Received 12 December 2024, accepted 7 February 2025, date of publication 17 February 2025, date of current version 21 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3542612

RESEARCH ARTICLE

Fresh or Rotten? Enhancing Rotten Fruit Detection With Deep Learning and Gaussian Filtering

LEOPOLD FISCHER-BRANDIES, (Member, IEEE), LUCAS MÜLLER, JUSTUS JOHANNES RIEGGER, AND RICARDO BUETTNER, (Senior Member, IEEE)

Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, 22043 Hamburg, Germany

Corresponding author: Leopold Fischer-Brandies (fb@hsu-hh.de)

This work was supported by the Open-Access-Publication-Fund of the Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg.

ABSTRACT We address the pressing issue of food waste by proposing a robust image classification model that can reliably detect rotten fruits. More than half of the fruit yield is lost along the supply chain, with post-harvest losses due to rottenness playing a pivotal role, as even a single decomposing piece can cause huge damage to nearby produce. Our transfer learning-based model uses the ResNet50 convolutional neural network architecture as a binary classification model to distinguish between fresh and rotten fruits. The model performance is enhanced with a Gauss filter and a dropout layer to ensure robustness and prevent overfitting. We achieve high accuracies beyond 99% on unseen test data, setting a new benchmark and outperforming previous efforts. Our work has theoretical and practical implications. To the best of our knowledge, we are the first to explore the use of Gauss filters to preprocess input images in fruit classification. We find that Gauss filters with small kernel sizes improve the performance of our model. Our research can improve post-harvest applications through automation. It can thus help reduce food waste, improve food safety, and reduce costs for growers, distributors, and retailers, thereby improving the overall efficiency of the supply chain.

INDEX TERMS Deep learning, Gaussian filter, ResNet50, transfer learning, food waste, supply chain efficiency, rotten fruit.

I. INTRODUCTION

The global demand for food is expected to increase by 35% to 56% by 2050 compared to 2010 [1]. Meanwhile, global warming makes food production more difficult, as it is expected to reduce crop frequency, complicate water supply, and diminish harvesting or planting time windows [2], [3], [4]. In the past, increasing global demand was primarily compensated by cropland expansion. However, this practice is unsustainable and causes environmental harm, besides increasing carbon emissions [5], [6]. Therefore, other ways to keep pace with rising food demand must be developed. Those involved in food production are responsible and interested in ensuring efficiency along the supply chain. As the population grows, urbanization rates increase, and food demand rises [7].

One significant lever that can help improve this situation is addressing the issue of food waste. In the case of fruit and vegetables, it is estimated that worldwide, more than 50% of the yield is being lost along the supply chain [8]. According to data from the United Nations Food and Agriculture Organization, over two billion tons of fresh fruits and vegetables were produced worldwide in 2021 [9]. Consequently, this implies that more than one billion tons of food are wasted before reaching consumers. Rotten crops

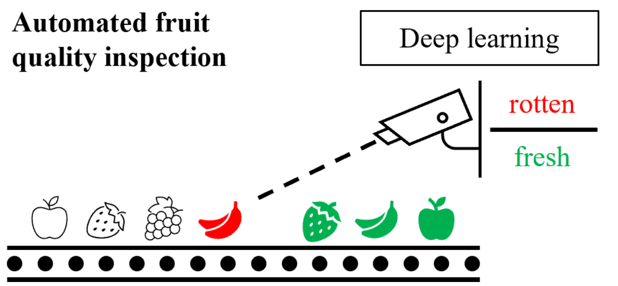

FIGURE 1. Automated rotten fruit detection using deep learning model as classifier.

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

are a primary factor related to this problem, as even a single decomposing fruit or vegetable can cause huge damage to the rest of the inventory [10], [11]. Therefore, reducing these post-harvest losses is a crucial part of pursuing a sustainable way of feeding a growing world population in the future [12]. Improving post-harvest technologies can be an important measure to achieve this goal. They can help reduce costs for growers, distributors, and retailers and enhance food safety while reducing food waste [13], [14]. Thus, such technologies can contribute to improving the overall efficiency of the supply chain. One of these post-harvest technologies is detecting rotten fruits and vegetables early by applying sophisticated artificial intelligence (AI) technology instead of relying on human labor [15]. The use of automated systems in detecting rotten fruits, in contrast to conventional methods such as the manual classification of fruit freshness, can be a cost-effective and efficient solution that can bring many benefits to different stakeholders along the supply chain [16]. In this context, computer vision and machine learning (ML) tools can have a significant influence in meeting the demand for more efficient food production [17], [18], [19]. The current challenge revolves around accurately classifying rotten and fresh fruits, a pivotal factor in enhancing supply chain efficiency and curbing food waste.

Machine learning techniques, particularly convolutional neural networks (CNN), present one way to address this problem, as CNN models have demonstrated promising results in various image classification methods [20]. Furthermore, there is a growing research focus on the automated identification of rotten fruit using AI. This heightened interest arises as novel computer vision methods that employ deep learning architectures advance, improving high-level scene comprehension and enhanced feature representation accuracy [21], [22]. Consequently, this research employs CNN models to address the problem of reliably identifying rotten fruits.

In previous research, several attempts have been made to achieve high accuracy in detecting damaged or decomposing fruit [20]. However, existing research is limited to using small, crafted, or non-peer-reviewed datasets or using only one or two fruits to train and validate their models. This negatively affects the robustness of the developed ML models and justifies further work on the problem of accurate classification of rotten and fresh fruits [23].

This paper aims to address and overcome this issue by creating a CNN image recognition and classification model using transfer learning. Specifically, the objective of this research is to develop a deep-learning model that can accurately distinguish between rotten and fresh fruits based on visual characteristics as depicted in figure 1. The model must be able to classify new images of fruits as either rotten or fresh, with a high level of accuracy and robustness, while also considering the imbalanced nature of the dataset and addressing any biases towards the majority class. The developed model should be able to extract features from the images automatically and learn the patterns that are associated with either class, such as discoloration or mold, even if these patterns are not explicitly defined by the user. To encompass the full spectrum of the trade-off between high performance and computational intensity, we first employed the three commonly used CNN architectures: VGG16, ResNet50, and MobileNetV3. Thereby, we also evaluate if models with less computational intensity yield similar promising results, enabling them to be potentially applicable in low-cost mobile applications, such as drones. Second, we apply Gaussian filters of varying sizes on the input images of the best-performing architecture to evaluate how such filters can improve the model's performance by regularizing the input.

To address the limitations of previous research, we use a large and balanced dataset of rotten and fresh fruits to train our model. Hence, we contribute to the existing body of knowledge by introducing a CNN model that can accurately classify the condition of several different fruits and demonstrating how CNN models can benefit from the addition of small Gaussian filters on the input layer. Furthermore, our approach utilizes a widely varied dataset comprising eight distinct types of fruits. This strategy is designed to mitigate the common issue of overfitting that is often observed in CNN models [24].

This research paper is structured as follows. In section II, we describe the application of machine learning technology in agriculture and compare existing approaches for the binary classification of fruit. The details of our CNN classification architectures, Gaussian filters, the applied dataset, preprocessing, and training and evaluation methods are then presented in section III. An analysis of the results follows in section IV. Section V then discusses the results in the context of the previously derived problem statement. We then conclude this paper by summarizing the most important aspects of our research, showing the limitations of our work, and highlighting opportunities for future research in section VI.

II. RESEARCH BACKGROUND

A. MACHINE LEARNING IN AGRICULTURE

ML is a powerful technology that enables a wide range of applications, and agricultural application is no exception. In recent years, there has been a substantial increase in scientific publications on the use of ML in agriculture. While the literature review conducted by Liakos et al. [18] identified 40 articles on the subject, that number quickly grew to 338 as shown by the review of Benos et al. [32], who repeated the methodology of Liakos et al. three years later.

In this paper, a model is introduced to differentiate between rotten and non-rotten fruit, thus constituting a binary classification model. An alternative to classification models is those that output numerical values through regression. For instance, Yu et al. [33] used a fully connected neural network to predict pears' firmness and soluble solid content using hyperspectral images. Both firmness and soluble solid content are numerical indicators of quality. Furthermore,

TABLE 1. Overview on related work.

Reference	Fruit	Dataset (Number of Images)	Classifier	Accuracy	Peer-reviewed Dataset?
Azizah et al. [25]	Mangosteen	RGB (120)	CNN	97.5%	Yes
Karakaya et al. [26]	Apples, Bananas, Oranges	RGB (1,200)	SVM	96.8%	No
Yogesh et al. [27]	Apples, Pears, Litchis, Mosambis, Pomegranates	RGB (2,000)	SVM	95.9%	No
Bhargava & Bansal [28]	Apples	RGB (13,851)	SVM	95.3%	Yes
Khoje et al. [29]	Guava, Lemons	RGB (1,340)	SVM	91.6%	Yes
Wang et al. [30]	Blueberries	Hyperspectral (557)	CNN (ResNet)	88.4%	Yes
Chandini & Maheswari [31]	Apples	RGB (75)	SVM	85.6%	Yes

instead of using binary classification, the classification of more than two classes can also be conducted. An example of this kind of approach is the work of Liu et al. [34], who used a support vector machine (SVM) based model to classify cucumbers into one of five categories (normal, watery, split/hollow, shriveled, and surface) using hyperspectral images. They achieved a classification accuracy of 91.10%.

The focus of the following section is on research that falls into the same category as our contribution, which is the binary classification of fruit.

B. APPLIED METHODS

The most commonly employed method in the application of ML in agriculture is the use of Artificial Neural Networks, especially convolutional neural networks (CNN) [32]. Support Vector Machines (SVM) are used as well, especially for binary classification. For instance, Bhargava and Bansal [28] build a model to separate between rotten and fresh apples. Their model is based on an SVM classifier trained on 13,851 red-green-blue (RGB) images. Bhargava and Bansal [28] achieved a classification accuracy of 95.27%.

Deep Learning techniques, especially CNNs, seem to outperform alternative methods in fruit classification. For instance, Wang et al. [30] compared the performance of different ML models in the classification of blueberries into rotten and fresh. The CNN used by Wang et al. [30], which is based on the ResNet architecture, demonstrated a superior classification accuracy with 88% compared to alternative ML methods; sequential minimal optimization (81%), multilayer perceptron (78%) linear regression (76%), random forest (73%) and bagging (71%) [30].

Furthermore, the findings of Karakaya et al. [26] support the hypothesis that CNNs deliver superior performance. Karakaya et al. [26] built an SVM model to detect rotten fruits. They compared five techniques for the feature extraction: Histogram Features (Hist), Gray Level Co-occurrence Matrix (GLCM), a combination of Hist and GLCM, Bag of Features (BoF), and a CNN (ResNet-50). They compared the performance of these techniques in separating rotten and fresh fruit. In this task, the features extracted by the CNN used by Karakaya et al. [26]; ResNet-50 outperformed the Hist. (93.83%), the GLCM (96.42%), the combination of Hist and GLCM (88.92%), and BoF (73.95%) with a classification accuracy of 96.72%.

C. DATATYPES

Common datatypes used in the related work are RGB and hyperspectral images. RGB images provide information to machine learning models about an object's geometry, texture, and color, which collectively constitute the basic characteristics of the object. [23]. The drawback of RGB images is that ML models based on them are sensitive to the background of the image and the lighting conditions [35]. In contrast, ML models based on hyperspectral images are less sensitive in this regard. However, they are much more costly to source since they require expensive sensors, increasing the setup cost of hyperspectral-based systems [23]. Furthermore, the processing of hyperspectral data requires significantly more computational power than that of other data types [23].

Overall, both RGB and hyperspectral images are used in binary fruit classification. Azizah et al. [25] utilized a CNN to distinguish between RGB images of "defective" and "fine" mangosteen, achieving a classification accuracy of 97.5%. Wang et al. [30] trained their models with several ML techniques in the classification of hyperspectral images of blueberries into either rotten or fresh.

D. DATA AMOUNT AND VARIETY

The majority of contributions in the binary classification between fresh and rotten fruit rely on one type of fruit for the training of their model. However, there are exceptions to that. Yogesh et al. [27] used 2,000 RGB Images from apples, pears, litchis, mosambis, and pomegranates to train their SVM-based model in the separation between rotten and fresh fruit. It remains unclear how their training dataset is composed between the different fruits. Yogesh et al. [27] solely present the accuracy of their model for the fruits apples (98.5%), pears (96.6%), and pomegranate (92.5%). The average classification accuracy for these three fruits is 95.87%.

Khoje et al. [29] used 1,340 RGB images of guavas and lemons to train and test their SVM-based model to distinguish between healthy and defective fruits. They achieved a classification accuracy of 91.42% for guavas and 91.72% for lemons, resulting in an average classification accuracy of 91.57%. However, it is important to note that the authors tested the model using the same dataset on which it was trained.

A shortcoming of existing research is that small datasets are often used. One example is the work of Chandini & Maheswari [31], who used 75 RGB images to train their SVM-based model, which separates rotten from fresh apples. They achieved a classification accuracy of 85.6 %. In table 1 we summarize the mentioned related research.

E. RESEARCH GAP

Generally, the application of deep learning techniques, especially CNN, has shown success in controlled environments. However, the application of such systems in the real world is lacking. A major problem hindering the implementation is the lack of robustness, meaning that the models work well with their respective training data but perform significantly worse in real-world applications since the real-world input data differs from the training data. A major contributor to this is the problem of overfitting [24]. To decrease overfitting and to create robust deep learning models, large and diverse datasets are required [23]. However, existing research mostly utilizes small or crafted datasets [23]. Furthermore, the diversity of the input data is limited by the fact that most of the time, only one and seldom two or three different kinds of fruits are used, which limits the diversity of the training data. Related to the lack of large and diverse datasets is the usage of non-peer-reviewed datasets. This is a problem since peer-reviewed datasets are critical to ensure a solid research foundation by questioning the methods of data collection and overall data quality and preventing possible biases toward research objectives [36].

In this context, to the best of the authors' knowledge, no study to date has investigated the problem of rotten fruit detection with a large and diverse, peer-reviewed dataset to achieve a highly accurate and robust CNN model. Consequently, the present research is necessary to contribute to the existing knowledge regarding real-world applicability by ensuring higher robustness and better performance.

III. METHODOLOGY

The objective of this study is to create a robust and accurate model for separating rotten and fresh fruits. We propose a CNN-based approach. To this extent, we first compare three CNN architectures while employing a transfer learning approach. Second, we explore using Gaussian filters on the best-performing model's input layer to regularize its input images and thereby improve its performance.

A. CONVOLUTIONAL NEURAL NETWORKS AND TRANSFER LEARNING

In deep learning, CNNs are commonly employed for machine vision due to their efficiency and speed [37]. A CNN consists of several stages, consisting of convolutional and pooling layers. Each convolutional layer contains a feature map connected to the previous layer through filters. This results in locally weighted sums that are then passed through a non-linearity function. The weights can be adjusted through training the model, enabling the CNN to detect features in an image. This training process requires a large number of images until the CNN is able to classify them reliably, and collecting a large image dataset can be an expensive task [38].

However, it is possible to use a CNN that has already been trained on a large dataset and then retrain the model with a much smaller sample of additional data. This process of transferring a model's weights is called transfer learning [39]. In transfer learning, the pre-trained model's weights are used, as the pre-trained model is already capable of recognizing features in the image. Next, the model output layer is adapted to fit the new classification task. In particular, the number of new classes is defined in a new classification head. This new output layer is then trained on the new dataset while the weights of the underlying base model remain unchanged. This process of training the classification head in the transfer learning process is called fine-tuning [38].

B. OUR NETWORK ARCHITECTURES

We created three CNN-based models based on commonly used CNN architectures, namely VGG16, ResNet50, and MobileNetV3Small, utilizing transfer learning to further increase the robustness of our models. CNNs based on transfer learning are considered to be more robust than novel CNNs since they are less sensitive to the training data [24].

Our first model architecture is based on Simonyan and Zissermann's VGG16 CNN architecture [21]. This architecture was designed for the ImageNet.com image classification competition and won the competition in 2014. We chose the VGG16 architecture as it can solve complex image classification problems [21], has shown high performance in the classification of RGB images [40], and is well-documented through the TensorFlow Python library [41]. Hence, VGG16 is an appropriate CNN architecture for our research.

The VGG16 architecture consists of five blocks of convolutional layers, wherein the first and second blocks each consist of two convolutional layers, while the remaining three blocks consist of three convolutional layers [21]. All thirteen convolutional layers use a Rectified Linear Unit (ReLu) [42] activation function [21]. The five blocks of convolutional layers are each followed by a max pooling 2D layer (with 2×2 kernel size), which decreases the resolution of the feature map and samples down the input for the next block [21], [43]. The max pooling operation was chosen in the model's architecture as it has shown to be superior for capturing invariances in image data compared to subsampling operations [43]. While the convolutional filters are small (3×3), the VGG16 architecture improves upon prior architectures by increasing the neural network depth through additional layers [21]. Additionally, the VGG16 architecture includes two fully connected layers with 4,096 units each and a 1,000-node softmax layer as classification head [21].

Our second model architecture is based on ResNet50, proposed by Kaiming et al. [44] from Microsoft Research in 2015. The ResNet architecture won the ImageNet classification competition in 2015. Like VGG16, the architecture has since been widely used for various image classification tasks.

Previous studies, for example, Karakaya et al. [26], indicated a strong performance of ResNet in the separation of fresh and rotten fruit.

The ResNet architecture generally consists of an initial convolutional layer of size 7×7, 64. This layer is followed first by a 3×3 max pool layer and second by varying amounts of four different types of residual blocks. Finally, these residual blocks are followed by a classification head consisting of an average pool layer and a 1000-neuron fully connected layer with softmax. The ResNet50 architecture builds upon the 34-layer ResNet architecture by replacing each of the 2-layer residual blocks with a 3-layer residual block, resulting in a net with 50 layers. Each 3-layer block consists of three convolutional layers (1×1, 3×3, 1×1). These layers vary in input and output dimensions and first decrease and then increase the dimensions. Additionally, the ResNet architecture introduces identity blocks, which are used as shortcuts in each residual block. As such, the identity blocks connect from the input of one 3-layer block to the input of the next 3-layer block. This is done directly whenever both blocks are of the same dimensions. When the last of the first three residual blocks is connected to the fourth residual block for increasing dimensions, a projection shortcut is used to match dimensions with the aid of a 1×1 convolutional layer.

Our third model architecture is based on the MobileNet, initially proposed by Howard et al. in 2017 [45]. Also built for the ImageNet competition, the MobileNetV3 architecture [46] combines the idea of residual blocks with a more efficient model architecture.

While there are two versions of MobileNetV3, their general architectural structure is the same, with the difference being the number of bottleneck layers and their respective input and kernel sizes. The architecture mostly uses h-swish non-linearity functions, with the exception of the first bottleneck layers, which use ReLu non-linearities. Each bottleneck block consists of three convolutional layers, which are similar to the residual blocks in the ResNet architecture. First, a 1×1 convolution layer block, then a 3×3 convolution layer block, followed by a 1×1 convolution layer block, where each block integrates batch normalization and non-linearity functions. Our third model particularly uses the MobileNetV3Small architecture, which consists of a 2-D convolutional layer followed by the bottleneck layers and another 2-D convolutional layer with a 1000-neuron classification head on top. The bottleneck layers in the MobileNetV3Small architecture consist of two bottleneck layers with a 3×3 kernel size, followed by eight bottleneck layers with a 5×5 kernel size [46].

For all three of our architectures, we used the weights of the ImageNet dataset as included in the Keras library (ImageNet.com).

However, we did not include the top of the base layer as we added our own binary classifier on top. We defined all three base models to be untrainable in the first stage of our training process, as we only wanted to train our custom classification heads [38]. These new classification heads each consist of a Global Average Pooling 2-D layer with 512 neurons. We then add a dropout layer with a factor of 0.2. The dropout layer randomly sets inputs to 0 during training, thereby increasing the robustness of the models by preventing overfitting [47]. Finally, we add a dense layer for prediction, which condenses down to a single neuron as we only want to return the most likely prediction. Our classification heads, therefore, have 513 trainable parameters in the dense layer.

An example of the resulting architecture of our models is depicted for our VGG16 model in figure 2. Our ResNet50 and MobileNetV3Small models follow this architectural approach analog to VGG16.

We used the Adam optimizer in the training of the models as it combines several benefits of other optimizers [48]. We set the learning rate of the Adam optimizer to 0.0001 for the first stage of training. In the first stage of training, we trained our models for up to 30 epochs. During this stage, we implemented a callback function to stop the training process when no improvement in accuracy was reached for more than four epochs. After the initial training process, we unfroze several top layers of the base models as depicted in table 2.

TABLE 2. Training and fine-tuning epochs.

Model	Layers unfrozen for fine-tuning
VGG16	15-19
ResNet50	151-175
MobileNetV3Small	151-229

These were then fine-tuned using the RMSProp optimizer with a lower learning rate of 0.00001 for a total of up to 40 training epochs. We again implemented a callback function to stop the training process when no improvements in validation accuracies were made over four consecutive epochs.

C. EVALUATION METHOD

For the purpose of evaluation, we used 10% of our dataset for validation in each epoch during training and 10% as unseen test data. We then predicted the labels of our test dataset with our model. Subsequently, we compared the results of this classification against the labels of the images in order to determine if the classification is correct. This allows us to determine the accuracy of the models as well as the type I and type II errors. We followed the well-established approach of plotting the results in a confusion matrix. Apart from accuracy, we also looked at other performance criteria such as the F1 score, balanced accuracy, the true positive rate, the positive predictive value, and Cohen's kappa.

D. GENERAL PREPROCESSING

In terms of preprocessing, we had to provide our models with the correct three-channel RGB input data format. Additionally, we resized all images to 250×250 pixels to decrease the use of computational resources during the training process. Subsequently, we randomly split our

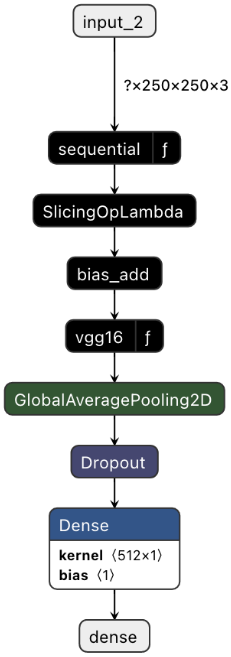

FIGURE 2. Model architecture.

dataset into 80% training, 10% validation, and 10% training images. Our validation images were used to validate training results during the training process, and the test images were exclusively used for the final evaluation of the models.

Furthermore, we also employ data augmentation to increase the number and variability of training images and thereby aim to reduce overfitting [49]. To augment our training dataset, we flipped and rotated the images in the training dataset in a random direction (an example of this augmentation is depicted in figure 3).

E. ADDING GAUSSIAN FILTERS TO PREPROCESSING

In addition to our general preprocessing, we also explored the impact of various Gaussian filter sizes on the performance of the best-performing architecture. Our study is, to the best of our knowledge, the first to apply a Gaussian filter in fruit classification.

The Gaussian filter is a low-pass filter that reduces high-frequency noise by smoothing the image while retaining edges [50]. To this extent, a convolution of the bell-shaped curve Gaussian function is applied to the input layer. This bell-shaped curve function is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where (x,y) are the coordinates of each pixel in the given image, and σ is the standard deviation of the Gaussian distribution that describes the width of the bell curve. The filter then calculated the value of each pixel (x,y) using a weighted average of the pixels in the kernel. The weighting of each pixel is determined by the Gauss function and its distance to the center pixel. As σ becomes larger, the Gaussian curve becomes wider, and pixels further from the center pixel are weighted more heavily. The kernel size describes how many pixels are included in the operation. This study used the OpenCV implementation [51] of the kernel to calculate σ as:

$$\sigma = 0.3 * ((k - 1) * 0.5 - 1) + 0.8 \tag{2}$$

where k is the kernel size. In this study, we explored the effect of various kernel sizes, i.e., 3×3, 5×5, 7×7, 8×8, 9×9, 11×11, 13×13, 15×15, 17×17, 19×19. The increasing kernel sizes result in increasingly stronger smoothing of the

FIGURE 3. Example of training image augmentation.

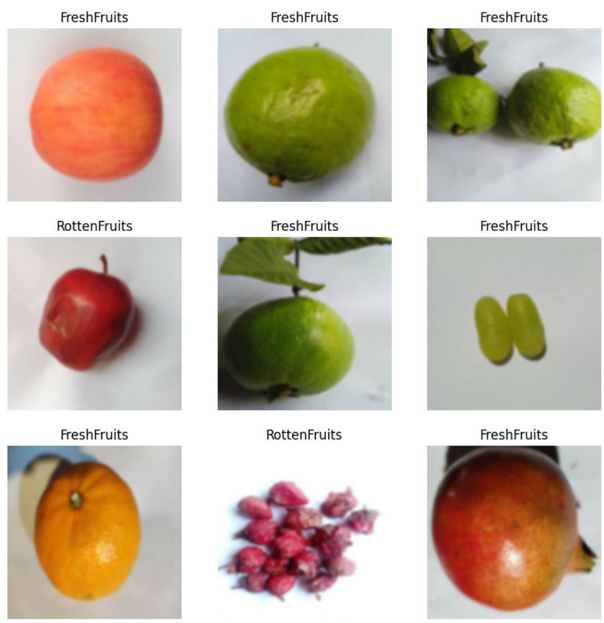

FIGURE 4. Example images with applied Gauss filter with kernel size 9 × 9.

input image. Figure 4 depicts example images with an applied Gaussian filter with a 9 × 9 kernel. We applied each filter to a fresh copy of our dataset and trained a new model based on our best-performing approach. We then employ the same evaluation method as previously elaborated to determine the performance.

F. DATASET

The objective of this study is to create three models that are both robust and high-performing in separating rotten and fresh fruits. Crucially, the models are trained to separate between fresh and rotten fruit, irrespective of the kind of fruit. To the authors' knowledge, this contribution represents the first attempt to utilize a two-class approach to distinguish rotten and fresh fruits using eight distinct kinds. While other authors, such as Karakaya et al. [26] and Yogesh et al. [27], have explored comparable approaches, our study is distinctive in its use of eight fruit types, as opposed to their use of three and five, respectively.

Furthermore, the dataset utilized in our research was obtained from a peer-reviewed source. The underlying assumption behind our approach is that the "rottenness" displays similar features across the different fruits. Since the fruits have different shapes, sizes, and colors, the training data is more diverse than a single kind of fruit, and diverse training data improves the robustness of a model [52]. In other applications, such as language processing, diverse training data has also led to more robust deep learning models [53].

Therefore, in order to enhance our models' robustness and address the limitations of previous research, they are trained on a large, diverse, and balanced dataset containing images of eight different fruits: grape, guava, jujube, pomegranate, strawberry, apple, banana, and orange [54]. The selected dataset of Sultana et al. [54] consists of 3,200 images of rotten and fresh fruit. It was published in Data in Brief in October of 2022 [54]. A unique aspect of this dataset is the inherent variance in the images. Sometimes, fruits overlap, creating a complex visual pattern. Additionally, the images were taken under varying light conditions, with differences in intensity, angle of incidence, and brightness, adding another layer of complexity to the dataset.

For each fruit, there are 200 original images of its rotten state and another 200 of its fresh state, all in RGB format. While our primary goal was to distinguish between rotten and fresh fruits rather than to identify the fruit type, we combined the images from different fruit categories. This results in a combined pool of 1,600 images of rotten fruits and another 1,600 of fresh ones. Examples of images from the dataset are depicted in figure 5.

It is noteworthy that our training and validation datasets remain balanced in terms of fruit type and freshness of the fruit, as we randomly split the data for each of the 16 classes separately before merging the datasets. With 80% of the images earmarked for training, 10% for validation, and 10% for testing, our final training dataset comprises 2,560 images. The validation and testing datasets each contain 320 images.

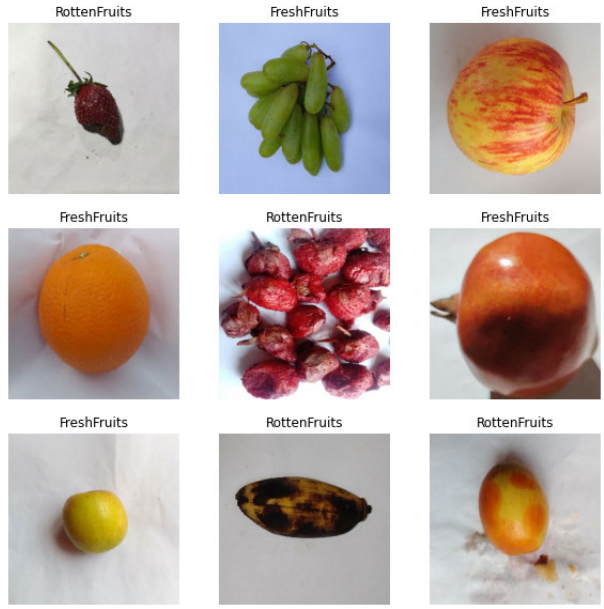

FIGURE 5. Example images from the dataset (without Gauss filter).

IV. RESULTS

We utilized the TensorFlow v2.11.0 framework with the included Keras Package [41], [49] to train our models. The training itself took place on the publicly available Google Collaboration tool. We used the standard hardware provided by Google, which consists of an NVIDIA Tesla K80 12 GB GPU and up to 16GB of RAM. The resulting classification accuracies over the whole training phase for the models VGG16, MobilNetV3Small, ResNet50 and ResNet50 with a Gauss filter (5 × 5 kernel), are depicted in figures 8-11 in the appendix of this paper.

To evaluate our models, we tested them on 320 previously unseen pictures, consisting of approximately 50% rotten and 50% fresh fruits. Illustrative example classifications of our VGG16 model are depicted in figure 6. The predictions and corresponding true values of all three models on our test dataset are shown in the confusion matrices in figures 12-14 in the appendix of this paper.

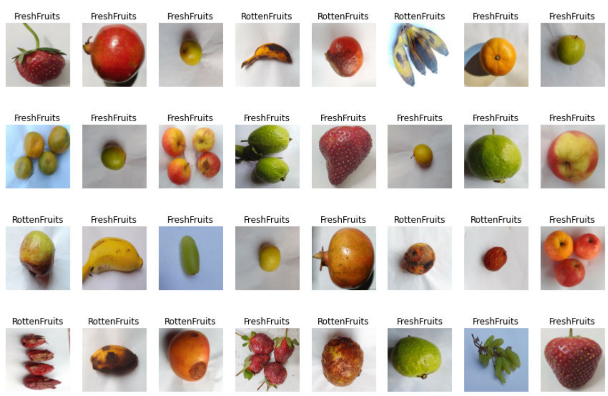

FIGURE 6. Examples of model predictions.

For the first model with the VGG16 architecture, out of the 320 fruits, 318 were classified correctly, and only two false predictions were made. Namely, two rotten fruit were wrongly classified to be fresh. However, the VGG16-based model correctly classified all of the fresh fruits. The second model with the ResNet50 architecture showed similar results, with 319 of 320 fruits correctly classified. The one false prediction was a rotten fruit, which was classified as fresh. The third model was designed with a MobileNetV3Small architecture. It correctly classified 309 of 320 images. Out of the eleven false predictions, nine were actually rotten fruits, incorrectly classified as fresh, while two fresh fruits were classified as rotten. To better evaluate the performance, several performance indicators were calculated for each of the models, which are presented in table 3. The indicators show that our models deliver a strong performance with a balanced accuracy rate of 99.38% for VGG16, 99.70% for ResNet50, and 96.71% for MobileNetV3Small.

TABLE 3. Performance indicators of initial models.

Performance Indicator	VGG16	ResNet50	MobileNetV3
Accuracy (%)	99.38	99.69	96.56
Balanced Accuracy (%)	99.38	99.70	96.71
True Positive Rate (%)	98.76	99.39	94.77
True Negative Rate (%)	100	100	98.65
Positive Predictive Value (%)	100	100	98.79
Prevalence (%)	50.31	51.56	53.75
Cohen's Kappa (%)	98.75	99.38	93.13
F1-Score (%)	99.38	99.70	96.74

As our ResNet50-based model achieved the highest performance, we utilized this architecture to evaluate the impact of the previously described Gauss filters on performance. To this extent, the models were tested with the same test images but with the corresponding Gauss filter applied to both the training and the test dataset. In order to ensure robust results, we evaluated each kernel size over ten random dataset splits. This results in the performance indicators depicted in table 4. The performance indicators show that when a Gauss filter kernel of size 3×3 or 5×5 is applied to the input images of our ResNet50-based model, it results in higher balanced accuracies of 99.66% compared to 99.50% without a Gauss filter applied to the input images. A kernel size of 7×7 leads to a balanced accuracy of 99.51%, which only very slightly outperforms the model with no filter applied to the input images. For kernel sizes above 7×7, the balanced performance drops below the model's with no filter as depicted in figure 7.

V. DISCUSSION

The models presented in this paper demonstrate a strong performance in classifying fruits as either rotten or fresh. This proves that the presented multi-fruit approach for identifying rotten fruits is feasible. By using eight different kinds of fruit, which are distinct in shape, size, and color, we utilized a diverse training and test dataset in order to build a robust system.

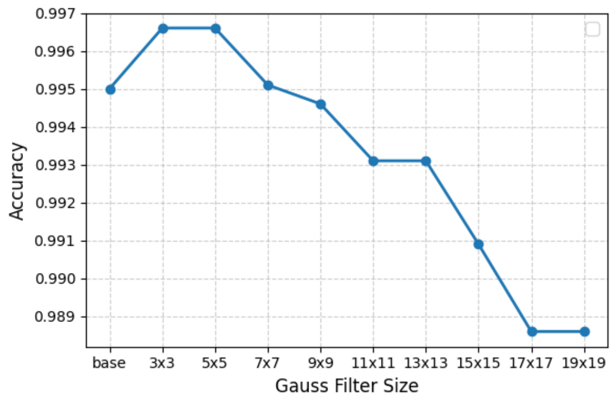

FIGURE 7. Model accuracies for Gauss filter sizes.

Compared to the related works presented in the research background, two out of the three presented models achieved a higher classification accuracy than related works, with a classification accuracy beyond 99%. That two CNN-based models exhibited a higher classification accuracy than SVM-based approaches, such as Bhargava and Bansal [28], supports the hypothesis that CNN-based models outperform SVM-based approaches. This hypothesis is further supported by the fact that the training dataset used to train the aforementioned SVM is substantially larger, with 13,851 RGB images compared to the 3,200 RGB images used in this paper.

The best performance is shown by the ResNet50-based model, which is also the most computationally intensive approach. VGG16 shows a marginally worse performance; for instance, the unbalanced classification accuracy is 0.31% lower compared to ResNet50. Meanwhile, VGG16 is less computationally intensive than ResNet but requires more resources than MobileNet. The latter exhibits the lowest performance out of the three. Among our observed models, we have noticed a direct correlation between higher computational resources and improved performance. For practitioners, this presents an insight into choosing the right architecture. Our findings suggest that there is a trade-off between performance and computational resources. Crucial for deciding the right architecture is the related cost to computational power and the importance of avoiding misclassifications in the individual use case.

We find that MobileNet would also be suitable for the separation of rotten and fresh fruit, given its classification accuracy of 96.56%. Given its relatively low computational demand, it is suitable for edge applications where processing power is typically scarce, such as autonomous field drones or sorting machines in rural areas. In these scenarios, SVM-based approaches could be considered as well since they are generally less computationally demanding than CNN-based approaches. In further research, MobileNet and SVM-based approaches could be compared to evaluate which method is the most suitable in such scenarios.

We also explored the usage of Gauss filters as an additional preprocessing step and found that our ResNet50 model seems to benefit from this approach for small kernel sizes.

TABLE 4. Performance indicators of ResNet50-based models with applied Gauss filter with increasing kernel sizes applied to the dataset (averaged over ten iterations).

Performance Indicator	ResNet50 (no filter)	3x3	5x5	7x7	9x9	11x11	13x13	15x15	17x17	19x19
Accuracy (%)	99.51	99.67	99.67	99.51	99.46	99.33	99.33	99.11	98.88	98.88
Balanced Acc (%)	99.50	**99.66**	**99.66**	99.51	99.46	99.31	99.31	99.09	98.86	98.86
True Positive Rate (%)	99.01	99.36	99.32	99.28	98.91	98.62	98.62	98.62	98.16	98.16
True Negative Rate (%)	100	99.96	100	99.74	100	100	100	99.57	99.57	99.57
Positive Predictive Value (%)	100	99.95	100	99.73	100	100	100	99.53	99.53	99.53
Prevalence (%)	49.49	49.17	49.17	49.29	49.29	48.44	48.44	48.44	48.44	48.44
Cohen's Kappa (%)	99.02	99.33	99.33	99.02	98.93	98.66	98.66	98.21	97.76	97.76
F1-Score (%)	99.50	99.66	99.66	99.50	99.45	99.30	99.30	99.07	98.84	98.84

For larger values, the image becomes too blurred for the model, and performance is heavily reduced. We thus suggest a Gauss filter with small kernel sizes as a method for preprocessing. Furthermore, future research could investigate the implementation of Gaussian filters in the layers of the architecture itself, as this may aid with regularization and thus lead to more robust models.

While the presented models perform the classification task with high accuracy, mistakes are still made. We presume that these errors occur because some of the fruits have irregularities in their skin, even when in fresh condition, while others do not exhibit such characteristics (see figure 6). As we do not label the different fruit types in the training and validation data, the models cannot differentiate these fruit-specific features and have to train on generalistic features of rottenness to generate the classification predictions.

However, we have intentionally decided on this training approach, as our assumption is that merging the fruit classes and limiting the classification into the two categories of fresh or rotten increases the robustness. As previously mentioned, we assume that the models have to recognize the classification across class-specific characteristics and thus develop a sharpened understanding of rottenness. The intention is that this may improve applicability across different fruit classes and improve real-world usability. However, this would need to be tested in future research.

This circumstance makes it difficult for the CNN model to classify them correctly. In the design of a model like ours, there is a trade-off between the type I and type II errors, meaning that the type I error can be decreased with a subsequent increase in a type II error and vice versa [55]. How this is handled in real-world applications is dependent on the related cost and impact of the two types of misclassification. The authors of this paper, however, assume that the type II error is more critical in most cases since the cost of a single fruit is typically very low, and the harvest becomes profitable through large volumes. Therefore, in most cases, the incorrect sorting-out of individual fruits is acceptable, considering the protection of large stocks from spreading rot. All of the three presented models show a higher type II than type I error, which should be evaluated before implementing them in a production setting. Depending on the related performance effects and subsequent trade-off, adaptions to decrease the type II error at a possible increase of the type I error can be made. Nevertheless, this consideration depends on different factors, which could involve aspects such as shipping time, food price, and the severity of the rottenness.

However, the proposed models distinguish rotten from fresh fruit with high classification accuracies ranging from 96.56% up to 99.66% (with the addition of a Gauss filter with a small kernel size of 3×3 or 5×5). Based on this, we depict a possible alternative to human labor. Especially considering that repetitive tasks such as sorting out fruits cause fatigue, which can drastically lower the attention of workers, negatively affecting detection and classification accuracy over time [56], [57].

In general, we suggest the application of our model in a scenario after harvesting and before storage or packaging since rot is particularly dangerous when many fruits are stationary and in close proximity, which facilitates quick spreading. Therefore, it could be useful to apply it, for instance, on a conveyor belt after harvesting and before the fruit is temporarily stored in the warehouse or packed in large quantities and transported further.

VI. CONCLUSION

We successfully built highly accurate binary rotten/fresh fruit classification models using a CNN approach, which has potentially high robustness due to a large and diverse dataset of eight distinct kinds of fruits. Therefore, we confirm that CNN architectures can be used to accurately identify rotten fruits for quality control. We employ transfer learning, which enables us to leverage the weights of the "ImageNet" dataset for each of our models.

This means we can train our models using a fraction of the data required for training a new model, capitalizing on the pre-trained models' feature detection capabilities. Through data augmentation, we were also able to greatly increase the amount of training data, leading to a more robust model. Notably, we augmented our training dataset only after splitting the dataset of Sultana et al. [54] into training, validation, and testing data so it wouldn't bias our evaluation or risk data leakage.

To the best of our knowledge, this work represents the first attempt to utilize a two-class approach to distinguish rotten and fresh fruits using eight distinct kinds. With a balanced accuracy of 99.70%, our best model outperforms previous studies using CNN and SVM approaches. The variety of eight distinct kinds of fruit also leads to robust classification models. Additionally, we suggest the use of a Gauss filter with a small kernel size to regularize the input images for the model, as this seems to improve model performance consistently.

Applying our model to real-world applications could also result in accuracy and efficiency gains as well as cost reductions when compared to manual quality control labor. Consequently, our model also has practical relevance, as these benefits could aid in decreasing the cost of quality control along the supply chain and ensuring high fruit quality. Identifying and removing rotten fruit from produce batches also prevents the emission of enzymes from rotten fruit, thereby increasing the shelf life of the entire batch.

A. LIMITATIONS

As with any research, our work has its limitations. First, while there is a wide range of different fruit images with various characteristics in the dataset of Sultana et al. [54], the pictures were ultimately taken in a lab environment [54]. As we use this dataset to train and evaluate our models, they lack external validation and need to be tested in a real-world application.

Generally, our models could perform worse when tested with images that include objects other than fruit, as our dataset only included the fruits in a controlled environment in front of a white background. This limitation applies to objects in the background of the fruit and dirt or leaves on the fruit itself. Therefore, achieving a comparable accuracy in real-world settings would likely require additional training. While the dataset contains images with various levels of brightness and light angles that produce shadows on the fruit, performance in real-world lighting conditions may vary.

A rotten fruit detection system using one of our models deployed in a real-world application would also need to capture high-quality images of the fruit. However, this could be difficult if the fruit is moving on a fast conveyor belt. Additionally, it must be considered that the fruit images used in our dataset are only 2-dimensional and display only the camera-facing side of the fruit. The system would, therefore, fail to identify fruit that is not visibly rotten in that respective viewing angle but might have a rotten spot on the other side.

Consequently, in a real-world application, the models would need to be paired with additional sensors and systems that enable a 360-degree view of each fruit, for example, by taking several pictures of each fruit that passes by on a turning conveyor belt. Additionally, the models are limited in that, while they can reliably classify rotten fruits, a different system would be necessary to remove the rotten fruit from its batch to be useful.

B. FURTHER RESEARCH

Future work could focus on fine-tuning the models on other rotten and fresh fruit datasets to evaluate the proposed models' robustness further. Additionally, different camera equipment could be utilized to evaluate the ability of

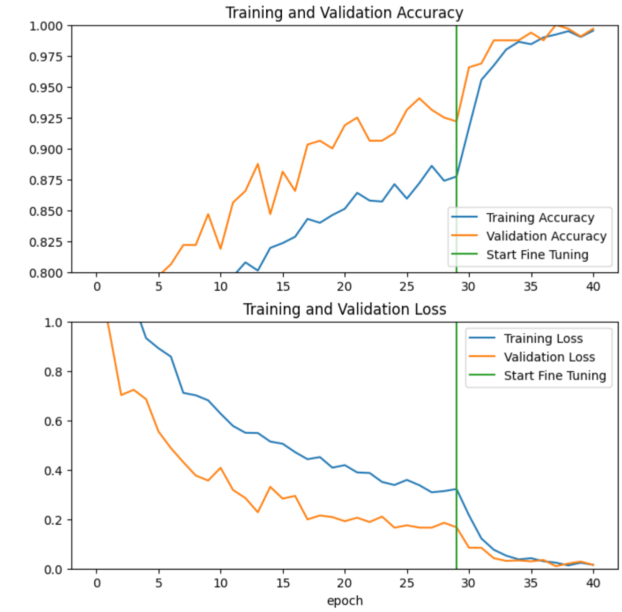

FIGURE 8. Training and validation accuracy graph for VGG16.

our approach to deal with more realistic conditions. New datasets should also contain more complex pictures with particularly challenging lighting situations, images of fruit with leaves, dirt, multiple kinds of fruit in a single image, and imperfect or noisy (non-white) backgrounds to represent real-world conditions accurately. Ideally, such a dataset could be collected from the actual implementation environment of the system to match the conditions as closely as possible. A real-world implementation could also include new, use-case-dependent, multi-class approaches that, for instance, not only identify if a fruit is rotten but also dirty, and thus enable the automatic detection of fruits that require cleaning before offering them to customers in supermarkets and thereby potentially improve the perceived quality of the fruit sold. Furthermore, future research could add more classes to the model to classify freshness and rottenness more precisely. Other conditions of the fruit, such as ripeness or infections with insects, could also be avenues for additional research.

An additional avenue for future research arises from addressing the type II error of our models, as seen in the confusion matrices in Figures 12-14 in the appendix of this paper. As stated above, this could be harmful in a real-life application and potentially cause damage to the yield if rotten fruits stay undetected in post-harvest processing stages. Consequently, future research could focus on preventing such errors or trading them.

Another promising direction for further research would be to adapt the current methodology to encompass the identification and subsequent classification of individual fruits within images that showcase multiple fruits. This would add the ability to differentiate rottenness by fruit instead of differentiating by image specifically. Such an extension would bolster the granularity and precision of detection and

FIGURE 9. Training and validation accuracy graph for ResNet50.

FIGURE 10. Training and validation accuracy graph for MobileNetV3Small.

pave the way for more intricate analyses in mixed fruit environments.

Finally, explainable AI represents another possible option for further research based on the present work. Explainable AI could determine what factors contributed to the predictions provided by the models and thus allow system users to interpret the results more deeply. In this context, example-based or similarity explanations provide more context and secondary information to help explain and emphasize the differences in the data and explain how the models act.

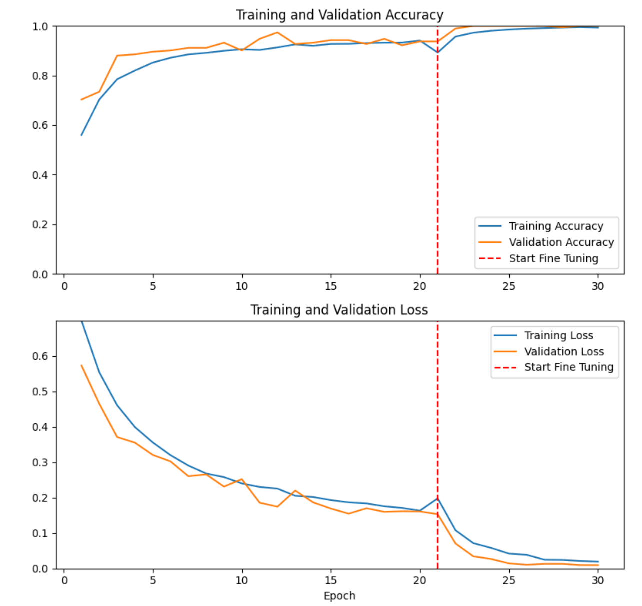

FIGURE 11. Training and validation accuracy graph for ResNet50 with Gauss filter with kernel size 5 × 5 applied to input images.

		Predicted	
		fresh	rotten
Actual	fresh	159	0
	rotten	2	159

FIGURE 12. Confusion matrix VGG16 model.

		Predicted	
		fresh	rotten
Actual	fresh	155	0
	rotten	1	164

FIGURE 13. Confusion matrix ResNet50 model.

This information could then be used to verify that the models behave as expected, recognize bias in the models, and

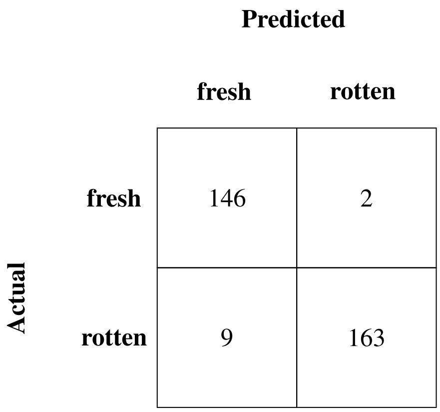

FIGURE 14. Confusion matrix MobilNetV3Small model.

generate ideas for improving them. For the specific case of a system that detects rotten fruits, it could be paired with a user-friendly interface that enables users to identify why the system has detected certain fruits as rotten. This is especially of interest, as those involved in food production are typically no experts in advanced IT technology. Therefore, there is a need for easy-to-understand insights into how these systems work [32]. For example, the system might falsely identify fruit as rotten if harvesting is done in muddy conditions, which leaves brown spots on the yield. Thus, in this example, specific measures, such as an additional washing process in advance of the detection system, could be implemented to ensure the correct functioning of the system.

APPENDIX

See Figures 8–14.

REFERENCES

[1] M. van Dijk, T. Morley, M. L. Rau, and Y. Saghai, "A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050," *Nature Food*, vol. 2, no. 7, pp. 494–501, Jul. 2021.

[2] P. Zhu, J. Burney, J. Chang, Z. Jin, N. D. Mueller, Q. Xin, J. Xu, L. Yu, D. Makowski, and P. Ciais, "Warming reduces global agricultural production by decreasing cropping frequency and yields," *Nature Climate Change*, vol. 12, no. 11, pp. 1016–1023, Nov. 2022.

[3] B. Zheng, K. Chenu, M. Fernanda Dreccer, and S. C. Chapman, "Breeding for the future: What are the potential impacts of future frost and heat events on sowing and flowering time requirements for Australian bread wheat (*Triticum aestivium*) varieties?" *Global Change Biol.*, vol. 18, no. 9, pp. 2899–2914, Sep. 2012.

[4] Y. Qin, J. T. Abatzoglou, S. Siebert, L. S. Huning, A. AghaKouchak, J. S. Mankin, C. Hong, D. Tong, S. J. Davis, and N. D. Mueller, "Agricultural risks from changing snowmelt," *Nature Climate Change*, vol. 10, no. 5, pp. 459–465, May 2020.

[5] C. Hong, J. A. Burney, J. Pongratz, J. E. M. S. Nabel, N. D. Mueller, R. B. Jackson, and S. J. Davis, "Global and regional drivers of land-use emissions in 1961–2017," *Nature*, vol. 589, no. 7843, pp. 554–561, Jan. 2021.

[6] W. F. Laurance, J. Sayer, and K. G. Cassman, "Agricultural expansion and its impacts on tropical nature," *Trends Ecol. Evol.*, vol. 29, no. 2, pp. 107–116, Feb. 2014.

[7] M. C. Hunter, R. G. Smith, M. E. Schipanski, L. W. Atwood, and D. A. Mortensen, "Agriculture in 2050: Recalibrating targets for sustainable intensification," *BioScience*, vol. 67, no. 4, pp. 386–391, Apr. 2017.

[8] J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, and A. Meybeck, *Global Food Losses and Food Waste—Extent, Causes and Prevention*. Rome, Italy: Food and Agriculture Organization of the United Nations, 2011.

[9] Food and Agriculture Organization of the United Nations. (2023). *FAOSTAT Online Database: Crops Livestock Products*. [Online]. Available: www.fao.org/faostat/en/#data/QCL

[10] D. Singh and R. R. Sharma, "Postharvest diseases of fruits and vegetables and their management," in *Postharvest Disinfection of Fruits and Vegetables*. Academic Press, 2018, pp. 1–52.

[11] J. Cao, C. Wang, S. Xu, Y. Chen, Y. Wang, X. Li, and C. Sun, "The effects of transportation temperature on the decay rate and quality of postharvest Ponkan (*Citrus reticulata* Blanco) fruit in different storage periods," *Scientia Horticulturae*, vol. 247, pp. 42–48, Mar. 2019.

[12] A. Elik, D. K. Yanik, Y. Istanbullu, and N. Aysar, "Strategies to reduce post-harvest losses for fruits and vegetables," *Int. J. Sci. Technol. Res.*, vol. 5, no. 3, pp. 29–39, 2019.

[13] P. Vithu and J. A. Moses, "Machine vision system for food grain quality evaluation: A review," *Trends Food Sci. Technol.*, vol. 56, pp. 13–20, Oct. 2016.

[14] J. F. S. Gomes and F. R. Leta, "Applications of computer vision techniques in the agriculture and food industry: A review," *Eur. Food Res. Technol.*, vol. 235, no. 6, pp. 989–1000, Dec. 2012.

[15] I. Charania and X. Li, "Smart farming: Agriculture's shift from a labor intensive to technology native industry," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100142.

[16] A. Chougule, A. Pawar, R. Kamble, J. Mujawar, and A. Bhide, "Recognizing fresh and rotten fruits using deep learning techniques," in *Data Engineering and Intelligent Computing* (Advances in Intelligent Systems and Computing), vol. 1407, V. Bhateja, S. C. Satapathy, C. M. Travieso-González, and V. N. M. Aradhya, Eds., Singapore: Springer, 2021, pp. 205–212.

[17] D. D. Bochtis, C. G. C. Sørensen, and P. Busato, "Advances in agricultural machinery management: A review," *Biosyst. Eng.*, vol. 126, pp. 69–81, Oct. 2014.

[18] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[19] M. S. Uddin and J. C. Bansal, "Introduction to computer vision and machine learning applications in agriculture," in *Algorithms for Intelligent Systems*. Singapore: Springer, 2021, pp. 1–8.

[20] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, "Application of deep learning in food: A review," *Comprehensive Rev. Food Sci. Food Saf.*, vol. 18, no. 6, pp. 1793–1811, Nov. 2019.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015, pp. 1–14.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[23] K. Hameed, D. Chai, and A. Rassau, "A comprehensive review of fruit and vegetable classification techniques," *Image Vis. Comput.*, vol. 80, pp. 24–44, Dec. 2018.

[24] Y. Yuan, L. Chen, H. Wu, and L. Li, "Advanced agricultural disease image recognition technologies: A review," *Inf. Process. Agricult.*, vol. 9, no. 1, pp. 48–59, Mar. 2022.

[25] L. M. Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," in *Proc. 7th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2017, pp. 242–246.

[26] D. Karakaya, O. Ulucan, and M. Turkan, "A comparative analysis on fruit freshness classification," in *Proc. Innov. Intell. Syst. Appl. Conf. (ASYU)*, Oct. 2019, pp. 1–4.

[27] Yogesh, A. K. Dubey, R. Ratan, and A. Rocha, "Computer vision based analysis and detection of defects in fruits causes due to nutrients deficiency," *Cluster Comput.*, vol. 23, no. 3, pp. 1817–1826, Sep. 2020.

[28] A. Bhargava and A. Bansal, "Classification and grading of multiple varieties of apple fruit," *Food Anal. Methods*, vol. 14, no. 7, pp. 1359–1368, Jul. 2021.

[29] S. A. Khoje, S. Bodhe, and A. Adsul, "Automated skin defect identification system for fruit grading based on discrete curvelet transform," *Int. J. Eng. Technol. (IJET)*, vol. 5, no. 4, pp. 3251–3256, 2013.

[30] Z. Wang, M. Hu, and G. Zhai, "Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data," *Sensors*, vol. 18, no. 4, p. 1126, Apr. 2018.

[31] A. A. Chandini and U. Maheswari B., "Improved quality detection technique for fruits using GLCM and MultiClass SVM," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 150–155.

[32] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, "Machine learning in agriculture: A comprehensive updated review," *Sensors*, vol. 21, no. 11, p. 3758, May 2021.

[33] X. Yu, H. Lu, and D. Wu, "Development of deep learning method for predicting firmness and soluble solid content of postharvest korla fragrant pear using vis/NIR hyperspectral reflectance imaging," *Postharvest Biol. Technol.*, vol. 141, pp. 39–49, Jul. 2018.

[34] Z. Liu, Y. He, H. Cen, and R. Lu, "Deep feature representation with stacked sparse auto-encoder and convolutional neural network for hyperspectral imaging-based detection of cucumber defects," *Trans. ASABE*, vol. 61, no. 2, pp. 425–436, 2018.

[35] L. Kabbai, M. Abdellaoui, and A. Douik, "Hybrid local and global descriptor enhanced with colour information," *IET Image Process.*, vol. 11, no. 2, pp. 109–117, Feb. 2017.

[36] M. S. Mayernik, S. Callaghan, R. Leigh, J. Tedds, and S. Worley, "Peer review of datasets: When, why, and how," *Bull. Amer. Meteorological Soc.*, vol. 96, no. 2, pp. 191–201, Feb. 2015.

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[38] C. Iorga and V.-E. Neagoe, "A deep CNN approach with transfer learning for image recognition," in *Proc. 11th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2019, pp. 1–6.

[39] K. R. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, May 2016.

[40] A. Ahmad, D. Saraswat, V. Aggarwal, A. Etienne, and B. Hancock, "Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems," *Comput. Electron. Agricult.*, vol. 184, May 2021, Art. no. 106081.

[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, and C. Citro. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[43] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks—ICANN* (Lecture Notes in Computer Science), vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds., Berlin, Germany: Springer, 2010, pp. 92–101.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[46] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[47] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015, pp. 1–15.

[49] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[50] G. Deng and L. W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," in *Proc. IEEE Conf. Rec. Nucl. Sci. Symp. Med. Imag. Conf.*, vol. 3, Oct. 1993, pp. 1615–1619.

[51] OpenCV. *GaussianBlur*. Accessed: Feb. 17, 2025. [Online]. Available: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html#gac05a120c1ae92a6060dd0db190a61afa

[52] Z. Gong, P. Zhong, and W. Hu, "Diversity in machine learning," *IEEE Access*, vol. 7, pp. 64323–64350, 2019.

[53] Y. Yu, S. Khadivi, and J. Xu, "Can data diversity enhance learning generalization?" in *Proc. 29th Int. Conf. Comput. Linguistics*, Oct. 2022, pp. 4933–4945.

[54] N. Sultana, M. Jahan, and M. S. Uddin, "An extensive dataset for successful recognition of fresh and rotten fruits," *Data Brief*, vol. 44, Oct. 2022, Art. no. 108552.

[55] K. Kim, H. Hong, G. Nam, and K. Park, "A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor," *Sensors*, vol. 17, no. 7, p. 1534, Jun. 2017.

[56] D. Kahneman, *Attention and Effor* (Prentice-Hall series in experimental psychology). Upper Saddle River, NJ, USA: Prentice-Hall, 1973.

[57] R. Langner and S. B. Eickhoff, "Sustaining attention to simple tasks: A meta-analytic review of the neural mechanisms of vigilant attention," *Psychol. Bull.*, vol. 139, no. 4, pp. 870–900, 2013.

LEOPOLD FISCHER-BRANDIES (Member, IEEE) received the B.Sc. degree in business administration and the M.Sc. degree in digitalization and entrepreneurship from the University of Bayreuth, Germany. He is currently pursuing the Ph.D. degree the Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg. His current research interests include deep learning, multi-agent systems, natural language processing, and detecting deepfakes.

LUCAS MÜLLER received the B.A. degree in philosophy and economics from the University of Bayreuth, Germany. He is currently pursuing the M.Sc. degree in digitalization and entrepreneurship with the University of Bayreuth. He is also engaged in research at the Chair of Hybrid Intelligence, Helmut-Schmidt-University, Germany. His current research interests include text mining, sentiment analysis, and fairness in AI.

JUSTUS JOHANNES RIEGGER received the B.A. degree in business management and psychology from Hochschule Furtwangen University, Germany, and the M.Sc. degree in digitalization and entrepreneurship from the University of Bayreuth, Germany, where he focused on the intersection of technology, innovation, and business strategy. His academic and professional interests lie in the fields of sustainability, explainable AI, and hybrid AI, with a particular emphasis on their practical applications in business and industry.

RICARDO BUETTNER (Senior Member, IEEE) received the Dipl.-Inf. degree in computer science and the Dipl.-Wirtsch.-Ing. degree in industrial engineering and management from the Technical University of Ilmenau, Germany, the Dipl.-Kfm. degree in business administration from the University of Hagen, Germany, the Ph.D. degree in information systems from the University of Hohenheim, Germany, and the Habilitation (venia legendi) degree in information systems from the University of Trier, Germany. He is currently a Chaired Professor of the Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Germany. He has published more than 160 peer-reviewed articles, including articles in *Electronic Markets*, *AIS Transactions on Human–Computer Interaction*, *Personality and Individual Differences*, *European Journal of Psychological Assessment*, *PLOS One*, and IEEE Access. He has received 18 international the best paper, the best reviewer, and service awards and award nominations, including the best paper awards by *AIS Transactions on Human–Computer Interaction*, *Electronic Markets* journal, and *HICSS*, for his work.