classifying BBCH stages differs only slightly, while the training process of InceptionV3 outperformed both other model training processes by less than the half of the needed duration. In summary, all methods show a promising potential to discriminate at the BBCH micro level.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared the data for this research on Kaggle: https://www.kaggle.com/datasets/schieckmartin/grapesdevelopmentstages.

Acknowledgments

This research project was partially supported by funds of the Federal Ministry of Food and Agriculture (BMEL), Germany based on a decision of the Parliament of the Federal Republic of Germany. The Federal Office for Agriculture and Food (BLE), Germany provides coordinating support for digitisation in agriculture as funding organisation, grant number FKZ 28DE102A18 and by program "Inicjatywa doskonałości – uczelnia badawcza" (IDUB), Poland, MPK: 9240450000, 8211104160.

References

Aguiar, A.S., Magalhães, S.A., Dos Santos, F.N., Castro, L., Pinho, T., Valente, J., Martins, R., Boaventura-Cunha, J., 2021. Grape bunch detection at different growth stages using deep learning quantized models. Agronomy 11 (9), 1890.

Cecotti, H., Rivera, A., Farhadloo, M., Pedroza, M.A., 2020. Grape detection with convolutional neural networks. Expert Syst. Appl. 159, 113588.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20 (1), 37–46.

Everingham, M., Eslami, S., Van Gool, L., Williams, C., Winn, J., Zisserman, A., 2015. The PASCAL visual object classes challenge: A retrospective. Int. J. Comput. Vis. 111 (1), 98–136. http://dx.doi.org/10.1007/s11263-014-0733-5.

Franczyk, B., Hernes, M., Kozierkiewicz, A., Kozina, A., Pietranik, M., Roemer, I., Schieck, M., 2020. Deep learning for grape variety recognition. Procedia Comput. Sci. 176, 1211–1220.

Gonzalez, S., Arellano, C., Tapia, J.E., 2019. Deepblueberry: Quantification of blueberries in the wild using instance segmentation. Ieee Access 7, 105776–105788.

Grimm, J., Herzog, K., Rist, F., Kicherer, A., Toepfer, R., Steinhage, V., 2019. An adaptable approach to automated visual detection of plant organs with applications in grapevine breeding. Biosyst. Eng. 183, 170–183.

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.

Jaccard, P., 1912. The distribution of the flora in the alpine zone.1. New Phytol. 11 (2), 37–50. http://dx.doi.org/10.1111/j.1469-8137.1912.tb05611.x, URL https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x.

Krug, S., Miethe, S., Hutschenreuther, T., 2021. Comparing BLE and NB-IoT as communication options for smart viticulture IoT applications. In: 2021 IEEE Sensors Applications Symposium. SAS, pp. 1–6. http://dx.doi.org/10.1109/SAS51076.2021.9530069.

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 159–174.

Meier, U., 2018. Growth stages of mono- and dicotyledonous plants: BBCH monograph. ISBN: 978-3-95547-071-5, URL https://www.openagrar.de/receive/openagrar_mods_00042351.

Meier, U., Bleiholder, H., Buhr, L., Feller, C., Hack, H., Heß, M., Lancashire, P.D., Schnock, U., Stauß, R., Van Den Boom, T., et al., 2009. The BBCH system to coding the phenological growth stages of plants–history and publications. J. Kulturpflanzen 61 (2), 41–52.

Moses, 2018. Imageai, an open source Python library built to empower developers to build applications and systems with self-contained computer vision capabilities. URL https://github.com/OlafenwaMoses/ImageAI.

Muhammad, N.A., Nasir, A.A., Ibrahim, Z., Sabri, N., 2018. Evaluation of CNN, alexnet and GoogleNet for fruit recognition. Indonesian J. Electr. Eng. Comput. Sci. 12 (2), 468–475.

Müller, E., 2019. Der Winzer 1: Weinbau. Ulmer.

Nayak, M.A.M., Manjesh, R., Dhanusha, M., 2019. Fruit recognition using image processing. Int. J. Eng. Res. Technol. (IJERT).

Padilla, R., Passos, W.L., Dias, T.L.B., Netto, S.L., da Silva, E.A.B., 2021. A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics 10 (3), http://dx.doi.org/10.3390/electronics10030279, URL https://www.mdpi.com/2079-9292/10/3/279.

Pereira, C.S., Morais, R., Reis, M.J., 2018. Pixel-based leaf segmentation from natural vineyard images using color model and threshold techniques. In: International Conference Image Analysis and Recognition. Springer, pp. 96–106.

Reis, M.J., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., Valente, A., Baptista, J., Ferreira, P.J.S., Cruz, J.B., 2012. Automatic detection of bunches of grapes in natural environment from color images. J. Appl. Log. 10 (4), 285–290.

Seng, K.P., Ang, L.-M., Schmidtke, L.M., Rogiers, S.Y., 2018. Computer vision and machine learning for viticulture technology. IEEE Access 6, 67494–67510.

Shen, L., Chen, S., Mi, Z., Su, J., Huang, R., Song, Y., Fang, Y., Su, B., 2022. Identifying veraison process of colored wine grapes in field conditions combining deep learning and image analysis. Comput. Electron. Agric. 200, 107268.

Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., Marinello, F., 2022. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. Agronomy 12 (2), 319.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

Wu, Z., Shen, C., Van Den Hengel, A., 2019. Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognit. 90, 119–133.

Xia, X., Xu, C., Nan, B., 2017. Inception-v3 for flower classification. In: 2017 2nd International Conference on Image, Vision and Computing. ICIVC, IEEE, pp. 783–787.

Yu, J., Sharpe, S.M., Schumann, A.W., Boyd, N.S., 2019. Deep learning for image-based weed detection in turfgrass. Eur. J. Agron. 104, 78–84.