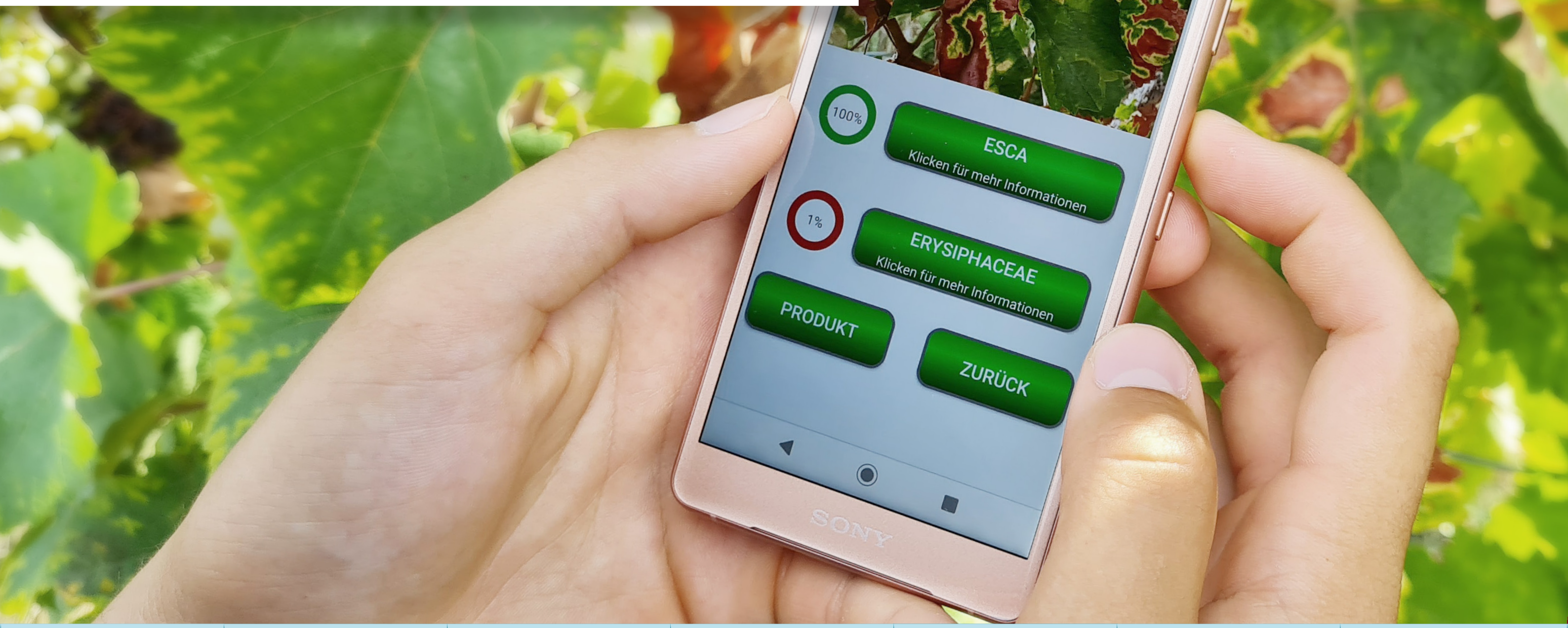
Paper 10 / 2022 doi: 10.7795/320.202210

[JUNGE wissenschaft]

JungforscherInnen publizieren
online | *peer reviewed* | original

Verlag:
Physikalisch-
Technische
Bundesanstalt
PTB



Mathematik & Informatik

Vine Leaf Disease and AI

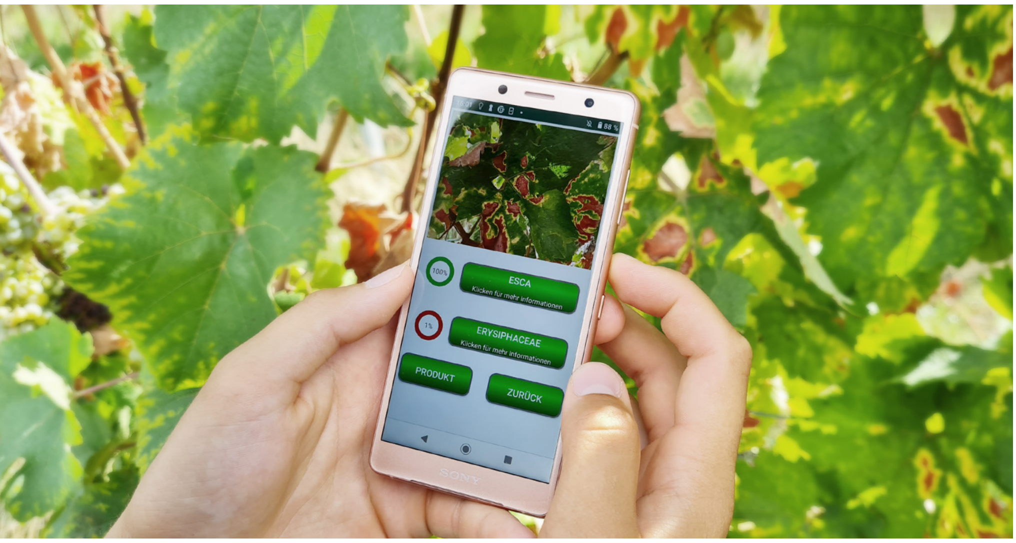
Detection of grapevine leaf diseases based on RGB-images, deep learning and its integration in a mobile application

This project enables early grapevine leaf disease identification on grape leaves by cell phone images, thereby allowing a precise usage of pesticides. The application is based on artificial intelligence (AI) which is trained to detect and differentiate the most common diseases. A continuous update of the extent and geographical location of disease spreading gives further valuable information to the winemakers using the application.

DIE JUNGFORSCHER

Maria-Theresa Licka (2003),
Elisabeth-von-Thadden-Schule,
Heidelberg
Mario Schweikert (2003),
Leibniz-Gymnasium,
Neustadt an der Weinstraße

Eingang der Arbeit:
4.2.2022
Arbeit angenommen:
20.6.2022



Vine Leaf Disease and AI

Detection of grapevine leaf diseases based on RGB-images, deep learning and its integration in a mobile application

1. Introduction

1.1 Primary Diseases on Grapevine Leaves

Grapevine cultivation has a very long tradition in Germany, wine production may date back as far as the celts [2]. The quality and annual yield of grapes directly reflect the "vitality" of the grapevines. Grapevines are vulnerable to numerous diseases caused by bacteria, fungi or viruses [3]. The rapid spread of these diseases can reduce the photosynthetic active area in large cultivation areas and by this quality and yield are significantly reduced. In some cases, a loss of productivity by death of entire plants is observed. Therefore, early detection of disease infestation is essential to take appropriate actions for plant protection.

In Germany, following the traditional grapevine agriculture, there are specific recommendations for the use of pesticides [4]. Currently more than 3,000 tons of chemical agents of all kinds are applied in the German grape vineyards annually for wine production. These pesticides and their residues pollute our environment and are also detectable in the wines [1].

1.2 Characteristics, Course of Infestation and Transmission Mode

Depending on environmental conditions, susceptibility of the host plant, and virulence of the pathogen, diseases differ in time and frequency of occurrence. Further, they have specific appearances and spreading characteristics depending on the type of pathogen [5]. The project was focused on the four most relevant disease infestations in spring and summer: powdery mildew, downy mildew, black wood disease and esca disease. In Table 1 the essential criteria of these most frequent analysed grapevine leaf diseases are presented.

1.3 Relevance of the Project

Wine production has had a significant economic relevance in Germany for many years. The average wine consumption per capita and year is about 28 litres [16]. In 2018, 10.3 million hectolitres of wine were produced in Germany [17]. Even though most of the wine produced is consumed domestically, around 1 million hectolitres of German wine are annually exported in over 100 countries worldwide [18]. The export value of German wine was approximately 305 million euros in 2019 [19]. After France, Spain, and Italy, Germany is the fourth-largest wine exporter in Europe and the eight-largest wine exporter worldwide [20]. These facts illustrate the economic factor "wine" in Germany, with wine consumption increasing worldwide.

A mix of up to ten different pesticides can be detected in German quality wines [15]. This quite shocking fact was already published in the German press in 2010. In 2015, the department of Viticulture in Siebeldingen, part of the federal Julius Kuehn-Institute for Agricultural Research, describes the European use of chemical substances in viticulture as "a very large quantity"; across Europe around 60 percent of the fungicides used in agriculture are spread in the grape vineyards – with a long-term average of about 90,000 tonnes per year [1]. Especially for fungal-like pathogens and diseases such as mildew the application of pesticides is very high.

doi: 10.7795/320.202210

Table 1: Powdery mildew, downy mildew, black wood disease, and esca disease under the following aspects: genus, features, course of infestation and transmission mode

Disease	Scientific name	Features	Course of infestation	Transmission mode
Powdery mildew 	*Erysiphaceae, Erysiphe necator* [6], *Oidium*[7]	White-grey spots consisting of myceli and conidiosphores develop on both sides of the leaf. The layer that forms on the leaf appears slightly moldy and mealy [7].	As a result of the infestation, the leaves often curl up, the flowers can no longer open, the fruits become hard, turn grey to black. Finally, the seeds may break, and the grapes are no longer suitable for harvest and consumption [7].	The wind carries the conidia and spreads of the disease for several kilometres [6].
Downy mildew 	*Plasmopara viticola*[7], *Peronospora*	Slightly purple or white-grey mealy coatings form on the lower side of the leaf. On the upper side of the leaf, either oil spots or red-coloured areas (see picture) are visible [7].	The consequence of the infestation is a wilting and dying of the leaves and in some cases even the whole young plant dies. The grapes become no longer suitable for harvest and consumption [7].	The wind carries the zoospores and spreads of the disease [7]
Black wood disease 	Bois noir, *Candidatus Phytoplasma solani* [8]	While a reddish to purple colouring of the leaves is characteristic for black wood disease in red-leaved varieties, the leaves of white grape varieties turn light green to yellow [8].	The curling of the leaves is a consequence of the infestation. The berries have an unripe and bitter taste, which makes them „useless" for consumption [8].	The disease infiltrates the sap ducts. The transmission occurs through a vector (carrier), e.g. cicadas or by grafting [8] [9].
Esca disease 	*Esca*	The most obvious characteristic sign of *Esca* disease on grapevines are the "tiger stripes" on the leaves. The leaf veins are usually surrounded by either a yellow or a red fringe. Another symptom can be the early yellow leaf colouring, which prematurely occurs in summer due to the infestation [9][10][11].	The leaves dry out and fall off as result of the infestation. The grapes also shrivel up and turn brown, comparable to leather berries, and burst. The whole grapevine will dry out and die subsequently [12] [13].	The disease is located in the grapevine's sap ducts and occurs via a vector (carrier), e.g., by cicadas and by grafting [14].

In an effort to achieve sustainable plant protection of grapevines by a reduction of used pesticides, precise knowledge on the occurrence of the disease infestation is important. This project therefore aims to enable an easier and earlier detection of infected grapevines, which can reduce the use of pesticides and increase the quality and quantity of grape harvest and the wine production. This would further lead to the environment and nature being less polluted with harmful chemical substances and agriculture becoming more sustainable with benefits for humans, animals and plants living on our planet.

1.4 AI Classification of Diseases Based on Smartphone-Quality Photos

Artificial Intelligence (AI) has achieved great results in image classification [21] with smartphone-quality photos [22]. Given the wide availability, using smartphone pictures coupled with AI to detect diseases early could help reduce the amount of pesticides to be used [23]. Therefore, we developed a smartphone app based on an artificial neural network to detect and differentiate grapevine leaf infestations. The immediate feedback of the AI based on the input image data enables the winegrowers to apply appropriate measures at early stages of infestation. This would prevent a large-scale spread of grapevine leaf diseases and subsequently reduces the large-scale need of pesticides.

The "Vine Leaf Disease and AI" app can easily be installed free of charge on an Android or iOS mobile phone. The users, mostly commercial winemakers or (seasonal) workers but also hobby winemakers, take a picture of a potential infected grapevine leaf. The app immediately gives feedback to the user, which grapevine leaf disease is the underlying cause of the infestation and in percentage how accurate the classification of the AI is. Additionally, the user receives an actual cartographical map showing the extent and type of regional and national grapevine leaf diseases spreads, a kind of "alarm system". The data will be stored on a server, so annually, monthly, or even weekly and daily data is constantly available for the user, for example to compare the regional amount of grapevine leaf disease infection with other years or seasons. To ensure data protection, all data is transmitted and stored pseudonymised in a database. By accepting the data protection agreement after the installation, the user acknowledges that the app requests the phone's location. Not only the installation of the app but also its use is very simple and intuitive. A frequent use of the app by many winegrowers will continuously train the underlying AI and make it even more precise in the future. Currently, about 300 users employ the application already.

2. Methodology

2.1 The Dataset

A web search of publicly available, suitable images of grapevine leaves was carried out and yielded no satisfying results. Therefore, more than 5,000 photos were taken of many infected grapevine leaves with a Samsung Galaxy M51, iPhone 12 Mini, and a Sony Xperia XZ2 Compact. Due to the rareness of many diseases, only the four most common and harmful leaf diseases were used for training. The dataset currently contains leaf images during different stages of infestation and was classified with the following diseases: powdery mildew, downy mildew, black wood disease, and esca disease. Additionally, healthy leaves without grapevine leaf disease infestation and images of the red and yellow autumn colouring were added. The total of seven classes were labelled by hand. In Table 2 the distribution of the data set is illustrated. For the evaluation during the training, 10 percent of the input data was split off as validation data. Furthermore, at least 100 images per class were used as test data. With this data, we were able to simulate how the model behaves on completely new data.

2.2 Data Pre-Processing

An image is represented as a matrix of pixel values, which corresponds to RGB from 0 to 255 over three matrices. The RGB-range was chosen so that colour differences can also be recognized. This creates an input size for the mesh of [224, 224, 3] To achieve the best possible results, the images were normalized between [-1,1] before training.

The images always have a typical background by taking a close-up of the grapevine leaf in the vineyard. To reduce the background as much as possible in the input image, Grabcut [24] and Haar cascades [25] were tested to extract

Table 2: Distribution of the data set by classes

Classes	Healthy	Red autumn color	Yellow autumn color	Powdery mildew	Downy mildew	Black wood	Esca disease	Others
Number of photos	523	590	651	564	663	524	778	972

the leaf data solely. To diversify the dataset and generate additional images, the following features of the photos were altered with data augmentation: brightness, zoom-range, shear-range, rotation, width and height change (see Fig. 1).

2.3 The Model

Since the model runs locally on mobile phones, a simple, accurate, but fast setup was necessary. A hyperparameter search program was created to find the balance between accuracy and speed. The program is based on the idea of a "GridSearch" [26] algorithm that systematically starts to train the model from a range of values and finally outputs the optimal values.

It is a classification with multiple classes via a Convolutional Neural Network (CNN) [27]. The basic layers of a CNN are:

- Convolutional layer: Extracting main information from the image with filters.
- Pooling layer: Reduces the result from the convolutional layer to the most important by pooling pixel values (e.g., maximum value, average value).
- Fully connected layer: Classification of the data from the preceding layers (one-dimensional vector $\vec{x}$ required, which is achieved via flattening layers) [28].

Transfer learning was applied to create a valid model. "MobileNet" [29] is based on a convolutional network. In addition to the convolutional, pooling, and fully connected layers described above, the following functions have been added to MobileNet:

- Batch normalization: Prevents the weights from becoming extremely large or small and speeds up the training.

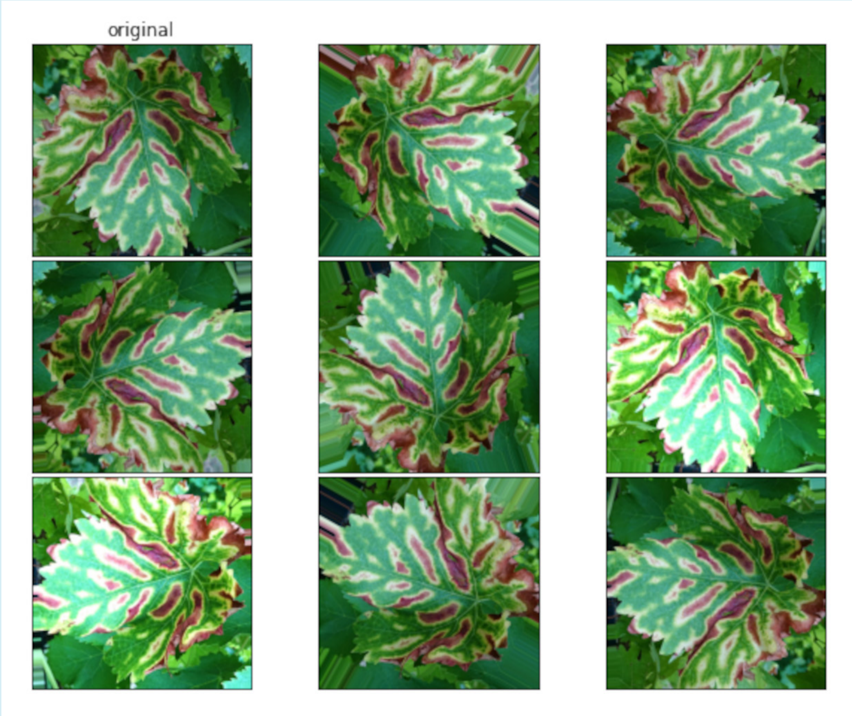


Fig. 1: Example for data augmentation. The picture on the upper left corner is the original, all other image appearances are slightly changed (for example rotated) with a TensorFlow generator. All on this way generated pictures were also used to train the model.

- Dropout: Deactivates random neurons in the training process to prevent overfitting, and the model develops a general disease pattern in each case [29].

The model uses the activation functions ReLU (Rectified Linear Unit) in the hidden layers and Softmax in the output layer.

ReLU: Formula (1)

The result of the Softmax function [30] is a vector in the value range [0,1] whose sum is one. The elements of the vector correspond to the probability for each class.

$$\text{Softmax}: \left(\phi_{Softmax}\left(\vec{x}\right)\right)_i = \frac{e^{x_i}}{\sum_{j=1}^{classes} e^{x_j}}$$

$x_i \in Z$ and classes $\in N$

In this case, the model includes the following seven classes: Real mildew, false mildew, esca, blackwood disease, red and yellow autumn color and healthy.

$\vec{x_i}$ corresponds to the input, which the function changes to a range of values from zero to one. If the input is negative, the output is close to zero, and the output is close to one for a large input.

For the calculation in a neuron, the weights vector $\vec{w}$ and the scalar bias are required, which were optimized with the gradient descent algorithm.

$$\phi_{ReLU}\left(\vec{x}\right) = \max\left(0, x_i\right) = \begin{Bmatrix} x_i \; for \; x_i \geq 0 \\ 0 \; for \; x_i < 0 \end{Bmatrix}$$

Formula (1)

$$result_i = (\phi(w \cdot input + bias))_i \qquad [32]$$

or transferred to all neurons, where n is the number of neurons in the layer:

$$\overrightarrow{result} = \phi\left(bias + \sum_{i=0}^{n} w_i \cdot input_i\right)$$

The cost function categorical cross entropy is based on the "labels" of our dataset and the most likely class "prediction" according to our model:

Formula (2)

The gradient descent method was used, which various optimizers can implement (e.g. Adam, SGD). The gradient (first derivative of the cost function) [34] shows in which direction the function rises the most, and the negative gradient shows in which direction the function falls. This gradient vector is normalized to length one and multiplied by the *learning rate* (step size). If the *learning rate is* too large, the global minimum may be skipped, and a local minimum found. If the step size is too small, the model can get stuck in a local minimum.

Here, the threshold value (bias) and the weight vector are among the adjustable hyperparameters. The result of the neuron is transferred to the next layer via the connections. The activation function normalizes the neuron's output in the process, as it can otherwise become arbitrarily large. In total, Keras MobileNet consists of 92 layers [29]. The training process was logged for both models with the callback's early stopping and checkpoints.

a) b)

Fig. 2: Picture of grapevine leaf (left) and demolished grapevine leaf after applying the segmentation algorithm Grabcut (right), which replaces all background pixels with black pixels.

$$L(\overrightarrow{Prediction}) = -\sum_{i=1}^{classes} label_i \cdot \ln(Prediction_i) \qquad [33]$$

Formula (2)

2.4 Training Set-up

We used Python 3 as a programming language and the following libraries: TensorFlow [35], Keras [36], OpenCV [37], NumPy [38] and Scikit-learn. A GPU-enabled Anaconda environment with CUDA was used, and training took place on a GeForce GTX 1050 TI 4GB.

2.5 The App

The app was programmed with ExpoGo [39] to achieve cross-platform compatibility. Before the photo is passed to the model, it must be cropped and normalized. This must be done in the same way as in the data pre-processing for the training data. This involves looping through each pixel of the matrix and performing the following calculation for float values:

$$\frac{PixelValue - 127}{128}$$

The model then runs locally.

2.6 The Server

The server is based on a Python Flask Server [40]. First, a CronJob (regularly executed program) was implemented,

Table 3: Validation accuracy and training time depending on the frozen layers

Frozen layers	None	3	12	20
Validation Accuracy after epoch 15	0.88	0.90	0.96	0.93
Time needed in minutes	25	23	25	23

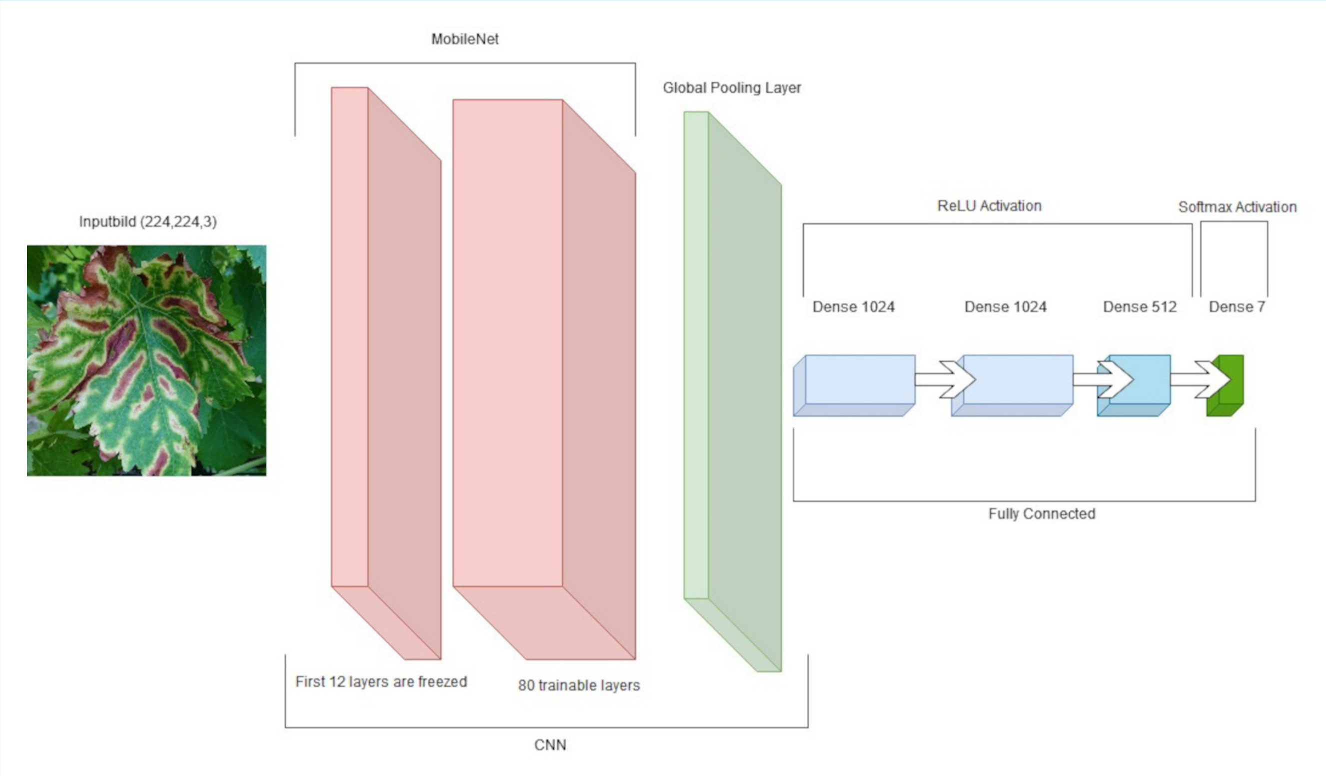


Fig. 3: After the image was processed by the MobileNet and Global Pooling Layer, it passes three fully connected layers before it activates the Softmax function in the last layer.

which creates plots. The four most frequent leaf diseases, autumn color and healthy of the current and last month and year are created and made available via the Flask server. Furthermore, the server accepts the post requests from the app and adds information from new leaf analyses or creates a plot with the four most common leaf diseases in a 10-kilometers radius on post request. The server saves new information in a Mongo Database [41]. It also converts the coordinate data to a city using the OpenStreetMap API [42], if possible in such a way that the user knows which region the plot is from. Debugging has taken place with Postman Agent [43], which simplifies the sending of API requests.

3. Results

3.1 Grabcut and Haar Cascades

As can be seen in Fig. 2, Grabcut did not result in satisfactory results, often removing discolorations of the leaves that were recognized as part of the background. Similarly, Haar cascades did not produce meaningful results as the dataset was not large enough. Therefore, the full images, including background, were used for the further training and evaluation of the model.

3.2 Machine Learning Model

MobileNet was placed in front of the fully connected layers (see Fig. 3). The MobileNet's fully connected output layer was replaced with our own layers with the neuron count

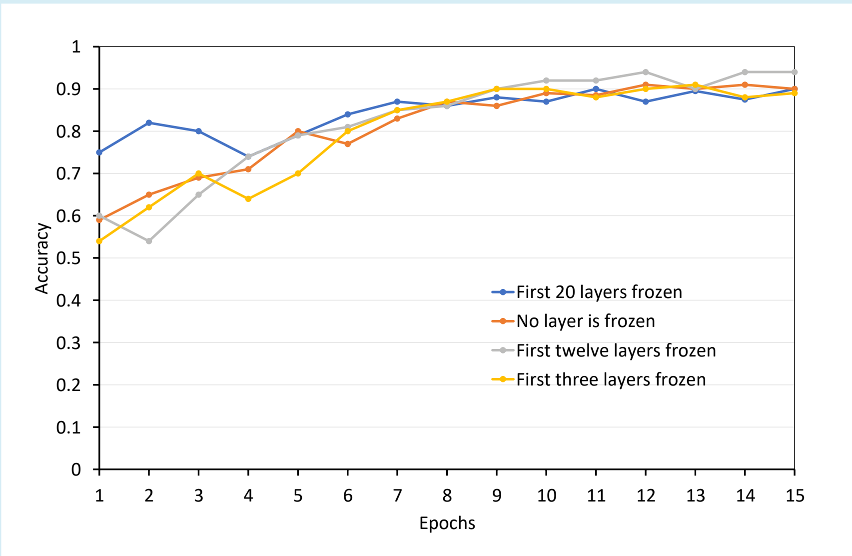


Fig. 4: Validation accuracy in dependence of frozen layers

doi: 10.7795/320.202210

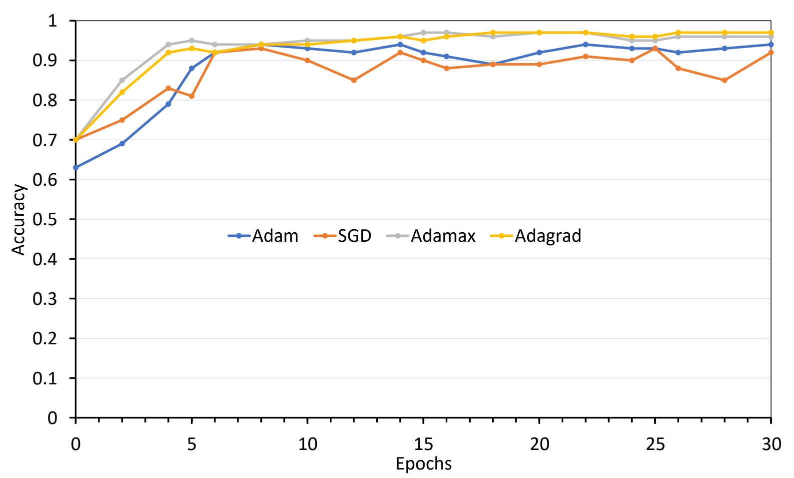


Fig. 5: Validation accuracy figure of different optimizers over 30 epochs

1024 → 1024 → 512 → 7 (class number), and the activation ReLU and Softmax. As MobileNet was pretrained, specifically to recognize basic structures in the first layers and details in the later ones, it was investigated how many of the initial first layers should be frozen to maximize accuracy. This then allowed for only changing and adapting the detailed last layers for the disease recognition.

In Fig. 4, the number of frozen layers only suggests a largely varying effect on the accuracies in the first epochs. Nevertheless, after 15 epochs, the distribution in Tab. 3 emerges as a function of the frozen layers.

While the training times change only minimally (Tab. 3) and can be regarded as natural fluctuations or standard deviations, the validation accuracy differs significantly depending on the number of frozen layers. This is because basic shapes are recognized in the first layers and details in the last layers. However, these details are not adapted to our leaf dataset and have to be retrained. The first twelve layers have been frozen for our final model because this setting leads to the best result.

The primary objective is to find the position of the global minimum in the cost function. If the prediction is close to the actual value, the loss is small. The further apart the values are, the greater the loss or they become arbitrarily large, as can be seen in Tab. 4. The sum sign can be omitted since this is a classification problem and the *labels* (categories) or only one class receive the value one for the actual value.

The optimizer also plays a crucial role in the training process regarding the time it takes to find the optimum. The optimizers Adam, SGD, Adamax, and Adagrad [44] were tested and the results have been recorded for 30 epochs (see Fig. 5).

Even if all optimizers start at different accuracies, sooner or later, they all lead to a high accuracy (see Fig. 5 and Tab. 5).

Adam [44] was used, although the fluctuations of the validation accuracy had to be taken into account. Adam calculates the learning rate for each parameter and has a momentum. The momentum uses not only the current gradient but also the gradient of the last optimization steps.

In Fig. 6 it is clear to see that MobileNet already starts with a training accuracy of 70 percent. This can be attributed to the fact that it can transfer learned patterns. After a final training until the convergence of the loss with 53 epochs, the values shown in Tab. 6 and Tab. 7 on the seven classes downy and powdery mildew, black wood, esca, red and yellow autumn blush, and healthy were achieved.

A model has been sufficiently trained when it no longer randomly predicts the label but chooses it by using the features. With seven classes, our model must have an accuracy above

$$\frac{100\%}{\textit{Number of classes}} = \frac{100\%}{7} \approx 14.29\%$$

to be classified as non-random.

It is also helpful to create a confusion matrix of the predicted and actual labels (see Fig. 7) to analyse which diseases have been classified incorrectly.

Table 4: Insertion of example values in categorical cross entropy

Real value	Forecast	Used in categorical cross entropy	Value
1	1	$-1 \cdot \ln(1)$	0
1	0.75	$-1 \cdot \ln(0.75)$	~0.288
1	0.5	$-1 \cdot \ln(0.5)$	~0.693
1	0	$-1 \cdot \ln(0)$	$+\infty$

Table 5: Comparison of different optimizers

Optimizer	Adam	SGD	Adamax	Adagrad
Validation Accuracy	0.98	0.94	0.96	0.97
Time needed in minutes	53	55	52	53

Table 6: Accuracy and loss of the MobileNet

	Accuracy	Loss
Training	0.97	0.1
Validation	0.95	0.16
Test	0.95	0.16

It can be seen in Fig. 7 that the diagonal stands out, because many predictions correspond to the real value. Only downy mildew is occasionally predicted as healthy. However, healthy leaves are not predicted as downy mildew. It is unclear why the model only confuses the disease for healthy and not the other way around. It could be the case because they look very familiar, especially at an early stage.

4. Discussion

Overall, the aim of this project was achieved as a smartphone app linked to a machine learning model was developed, which is available free of charge in the Play- and AppStore and was downloaded over 770 times (state: November 2022). It has been shown that it is possible to classify different grapevine leaf diseases through picture data (mobile images). Furthermore, the classifying algorithm is usable on smartphones.

The decision to use smartphone photos combined with an app for this project proves to be reasonable given the high usage and quality of smartphone cameras. Already in 2017, 85 percent of all pictures were taken with a smartphone. Digital cameras accounted for only 10.3 percent, while tablets made up 4.7 percent of all pictures taken [46].

It is not surprising that the number of photos taken has almost doubled since 2013 as the quality of smartphone images is getting increasingly better. In addition, more and more people own a smartphone that they always have on hand, unlike a tablet or digital camera [47]. Additionally, the quality of smartphone pictures has evolved very fast. The pioneer of the first generation of smartphones with photo and video functions took pictures with a resolution of 0.1 megapixels (100,000 pixels). Nowadays, modern smartphones can take even better pictures than normal cameras. The modern 4k resolution (also called Ultra-HD) displays films with 3840 to 2160 pixels [48].

MobileNet has been used as an algorithm because it was important to run the model locally on the smartphone in order to use it directly in the grape vineyard, and to also allow for a use without internet connection. The neural network is not only fast, straightforward and has a light depth, but it also needs little storage space and has a low power consumption. Another benefit of using MobileNet is,

Table 7: Calculating the metrics of the model [45]

Classes	Healthy	Yellow autumn colouring	Red autumn colouring	Powdery mildew	Downy mildew	Black wood	Esca
Precision	0.86	1	1	0.99	0.9	1	0.93
Recall	0.99	0.98	1	0.78	0.99	0.92	0.99

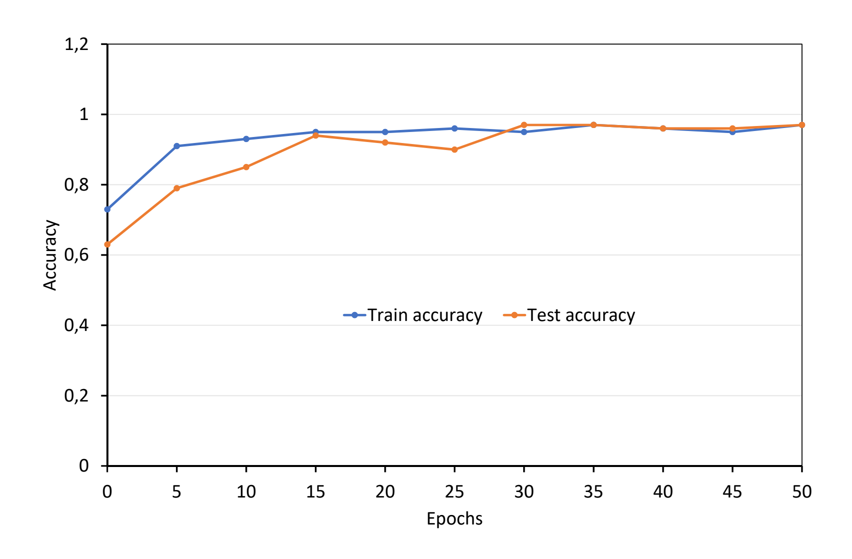


Fig. 6: Accuracy figures over 50 epochs

kind. Although the dataset already contains images of diseases at the earliest stage they can be classified, it would be interesting to create a method and corresponding model allowing to identify already infected grapevine leaves even if the infestation is not yet visible for humans. To achieve this, plants would have to be observed over a long term. Generally, it is to be expected that a large number of images will produce better results. Considering that many natural influences can change the appearance of leaves, such as sunlight or raindrops on the leaves, it is necessary to add more complex situations. On the one hand, the image database can be enlarged by adding more data to each class, and on the other hand, it can be diversified through new classes.

that the model was already pretrained and had learned to recognize basic structures. Even though the accuracy of MobileNet is relatively high (in general), it is possible that the accuracy would rise even more with a non-local working algorithm specialized on this task. Summarizing MobileNet was a good choice because of its effectively maximized accuracy, while allowing for limited resources, for an on-device or embedded application [54].

While the model achieves an accuracy of 95 percent, there is still an occasional risk of confusion between healthy leaves and those infested with powdery mildew (see Fig. 7). This is the case when the powdery mildew infestation is sparse and only shows a marginal fungal layer. Instead of using only one multi-class classification model, an extra classifier only differentiating between those two classes could expand the classification pipeline to reduce this problem. An approach to use “One-vs-One” or “One-vs-Rest” classifiers for multi-class classification can boost the accuracy and decrease mix-ups [55].

Because a data augmentation algorithm was employed, we could create a strong proof of concept with a relatively small image database. Further reasons for the stable functionality are high-quality images and very distinct disease patterns. The recognition of barely visible symptoms is currently challenging to realize. New approaches of capturing more wavelengths and a more precise distinction between the different diseases with hyperspectral or infrared should be considered. Accuracy and utility can be remarkably improved with more pictures of this kind.

When expanding the dataset, the read-in process can be further optimized via a Tensorflow Input Pipeline (input method) [52]. Using the TensorflowRecord [53] format, image objects are saved in a binary format to speed up the loading process. Large amounts of data can be stored in smaller files, named shards, to reduce memory workload. Since the number of images is

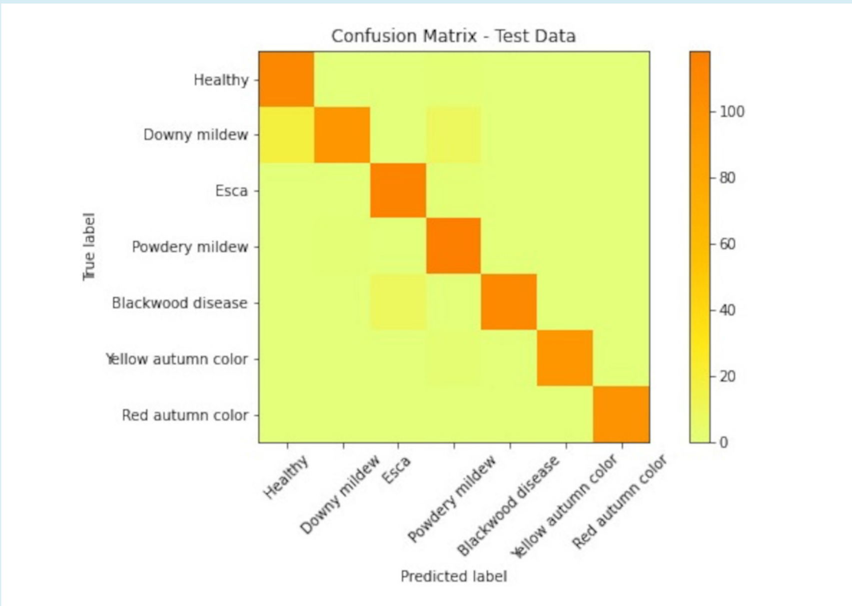


Fig. 7: The confusion matrix on test data depicts which classes are mixed up.

doi: 10.7795/320.202210

not yet too large, the conceptualization does not currently offer a significant advantage because it is more time-intensive to implement.

Numerous papers prove other use cases of plant disease recognition with mobile phone images and machine learning. While the Plant Village [49] data set is widely used as a data basis [50], we were able to create a data set with more grapevine leaf disease classes and in an environment with different backgrounds. The researchers of "Soybean Plant Disease Identification Using Convolutional Neural Network." [51] also came to the conclusion that the implementation of a segmentation algorithm does not allow any advantage in recognition accuracy. Grayscale images, also experimentally investigated in [51] to avoid bias from external image influences in the model, resulted in significantly lower accuracy, which is why the researchers decided to use the data set recorded in an uncontrolled environment. Another example is the paper "Automated identification of sugar beet diseases using smartphones", [23] which is also based on RGB smartphone images. Still, it describes an additional feature extraction step with colour filters before classification, which improves their result. As a meaningful evaluation of the results, their classification model was compared with sugar beet experts to demonstrate the advantage of the machine learning approach.

5. Summary

A portable app with a local running model was created to take pictures of the grapevine leaves and allow users a classification of possible diseases directly in the vineyard. An instructional video for the installation and use of the app has been provided for users (professional and hobby winemakers) on YouTube (YouTube channel: INFOrmAtIc Teens). In addition, the app includes information on the current local and national spread of grapevine leaf disease infestations as well as an overview of the temporal course of spread patterns in the past. With the app, photos were classified within ten seconds with a 97 percent accuracy on training data. It was shown that four different grapevine diseases, the autumn colours, and healthy leaves can be accurately identified using smartphone-quality photos. Since misclassifications are still possible, the dataset will be further extended and annotated. However, overall this project has shown that artificial intelligence combined with mobile devices can aid winemakers to precisely diagnose diseases in the vineyard as a basis for planning plant protection strategies. Therefore, the app provides a valuable opportunity for winegrowers to reduce the need and use of pesticides in the grape vineyard in the future – with more sustainable grape products and a habitat that will be less polluted with environmental toxins.

Acknowledgements

During our project we received support from the Dienstleistungszentrum Ländlicher Raum (DLR), the Julius Kühn-Institute (JKI), Pollichia. We were guided by experts from the DLR in classifying the various leaf diseases.

The project is supervised by the Elisabeth-von-Thadden-Schule, by Matthias Schenkel, the Jugend forscht AG Neustadt, by Mr. Sergej Buragin and by Carina Kaltenbach.

Sources and Literature

[1] Walther Rosenberger (2015). Pflanzenschutzmittel im Wein Winzer schwingen die chemische Keule. Stuttgarter Nachrichten.; https://www.stuttgarter-nachrichten.de/inhalt.pflanzenschutzmittel-im-wein-winzer-schwingen-die-chemische-keule.af7f64bd-b0cc-417c-8165-5bb9898ae95c.html

[2] Karl-Josef Gilles: Bacchus und Sucellus 2000 Jahre römische Weinkultur an Mosel und Rhein. Rhein-Mosel-Verlag, Briedel, 1999.; https://www.planet-wissen.de/natur/fluesse_und_seen/die_mosel/pwieweinbauandermosel100.html

[3] Günter Schruft, Hanns-Heinz Kassemeyer „Krankheiten und Schädlinge der Weinrebe" Verlag Th. Mann, Gelsenkirchen; https://www.syngenta.de/file/26781/download?token=EH0IdJsB

[4] Jochen Kramb (last updated 22.10.2021). Pflanzenschutz im Weinbau

[5] Heide, Carl. Der Wein: Weinbau und Weinbereitung Chemie und Untersuchung des Weines Springer-Verlag, 2013.

[6] B. Altmayer, B. Fader, M. Harms, R. Ipach, U. Ipach, H.-P. Lipps, K.-J. Schirra, B. Ziegler (2010): Sachkunde im Pflanzenschutz (Weinbau). 6. überarbeitete Auflage. Dienstleistungszentrum Ländlicher Raum (DLR) Rheinpfalz, Abteilung Phytomedizin. Neustadt an der Weinstraße. And P.M. Kirk, P.F. Cannon, D.W. Minter and J.A. Stalpers CABI Europe - UK (Hrsg.) (2011): Ainsworth & Bisby's Dictionary Of The Fungi. 10. Auflage CPI Group (UK) Ltd. Croydon. ISBN 978-1-84593-933-5

[7] Viret, Olivier, Werner, Siegfried, Pierre-Henri, Dubuis, and Katia, Gindro. „Echter Mehltau". Eidg. Departement für Wirtschaft, Bildung und Forschung WBF, Agroscope, AMTRA, Version von 2014.

[8] Ipach, U. (2006): Die Schwarzholzkrankheit – Eine neue Gefahr für die Reben? Tagungsband 59. Weinbautage 2006 (DLR Rheinpfalz, Neustadt an der Weinstraße)

[9] Leppert, Christine. „Untersuchungen zum Auftreten der Schwarzholzkrankheit in Reben".

[10] Agroscope RAC Changins (O. Viret), & Agroscope FAW Wädenswil (W. Siegfried), A. (2004). Esca Esca, protection phytosanitaire maladies cryptogamiques.

doi: 10.7795/320.202210

[11] B. Altmayer, J. Eichhorn, B. Fader, A. Kortekamp, R. Ipach, U. Ipach, H.-P. Lipps, K.-J. Schirra, B. Ziegler (2013): Sachkunde im Pflanzenschutz (Weinbau). 8. Überarbeitete Auflage. Dienstleistungszentrum Ländlicher Raum (DLR) Rheinpfalz, Abteilung Phytomedizin. Neustadt an der Weinstraße.

[12] Fischer, Jochen. „Untersuchungen zur Esca-Erkrankung von Weinpflanzen: Pilz-Pflanze und antagonistische Interaktionen." Dissertation, Universitätsbibliothek Mainz, 2015.

[13] Christoph Hoffmann, Katja Schulze, Melanie Wiwiorra, Biologische Bundesanstalt für Land- und Forstwirtschaft, Institut für Pflanzenschutz im Weinbau, Bernkastel-Kues; Michael Fischer, Staatliches Weinbauinstitut Freiburg, in Zusammenarbeit mit Gerlinde Nachtigall, Referat für Presse und Information der BBA; Biologische Bundesanstalt für Land- und Forstwirtschaft Messeweg 11/12, 38104 Braunschweig; Informationsblatt der BBA: Esca der Weinrebe, 2007.

[14] Dr. Michael Fischer (2013). Informationsblatt des JKI: Esca-Krankheit der Weinrebe. (Bundesforschungsinstitut für Kulturpflanzen Messeweg 11/12, 38104 Braunschweig).

[15] Pestizide im Wein; Verseuchte Tropfen. Süddeutsche Zeitung 2010

[16] Frankfurter Allgemeine (2018). Es wird wieder mehr Wein getrunken. Article.

[17] Das Statistische Bundesamt (2018). Weinerzeugung in Deutschland 2020: Gut 2 % mehr Wein und Most als 2019. Statistic.

[18] Wirtschaft, L.P. Rheinland-Pfalz. Gold im Glas. https://mwvlw.rlp.de/de/themen/weinbau/#:~:text=Deutschland%20ist%20nach%20den%20gro%C3%9Fen,Weinexporte%20stammen%20aus%20Rheinland%2DPfalz. (last recall 22.10.2021).

[19] Statista (2021). Exportwert von Wein aus Deutschland in den Jahren 2000 bis 2020. Online-statistic.

[20] Verband Deutscher Weinexporteure e.V. (last recall 2021). Homepage Verband Deutscher Weinexporteure e.V.. Website.

[21] Sladojevic, Srdjan, Marko, Arsenovic, Andras, Anderla, Dubravko, Culibrk, and Darko, Stefanovic. "Deep neural networks based recognition of plant diseases by leaf image classification".Computational intelligence and neuroscience 2016 (2016).

[22] Ignatov, Andrey, Radu, Timofte, William, Chou, Ke, Wang, Max, Wu, Tim, Hartley, and Luc, Van Gool. "AI Benchmark: Running Deep Neural Networks on Android Smartphones." . In Proceedings of the European Conference on Computer Vision (ECCV) Workshops.2018.

[23] Hallau, L., M., Neumann, B., Klatt, B., Kleinhenz, T., Klein, C., Kuhn, M., Röhrig, C., Bauckhage, K., Kersting, A.-K., Mahlein, U., Steiner, and E.-C., Oerke. "Automated identification of sugar beet diseases using smartphones". Plant Pathology 67, no.2 (2018): 399-410.

[24] Rother, C., Kolmogorov, V., & Blake, A. (2004). GrabCut: Interactive Foreground Extraction Using Iterated Figure Cuts. ACM Trans. Figure., 23, 309-314. Implemented by OpenCV

[25] Adrian Rosebrock (2020). OpenCV GrabCut: Foreground Segmentation and Extraction. Pyimagesearch.

[26] LaValle, S. M., Branicky, M. S., & Lindemann, S. R. (2004). On the relationship between classical grid search and probabilistic roadmaps. The International Journal of Robotics Research, 23(7–8), 673–692.

[27] LeCun, Yann, Patrick Haffner, Léon Bottou, und Yoshua Bengio. „Object Recognition with Gradient-Based Learning". In Shape, Contour and Grouping in Computer Vision, 319. Berlin, Heidelberg: Springer-Verlag, 1999

[28] Shin, H., H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, und R. M. Summers. „Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning". IEEE Transactions on Medical Imaging 35, Nr. 5 (2016): 1285–98.

[29] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, und Hartwig Adam. „MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". CoRR abs/1704.04861 (2017).

[30] Nwankpa, Chigozie, Winifred Ijomah, Anthony Gachagan, und Stephen Marshall. „Activation Functions: Comparison of trends in Practice and Research for Deep Learning". CoRR abs/1811.03378 (2018). (Page 8, "E. Rectified Linear Unit (ReLU) Function")

[31] J. Nathanael Philipp (2016). Objekterkennung mit Hilfe von Convolutional Neural Networks am Beispiel ägyptischer Hieroglyphen. Master thesis.

[32] Avinash Sharma V (2017). Understanding Activation Functions in Neural Networks. Medium; The Theory Of Everything.

[33] Categorical crossentropy. (last recall 24.10.2021); https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy

[34] Kriesel, David. Ein kleiner Überblick über Neuronale Netze, 2007. (Page 64), https://www.dkriesel.com/science/neural_networks

[35] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, u. a. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015.

[36] Chollet, F., & others. (2015). Keras.; https://github.com/keras-team/keras

[37] Bradski, G. (2000). The OpenCV Library. Dobb's Journal of Software Tools.

[38] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del R\'io, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy and Warren Weckesser, Hameer Abbasi, & Christoph Gohlke and Travis E. Oliphant (2020). Array programming with NumPy. Nature, 585(7825), 357–362.

[39] https://expo.dev/client 24.10.2021 11:55

[40] Grinberg, M. (2018). Flask web development: developing web applications with python. " O´Reilly Media, Inc."

[41] https://www.mongodb.com/ (last recall 11.08.2021 14:05)

[42] OpenStreetMap contributors. (2017). Planet dump retrieved from. https://www.openstreetmap.org (Using API)

[43] www.postman.com/ (last recall 11.08.2021 14:00)

[44] Optimizer Source: Ruder, Sebastian. "An overview of gradient descent optimization algorithms." (2016).

[45] Sokolova, Marina, Nathalie, Japkowicz, and Stan, Szpakowicz. "Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation." (2006).

[46] Matthias Olschewski, „Grafik zeigt: So viele Fotos werden jedes Jahr auf der Welt geschossen". Business Insider (2017).

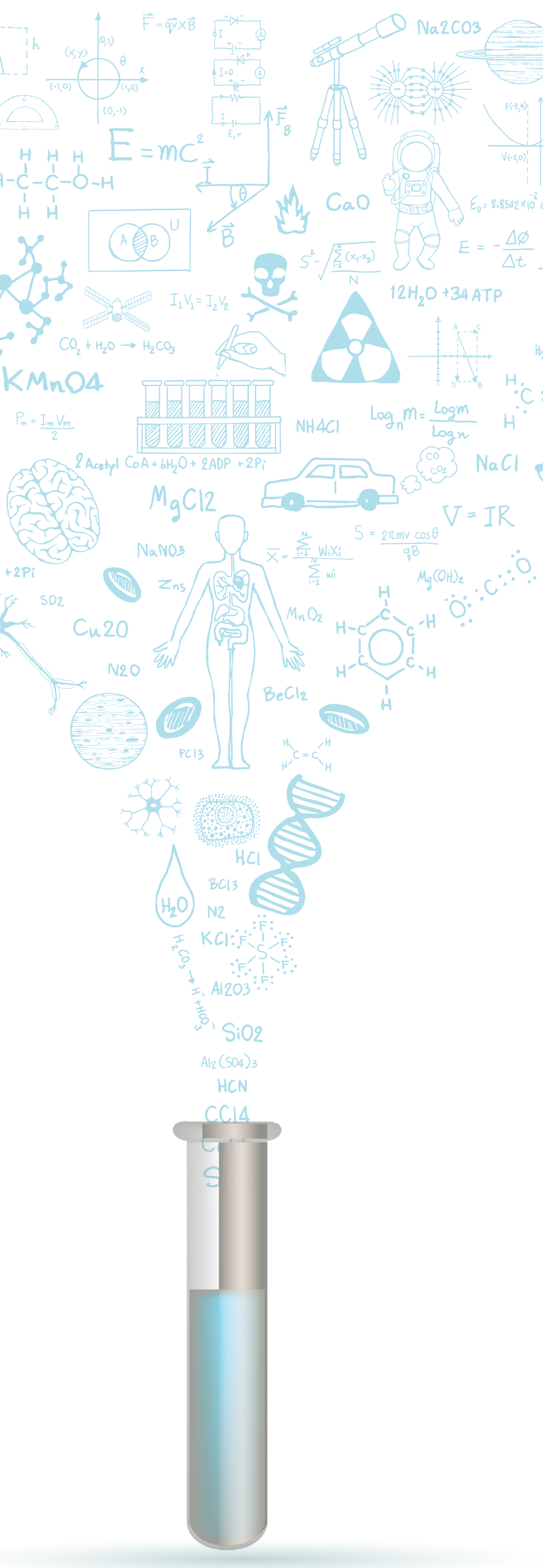
doi: 10.7795/320.202210

[47] Matthias Olschewski, „So viele Bilder werden jedes Jahr auf der Welt geschossen“. Gründerszene (2017).
[48] Marc Fleischmann,“Entwicklung der Handy-Kamera: Vom ersten Foto bis Hollywood“. Online article: IT-ZOOM (2021).
[49] David P. Hughes,, and Marcel Salathé. “An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing”.CoRR abs/1511.08060 (2015).
[50] Mohanty, Sharada P., David P., Hughes, and Marcel, Salathé. “Using Deep Learning for Image-Based Plant Disease Detection”. Frontiers in Plant Science 7 (2016).
[51] Wallelign, Serawork, Mihai, Polceanu, and Cedric, Buche. “Soybean Plant Disease Identification Using Convolutional Neural Network.” In FLAIRS-31 (pp. 146–151).2018.
[52] tf.data: Create TensorFlow Pipelines. (last recall 24.10.2021); tensorflow.org
[53] tf Record: Load TF-Records. (last recall 15.01.2022). tensorflow.org/tutorials/load_data/tfrecord.
[54] Abhijeet Pujara, “Image Classification With MobileNet”. Analytics Vidhya, Medium (2020); https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470
[55] Varpa, Kirsi, Henry, Joutsijoki, Kati, Iltanen, and Martti, Juhola. “Applying one-vs-one and one-vs-all classifiers in k-nearest neighbour method and support vector machines to an otoneurological multi-class problem”. Studies in health technology and informatics 169 (2011): pp. 579-583.

doi: 10.7795/320.202210

Publiziere auch Du hier!

Forschungsarbeiten von Schüler/Inne/n und Student/Inn/en



In der Jungen Wissenschaft werden Forschungsarbeiten von SchülerInnen, die selbstständig, z. B. in einer Schule oder einem Schülerforschungszentrum, durchgeführt wurden, veröffentlicht. Die Arbeiten können auf Deutsch oder Englisch geschrieben sein.

Wer kann einreichen?

SchülerInnen, AbiturientInnen und Studierende ohne Abschluss, die nicht älter als 23 Jahre sind.

Was musst Du beim Einreichen beachten?

Lies die Richtlinien für Beiträge. Sie enthalten Hinweise, wie Deine Arbeit aufgebaut sein soll, wie lang sie sein darf, wie die Bilder einzureichen sind und welche weiteren Informationen wir benötigen. Solltest Du Fragen haben, dann wende Dich gern schon vor dem Einreichen an die Chefredakteurin Sabine Walter.

Lade die Erstveröffentlichungserklärung herunter, drucke und fülle sie aus und unterschreibe sie.

Dann sende Deine Arbeit und die Erstveröffentlichungserklärung per Post an:

Chefredaktion Junge Wissenschaft
Dr.-Ing. Sabine Walter
Paul-Ducros-Straße 7
30952 Ronnenberg
Tel: 05109 / 561508
Mail: sabine.walter@verlag-jungewissenschaft.de

Wie geht es nach dem Einreichen weiter?

Die Chefredakteurin sucht einen geeigneten Fachgutachter, der die inhaltliche Richtigkeit der eingereichten Arbeit überprüft und eine Empfehlung ausspricht, ob sie veröffentlicht werden kann (Peer-Review-Verfahren). Das Gutachten wird den Euch, den AutorInnen zugeschickt und Du erhältst gegebenenfalls die Möglichkeit, Hinweise des Fachgutachters einzuarbeiten.

Die Erfahrung zeigt, dass Arbeiten, die z. B. im Rahmen eines Wettbewerbs wie **Jugend forscht** die Endrunde erreicht haben, die besten Chancen haben, dieses Peer-Review-Verfahren zu bestehen.

Schließlich kommt die Arbeit in die Redaktion, wird für das Layout vorbereitet und als Open-Access-Beitrag veröffentlicht.

Was ist Dein Benefit?

Deine Forschungsarbeit ist nun in einer Gutachterzeitschrift (Peer-Review-Journal) veröffentlicht worden, d. h. Du kannst die Veröffentlichung in Deine wissenschaftliche Literaturliste aufnehmen. Deine Arbeit erhält als Open-Access-Veröffentlichung einen DOI (Data Object Identifier) und kann von entsprechenden Suchmaschinen (z. B. BASE) gefunden werden.

Die Junge Wissenschaft wird zusätzlich in wissenschaftlichen Datenbanken gelistet, d. h. Deine Arbeit kann von Experten gefunden und sogar zitiert werden. Die Junge Wissenschaft wird Dich durch den Gesamtprozess des Erstellens einer wissenschaftlichen Arbeit begleiten – als gute Vorbereitung auf das, was Du im Studium benötigst.

doi: 10.7795/320.202210

Richtlinien für Beiträge

Für die meisten Autor/Inn/en ist dies die erste wissenschaftliche Veröffentlichung. Die Einhaltung der folgenden Richtlinien hilft allen – den Autor/innen/en und dem Redaktionsteam

Die Junge Wissenschaft veröffentlicht Originalbeiträge junger AutorInnen bis zum Alter von 23 Jahren.

- Die Beiträge können auf Deutsch oder Englisch verfasst sein und sollten nicht länger als 15 Seiten mit je 35 Zeilen sein. Hierbei sind Bilder, Grafiken und Tabellen mitgezählt. Anhänge werden nicht veröffentlicht. Deckblatt und Inhaltsverzeichnis zählen nicht mit.
- Formulieren Sie eine eingängige Überschrift, um bei der Leserschaft Interesse für Ihre Arbeit zu wecken, sowie eine wissenschaftliche Überschrift.
- Formulieren Sie eine kurze, leicht verständliche Zusammenfassung (maximal 400 Zeichen).
- Die Beiträge sollen in der üblichen Form gegliedert sein, d. h. Einleitung, Erläuterungen zur Durchführung der Arbeit sowie evtl. Überwindung von Schwierigkeiten, Ergebnisse, Schlussfolgerungen, Diskussion, Liste der zitierten Literatur. In der Einleitung sollte die Idee zu der Arbeit beschrieben und die Aufgabenstellung definiert werden. Außerdem sollte sie eine kurze Darstellung schon bekannter, ähnlicher Lösungsversuche enthalten (Stand der Literatur). Am Schluss des Beitrages kann ein Dank an Förderer der Arbeit, z. B. Lehrer und Sponsoren, mit vollständigem Namen angefügt werden. Für die Leser kann ein Glossar mit den wichtigsten Fachausdrücken hilfreich sein.
- Bitte reichen Sie alle Bilder, Grafiken und Tabellen nummeriert und zusätzlich als eigene Dateien ein. Bitte geben Sie bei nicht selbst erstellten Bildern, Tabellen, Zeichnungen, Grafiken etc. die genauen und korrekten Quellenangaben an (siehe auch Erstveröffentlichungserklärung). Senden Sie Ihre Bilder als Originaldateien oder mit einer Auflösung von mindestens 300 dpi bei einer Größe von 10 · 15 cm! Bei Grafiken, die mit Excel erstellt wurden, reichen Sie bitte ebenfalls die Originaldatei mit ein.
- Vermeiden Sie aufwendige und lange Zahlentabellen.
- Formelzeichen nach DIN, ggf. IUPAC oder IUPAP verwenden. Gleichungen sind stets als Größengleichungen zu schreiben.
- Die Literaturliste steht am Ende der Arbeit. Alle Stellen erhalten eine Nummer und werden in eckigen Klammern zitiert (Beispiel: Wie in [12] dargestellt …). Fußnoten sieht das Layout nicht vor.
- Reichen Sie Ihren Beitrag sowohl in ausgedruckter Form als auch als PDF ein. Für die weitere Bearbeitung und die Umsetzung in das Layout der Jungen Wissenschaft ist ein Word-Dokument mit möglichst wenig Formatierung erforderlich. (Sollte dies Schwierigkeiten bereiten, setzen Sie sich bitte mit uns in Verbindung, damit wir gemeinsam eine Lösung finden können.)
- Senden Sie mit dem Beitrag die Erstveröffentlichungserklärung ein. Diese beinhaltet im Wesentlichen, dass der Beitrag von dem/der angegebenen AutorIn stammt, keine Rechte Dritter verletzt werden und noch nicht an anderer Stelle veröffentlicht wurde (außer im Zusammenhang mit **Jugend forscht** oder einem vergleichbaren Wettbewerb). Ebenfalls ist zu versichern, dass alle von Ihnen verwendeten Bilder, Tabellen, Zeichnungen, Grafiken etc. von Ihnen veröffentlicht werden dürfen, also keine Rechte Dritter durch die Verwendung und Veröffentlichung verletzt werden. Entsprechendes Formular ist von der Homepage www.junge-wissenschaft.ptb.de herunterzuladen, auszudrucken, auszufüllen und dem gedruckten Beitrag unterschrieben beizulegen.
- Schließlich sind die genauen Anschriften der AutorInnen mit Telefonnummer und E-Mail-Adresse sowie Geburtsdaten und Fotografien (Auflösung 300 dpi bei einer Bildgröße von mindestens 10 · 15 cm) erforderlich.
- Neulingen im Publizieren werden als Vorbilder andere Publikationen, z. B. hier in der Jungen Wissenschaft, empfohlen.

doi: 10.7795/320.202210

Impressum

Junge Wissenschaft
c/o Physikalisch-Technische Bundesanstalt (PTB)
www.junge-wissenschaft.ptb.de

Redaktion
Dr. Sabine Walter, Chefredaktion
Junge Wissenschaft
Paul-Ducros-Str. 7
30952 Ronnenberg
E-Mail: sabine.walter@verlag-jungewissenschaft.de
Tel.: 05109 / 561 508

Verlag
Dr. Dr. Jens Simon,
Pressesprecher der PTB
Bundesallee 100
38116 Braunschweig
E-Mail: jens.simon@ptb.de
Tel.: 0531 / 592 3006
(Sekretariat der PTB-Pressestelle)

Design & Satz
Sebastian Baumeister
STILSICHER - Grafik & Werbung
E-Mail: baumeister@stilsicher.design
Tel.: 05142 / 98 77 89