



## OPEN Intelligent pear variety classification models based on Bayesian optimization for deep learning and its interpretability analysis

Tao Lu<sup>1,2,3,4</sup>, Fanqianhui Yu<sup>5,6,7</sup>✉, Yanting Yu<sup>4,8</sup> & Linlin Zhang<sup>9</sup>

Accurate classification of pear varieties is crucial for enhancing agricultural efficiency and ensuring consumer satisfaction. In this study, Bayesian optimized (BO) deep learning is utilized to identify and classify nine types of pears from 43,200 images. On two challenging datasets with different intensities of added Gaussian white noise, Bayesian optimization automatically searched for the optimal hyperparameters and identified two optimal models, whose classification performance was objectively evaluated. The results indicate that dataset configuration significantly impacts classification outcomes. The optimal model A achieved an accuracy of 97.29% on dataset A (training-to-testing ratio = 21:10), while the optimal model B achieved an accuracy of 90.39% on dataset B (training-to-testing ratio = 1:10). This study also explored the impact of different proportions of validation sets within the training set on the performance of the optimal models. Additionally, on the original Fruit360 dataset, the accuracy of the BO optimal model reached 100% (training-to-testing ratio = 12:5). Furthermore, feature visualization, strongest activations, and local interpretable model-agnostic explanations (LIME) techniques were used to demonstrate the optimal models' understanding of pear images with noise and to reveal how and why the models make classification decisions. In summary, this study's dataset configuration is closer to real agricultural applications, and BO deep learning addresses the challenge of manually finding optimal hyperparameters for CNNs in agricultural applications, the interpretability methods enhance the transparency and reliability of CNN-based models. These laying the foundation for the widespread application of deep learning methods in agriculture.

**Keywords** Pear varieties, Deep learning, Bayesian optimisation, Visualization methods, Model interpretability

Pears are the second most consumed pome fruit worldwide<sup>1</sup>. Today, the area devoted to world pear production is approximately 1.5 million ha, yielding 25 million tons pears annually<sup>2</sup>. Pears have a pleasant sweet taste, are a source of fiber and vitamin C, and have low sodium content, which make it a healthy snack if eaten fresh<sup>2</sup>. In addition, Pears have many beneficial properties for human health, such as antitussive, anti-inflammatory, antihyperglycemic, and diuretic<sup>3</sup>. Therefore, pears are very popular in daily life, with consumption by the adult population ranging from 23 to 108 g/day<sup>4</sup>. As an ancient fruit in temperate regions<sup>5</sup>, there are many groups of pears, such as the white pear, sand pear, and Ussurian pear. Typically, several varieties of fruits are grown simultaneously in an orchard, so it is easy to mix different varieties of fruits with similar appearance during

<sup>1</sup>School of Mechanical and Automotive Engineering, Qingdao University of Technology, Qingdao 266520, China.

<sup>2</sup>Key Laboratory of Industrial Fluid Energy Conservation and Pollution Control (Qingdao University of Technology), Ministry of Education, Qingdao 266520, China. <sup>3</sup>Institute of Oceanology, Chinese Academy of Sciences (IOCAS), Qingdao 266000, China. <sup>4</sup>Institute of Oceanographic Instrumentation, Shandong Academy of Sciences/Qilu University of Technology, Qingdao 266100, China. <sup>5</sup>Haide College, Ocean University of China, Qingdao 266100, China. <sup>6</sup>College of Food Science and Engineering, Ocean University of China, Qingdao 266003, China. <sup>7</sup>Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China. <sup>8</sup>Laoshan Laboratory, Qingdao 266000, China. <sup>9</sup>Qingdao MingDe Environmental Protection Instrument Co., Ltd., Qingdao 266000, China.

✉email: fanqianhuiyu@outlook.com

harvesting and marketing process<sup>6,7</sup>. Although different varieties of fruits are similar in external appearance, they have different intrinsic qualities in terms of taste or nutritional value<sup>8,9</sup>. For example, There is a great diversity of pear varieties in China due to its widespread consumption. Each kind is characterized by different appearances and contents of phenolic compounds, nutritional ingredients, antioxidant and anti-inflammatory activities and some other properties<sup>3</sup>. Which is why people spend a lot of time sorting, packing and labeling fruits before selling them<sup>8,9</sup>. In addition, the identification and classification of fruits is a necessary task as it smart agricultural applications<sup>10</sup>, such as mechanized automatic picking and fruit harvest assessment in large orchards, intensive processing of fruits in factories and integrated fruit weighing in supermarkets to automatically complete fruit price calculations, etc. In the traditional agricultural planting and harvesting process, fruit identification mainly relies on manual labor, which is highly subjective, slow and unable to meet large orchard applications<sup>11</sup>. While commonly used physicochemical analysis methods based on fruits components testing are time-consuming, expensive to run, and complex in samples preparation<sup>12,13</sup>. Subsequently, with industrial development and technological advances, a number of rapid and non-destructive techniques for differentiating fruits varieties have emerged, such as electronic nose, visible and near-infrared spectroscopy, and image processing-based methods<sup>14,15</sup>. Especially the application of machine vision and deep learning will improve the efficiency of all aspects of agricultural.

Deep learning has been successfully applied as a non-destructive technique for automatic identification, classification and detection of fruits and vegetables with the advantages of fast, convenience, low cost, and high accuracy<sup>16</sup>. In particular, Convolutional Neural Networks (CNNs), a deep learning-based framework with strong capabilities in automatic feature learning of images, have achieved impressive results in various food and agricultural challenges<sup>17,18</sup>. Recently, CNNs have been used for fruit recognition tasks, but mainly for quality assessment and bruise detection<sup>11</sup>. For instance, Agus Pratondo<sup>19</sup> used the classifier built by Inception-v3 to classify three types of pears and compared it with several traditional machine learning algorithms. The results showed that the classifier built using Inception-v3 had the best performance, with an accuracy of 94.00%. Ismail et al.<sup>20</sup> utilized the EfficientNet methodology to develop a system, which was trained and evaluated on two fruit datasets. The average accuracy for the two testing datasets was 96.7% and 93.8%, respectively. Rojas-Aranda, J. L., et al.<sup>21</sup> presented a three-fruit system based on CNN architecture and MobileNetV2. This method achieved 95% accuracy without using plastic bags to wrap the fruits. Gill, Harmandeep Singh, et al.<sup>22</sup> employed CNN, Recurrent Neural Networks, and Long Short-Term Memory deep learning methods to extract optimal image features, and to select features after extraction, and finally, use extracted image features to classify the fruits. José Naranjo-Torres<sup>23</sup> in his review found that CNNs are highly efficient for addressing critical tasks on fruit image processing within the agro-food industry. However, CNN-based approaches should still face important challenges in order to apply them in real-world scenarios. For example, search of CNN parameters, the number of layers and filters when proposing a CNN architecture for a specific problem, as well as determining the parameters and hyperparameters of the model, remains a relevant problem commonly solve by trial-and-error tuning until getting the best settings, which is very time-consuming for deep learning models. Additionally, images captured during agricultural production and processing are often noisy, and the “black-boxes” nature of deep learning results in a lack of model interpretability, these factors further impact the widespread application of CNNs in the agriculture.

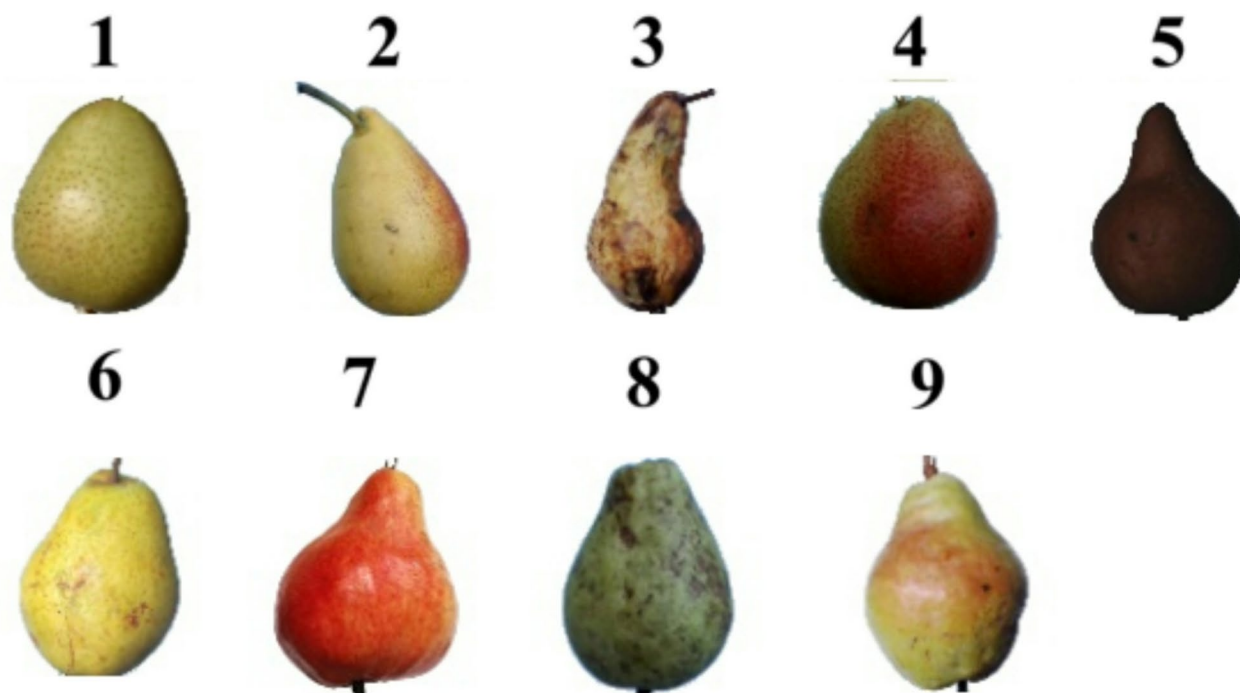
Accordingly, the contributions of this study are as follows.

1. We used BO deep learning to classify 9 categories of pears. Different hyperparameters and corresponding model performances were evaluated and compared, and two of the optimal models were further analyzed.
2. We set up two datasets and different validation set proportions to study the performance impact of data configuration on BO deep learning. Specifically, one is a common dataset configuration, i.e. more training data-less testing data, and the other is designed to approach the reality that the testing set is infinite, i.e. less training data-more testing data<sup>11</sup>. In addition, we added different degrees of Gaussian white noise to the data to be closer to actual agricultural production applications.
3. We used three visualization methods (feature visualization, strongest activations and LIME techniques) to reveal how two optimal models make classification decisions as “black boxes”.

## Materials and methods

### Fruits-360 dataset

“Fruits-360” (<https://www.kaggle.com/moltean/fruits>, Version: 2020.05.18.0, accessed on 10 October 2022) is a publicly available benchmark dataset<sup>8,24</sup>, which has been employed by several studies to evaluate their proposed models. For example, Siddiqi<sup>25</sup> used this dataset to classify different categories of fruits and illustrated that the Fruits-360 dataset is larger compared to other fruit datasets. Kodors et al.<sup>26</sup> used this dataset to classify apples and pears in order to compare the performance of different deep learning architectures. Choudhary, K., et al.<sup>27</sup> developed a fruit recognition approach using this dataset, i.e. the CNN-based ResNet-50 method was employed for extracting features and fruit identification, and has been determined to be 99% accurate. Rahman M M et al.<sup>28</sup> in the literature review section, enumerated and compared ten pieces of literature on fruit recognition and classification, among which four utilized the Fruit-360 dataset. Based on this, an enhanced version of “Fruits-360” dataset was employed in this study to objectively evaluate and demonstrate the performance of BO deep learning models and to facilitate researchers to reproduce our work. Fruit-360 as a comprehensive dataset featuring 131 different categories of fruits and vegetables, totaling 90,483 images. Among them, pear includes 9 types, namely Pear, Pear 2, Pear Abate, Pear Forelle, Pear Kaiser, Pear Monster, Pear Red, Pear Stone, and Pear Williams. Each image (100\*100 pixels) is of a single pear on a white background, as shown in Fig. 1.



**Fig. 1.** Nine classes of pears: (1) Pear, (2) Pear 2, (3) Pear Abate, (4) Pear Forelle, (5) Pear Kaiser, (6) Pear Monster, (7) Pear Red, (8) Pear Stone and (9) Pear Williams.

Class	Dataset A		Dataset B	
	Traing Set	Testing Set	Traing Set	Testing Set
Pear	3600	1200	1200	3600
Pear 2	3600	1200	1200	3600
Pear Abate	3600	1200	1200	3600
Pear Forelle	3600	1200	1200	3600
Pear Kaiser	3600	1200	1200	3600
Pear Monster	3600	1200	1200	3600
Pear Red	3600	1200	1200	3600
Pear Stone	3600	1200	1200	3600
Pear Williams	3600	1200	1200	3600
Total	32,400	10,800	10,800	32,400

**Table 1.** The number of Pear images per class for training and testing in the two datasets.

### Training and testing datasets set-up

Two datasets are constructed using images of 9 categories of pears in Fruits-360, as shown in Table 1. Dataset A is configured based on the original training and test sets provided by Fruits-360, and Dataset B is an inverted version of Dataset A. That is, the test set of Dataset B corresponds to the training set of Dataset A, and the training set of Dataset B corresponds to the test set of dataset A. In addition, in the original “Fruits-360” data set, the fifth category (Pear Kaiser) has the least training and testing data, 300 and 102 images respectively, while the eighth respectively (Pear Stone) has the most, 711 and 237 images respectively. In order to avoid data imbalance, the image of each category is extracted from the original data set, i.e. 300 and 100 images were randomly selected from the training and test set of each category respectively, and data augmentation was then performed on each image to construct the two datasets in this study.

Data augmentation is a commonly utilized technique that enhance the training effect of deep learning<sup>29</sup>. This approach mitigates the issue of overfitting in deep networks by effectively expanding the dataset size, particularly when training set is limited. Common data augmentation strategies encompass techniques such as geometric rotation, adversarial training, and generative adversarial networks (GANs), etc<sup>30</sup>. Despite their utility, these approaches exhibit limitations, for instance, geometric rotation may not effectively resolve the issue of low accuracy in CNN when identifying images with noise, while GANs are characterized by their complexity and the difficulty associated with training them<sup>31</sup>. Additive Gaussian white noise is a fundamental noise model used

in information theory to mimic the effect of many random processes that occur in nature environments<sup>32</sup>. The integration of Gaussian white noise into images provides a straightforward method for dataset augmentation. Consequently, to enlarge the dataset size and simulate varying qualities of images encountered in real-world scenarios, data augmentation is performed by injecting Gaussian white noise with a mean ( $M$ ) ranging from 0 to 1 in increments of 0.1 and a fixed variance of 0.01. Pear images with the addition of different levels of gaussian white noise as shown in Fig. 2.

Compared to other studies<sup>33–35</sup>, which generally employed only one dataset (more training data – less testing data, it similar to the data configuration in the original “Fruit-360”) to evaluate the performance of models, we set up these two datasets in order to find out the effect of different dataset configuration on classification results. This is because it is often difficult to obtain large training sets in practice, and even when large training sets are available, manually labeling them is a time-consuming and laborious work<sup>36</sup>. In addition, a larger training set means more computer resource consumption and longer training time for the same model. Therefore, these dataset configuration, especially dataset B, is closer to real applications and helps reflect the real performance of the models.

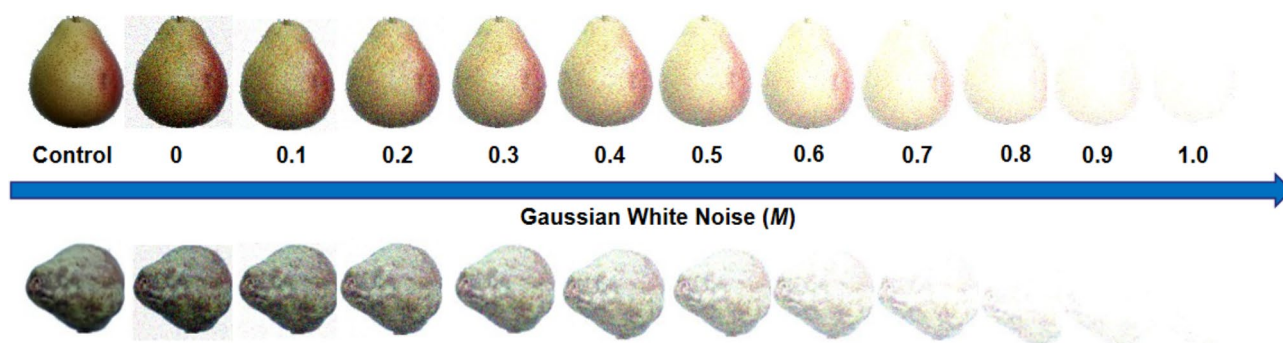
### Deep learning using bayesian optimization

In the implementation of deep learning, the determination of appropriate network depth and hyperparameters typically relies on the practitioner’s expertise and experience, these parameters need to be continuously adjusted to train the model and further to determine the optimal parameters and corresponding optimal models. This process often requires repetition, particularly when the training dataset is modified or the network requires periodic updates, resulting in considerable time and computing expenditure. BO uses an objective function to train the model, which makes it suitable for handling expensive evaluations in the deep learning training process. Because of its powerful hyperparameter adjustment strategy and its high efficiency, which applied in more and more research of deep learning. In additional, BO reduces the manual trial and error often required in deep learning applications by dynamically balancing exploration and leveraging existing information in the hyperparameter space, thereby speeding up the development process and making it possible to discover better configurations than manual tuning. It is often more efficient at finding optimal hyperparameter combinations than traditional methods such as random search.

#### Choose hyperparameters to optimize

In this study, BO is employed to achieve optimal hyperparameters selection. The hyperparameters include network section depth (NSD), initial learning rate, stochastic gradient descent (SGD) momentum, and L2 regularization strength, these hyperparameters can significantly affect the training process and performance of the model. The determination of hyperparameters search range values based on experience and expertise, literature, or a few simple attempts. The specific information is as follows.

- (1) NSD. The architecture of the network is divided into three sections, each containing an identical number of convolutional layers, such that the total number of convolutional layers equals three times the NSD. In order to ensure the number of parameters and the required amount of computation are roughly the same for different NSDs in each iteration, the objective function takes the number of convolutional filters in each layer proportional to  $1 / \sqrt{\text{NSD}}$  and this parameter range values is set to from 1 to 3.
- (2) Initial learning rate determines the size of the steps taken during the optimization process. A good learning rate is crucial for efficient convergence during training. In additional, the optimal initial learning rate can vary depending on the dataset and network architecture. This parameter range value is set to from  $1e-2$  to 1.
- (3) SGD Momentum introduces inertia to the parameter updates, helping the optimization algorithm to navigate through the loss landscape more efficiently. By incorporating information from previous updates, momentum can smooth the trajectory of parameter updates, leading to faster convergence and fosters better generalization. This parameter range value is set to from 0.8 to 0.98.



**Fig. 2.** Pear images with the addition of different levels of gaussian white noise.

- (4) L2 regularization strength. L2 regularization are used to mitigate overfitting by penalizing large weights in the network. The regularization strength controls the impact of this penalty on the loss function and appropriate value is important to balancing between reducing overfitting and maintaining model performance. This parameter range value is set to from 1e-10 to 1e-2.

#### *Perform BO and objective function*

The objective function for the Bayesian optimizer is established to minimize the classification error on the validation set while tuning hyperparameters for training CNNs. It leveraging past evaluations to guide subsequent iterations toward the goal of efficiently exploring the hyperparameter space. The objective function takes hyperparameters as input and trains a CNN model using these hyperparameters on the training data. The model's performance is then assessed by evaluating its classification error on the validation set. This error serves as the optimization criterion, guiding the BO process towards hyperparameter configurations that yield lower validation set error.

#### *Validation set configuration and optimal model selection*

After reaching the set maximum number of iterations, the hyperparameters yielding the lowest validation set error are selected as the optimal configuration. A final optimal model is trained using these optimal hyperparameters on the training dataset. The performance of optimal model is then evaluated on the independent test set, providing an unbiased assessment of the model's performance on unseen data. Among them, we also considered the impact of different configurations of validation sets on the performance of the optimal models.

### **Image processing**

The original size of the images in the “Fruit-360” dataset and the noise-added images both are 100\*100 pixels. The method in this study serves as a model trained from scratch, and the size of the input image does not have to be limited to the size requirements of the pretrained networks. In order to reduce the model's reliance on high-performance computer for storage, training, and inference, and further enhance the practical applicability of the models, all images are resized to 32\*32 pixels. The adjusted image is greatly reduced size whether compared with the original image or the size requirements of the pretrained network for the input image. For instance, in previous research, all images were resized to fit the input size requirements of each pre-trained network. Specifically, 227\*227 pixels for AlexNet, and 224\*224 pixels for VGG-19, ResNet-18, ResNet-50 and ResNet-101<sup>11</sup>.

### **Metrics for performance evaluation of optimal model**

Given that BO selects the optimal model based on the minimum error observed on the validation set, it is possible that the optimal model overfits on the validation set. Therefore, an independent testing set is used to test the performance of the optimal model, and it is visually displayed through the confusion matrix, precision, recall and F1-Score. Among them, confusion matrix is a table layout used to describe and visualize the performance of a trained model on a testing set<sup>37</sup>.

### **Visualization methods**

Deep learning models are often perceived as opaque or “black boxes.” While their remarkable performance is undeniable, but their lack of interpretability poses a challenge for widespread adoption, particularly in fields like food science. Hence, this study employs three visualization techniques to elucidate the inner workings of the optimal models obtained through BO, enhancing its interpretability and credibility in food applications.

#### *Feature visualization*

Features are generally the physical characteristics of an object that can be used to distinguish it from other objects. A fruit has many physical characteristics including color, texture, shape, and size, which are used by traditional fixed-features based machine learning methods for recognition and classification tasks, such as detecting the defects or maturity of fruits<sup>38,39</sup>. However, such fixed, simple features-based classifiers are not robust or suitable for complex tasks because fruits have many inter-class and intra-class similarities and variations, especially inter-class similarities and intra-class variations pose significant challenges<sup>10,40</sup>. In contrast to fixed-simple features-based machine learning methods, CNNs are able to automatically learn and integrate features from training images and use them for classification task<sup>41</sup>. Specifically, the convolutional layers act as feature extractors for the input images whose dimensionality is subsequently reduced by the pooling layers, and the fully connected layers act as classifiers<sup>42,43</sup>.

#### *Strongest activations*

The purpose of presenting the strongest activations was to observe and compare how the optimal models recognize pears. In the strongest activation images, strong positive activation is shown by white pixels and strong negative activation is shown by black pixels<sup>43</sup>. We focus on the white areas in the images as they indicate the areas recognized by the optimal models<sup>44</sup>. To elucidate the discriminative features learned by the model, one image of pear was randomly select from the testing set and its series of images are fed into the trained model separately. Then, to show the strongest activations of the last convolutional layer<sup>11</sup>. This approach offers insights into the regions of interest within the input images that contribute significantly to the model's classification decisions.

#### *LIME*

Since LIME typically uses simple and more interpretable models (e.g. linear model or decision tree model) to locally approximate the predictions of the target black-box model, LIME was applied here to figure out how

optimal models make classification decisions on pears in order to further improve the interpretability of the models<sup>45,46</sup>.

To provide interpretable insights at the instance level, the image of pear was randomly selected from the testing set and fed into the trained model to show the corresponding LIME image using the method “local interpretable model-agnostic explanations”. This method facilitates the understanding of how the model makes decisions by highlighting the important regions of the input image that influence the model’s output, thereby enhancing interpretability and trustworthiness.

### Computer configuration and model hyperparameters

The BO for Deep Learning were implemented using MATLAB R2023b version, running on the same personal desktop with Intel(R) Core i9-13900kF CPU\*1, 32G RAM\*2 and NVIDIA® GeForce RTX 4090 GPU\*1, and trained by SGD with Momentum. In addition, the hyperparameters of deep learning based on BO comprise both fixed values and those within specified search ranges. Among the fixed hyperparameters, learn rate drop factor=0.1, learn rate drop period=40, minibatch size=256, and max epochs=60. In addition, perform objective function evaluation 30 times to better exploit the BO.

## Results and discussion

### Performance of the optimal models

Tables 2 and 3 present the results from the BO process for hyperparameter tuning in deep learning model. Each row represents an iteration of the BO process. Once employed this type of table can be useful for tracking the progress of hyperparameter optimization, observing how changes in hyperparameters affect model performance, and determining the best combination of hyperparameters for optimal model performance. Based on these two tables, the optimal models and their hyperparameters under the two data set configurations were determined. The optimal models trained using dataset A and dataset B are defined as model A and model B respectively.

Figures 3 and 4 show the progression of minimum objective values over the number of function evaluations during the BO process. These graphs help in visualizing the effectiveness of the BO, specifically in how quickly

Iteration	Evaluate result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	SectionDepth	Initial Learnrate	Momentum	L2Regularization
1	Best	0.011317	81.741	0.011317	0.011317	2	0.19179	0.95759	2.04e-07
2	Accept	0.038272	56.976	0.011317	0.012935	1	0.35261	0.83669	0.00042837
3	Accept	0.012243	78.148	0.011317	0.011356	2	0.027656	0.86189	2.9503e-07
4	Accept	0.03251	55.573	0.011317	0.011342	1	0.011394	0.84719	1.0957e-10
5	Accept	0.012654	78.279	0.011317	0.011324	2	0.034893	0.97771	1.013e-10
6	Accept	0.89228	101.48	0.011317	0.012079	3	0.63179	0.80198	0.0067596
7	Accept	0.88796	79.599	0.011317	0.011338	2	0.90473	0.80143	0.0060218
8	Accept	0.027778	77.945	0.011317	0.011335	2	0.69359	0.88955	0.00054117
9	Accept	0.012449	75.906	0.011317	0.011325	2	0.030223	0.92472	8.3496e-07
10	Best	0.011008	101.73	0.011008	0.01096	3	0.1125	0.94039	1.2771e-10
11	Accept	0.023868	58.908	0.011008	0.011009	1	0.96687	0.85375	4.3014e-08
12	Accept	0.11708	84.737	0.011008	0.011024	1	0.21905	0.87412	0.0027897
13	Accept	0.012449	162.74	0.011008	0.011015	2	0.019318	0.90549	1.6603e-07
14	Accept	0.32613	205.62	0.011008	0.010972	3	0.70225	0.96842	0.0055678
15	Accept	0.012551	209.7	0.011008	0.010965	3	0.87991	0.88112	1.1254e-10
16	Accept	0.024691	115.72	0.011008	0.011021	1	0.3082	0.96016	1.3316e-10
17	Accept	0.065844	213.6	0.011008	0.011078	3	0.073997	0.91219	0.0084137
18	Accept	0.014815	157.87	0.011008	0.011056	2	0.69294	0.94309	3.0238e-07
19	<b>Best</b>	<b>0.010082</b>	<b>98.946</b>	<b>0.010082</b>	<b>0.0091958</b>	<b>3</b>	<b>0.03476</b>	<b>0.88586</b>	<b>1.9761e-07</b>
20	Accept	0.01893	56.27	0.010082	0.0090156	1	0.14646	0.92085	1.0349e-10
21	Accept	0.020885	56.512	0.010082	0.00873	1	0.064067	0.84708	0.0002328
22	Accept	0.018313	57.275	0.010082	0.0083593	1	0.2996	0.86691	1.2213e-10
23	Accept	0.017798	57.211	0.010082	0.010177	1	0.19813	0.84217	9.1636e-07
24	Accept	0.020782	56.127	0.010082	0.0085004	1	0.011168	0.95151	1.2674e-10
25	Accept	0.01893	55.777	0.010082	0.0086326	1	0.36189	0.93413	5.5029e-09
26	Accept	0.01142	100.75	0.010082	0.0068073	3	0.25867	0.95297	8.8809e-09
27	Accept	0.020885	56.994	0.010082	0.0065373	1	0.034056	0.89705	5.7064e-06
28	Accept	0.021399	57.158	0.010082	0.006589	1	0.43922	0.89712	1.2497e-10
29	Accept	0.022737	56.629	0.010082	0.0053253	1	0.34545	0.87649	8.2626e-09
30	Accept	0.021091	56.253	0.010082	0.0052978	1	0.021241	0.9676	2.2272e-09

**Table 2.** The results from the BO process for hyperparameter tuning in a deep learning model on dataset A. Bold values indicate the optimal model A.

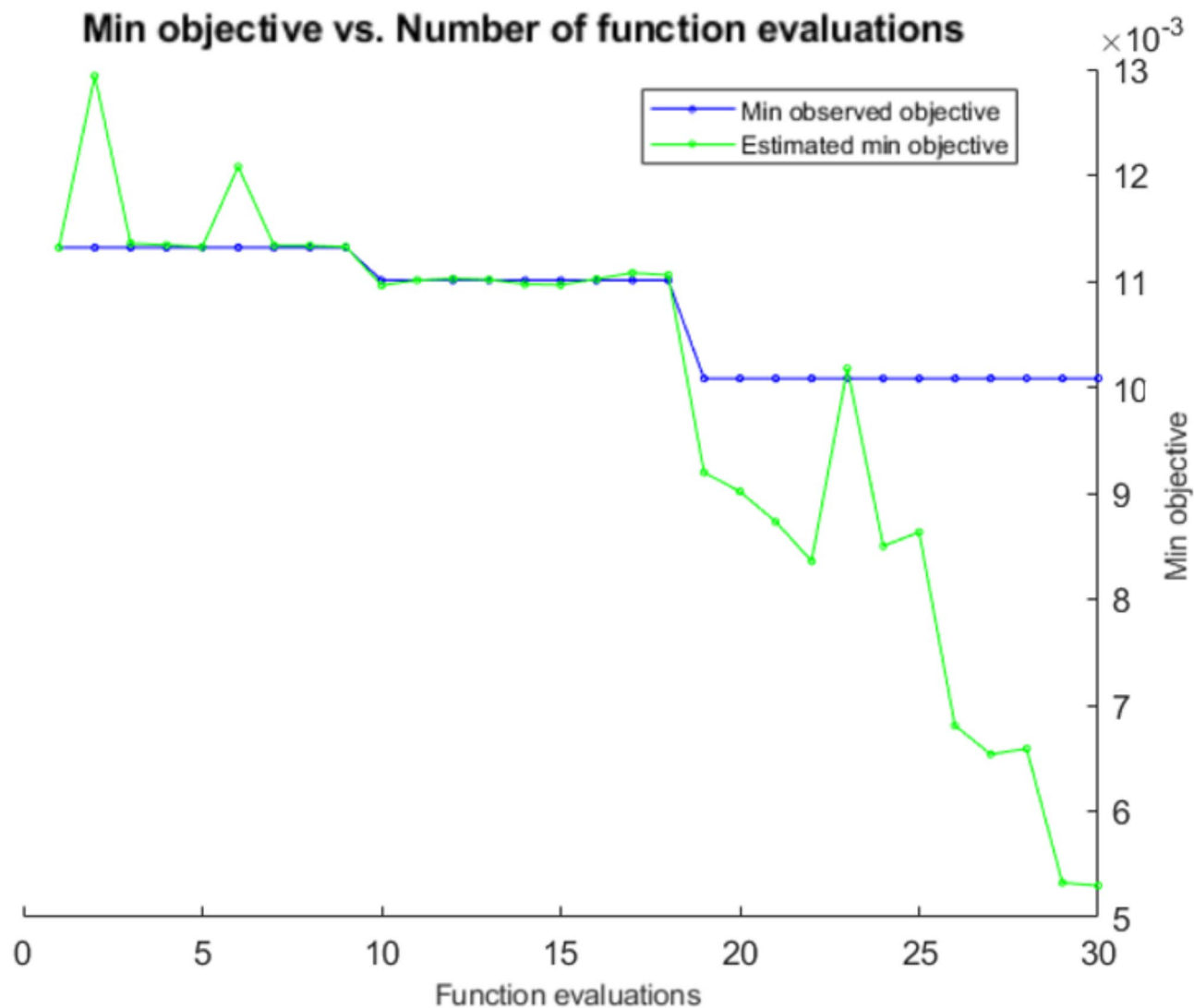
Iteration	Evaluate result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	SectionDepth	Initial Learnrate	Momentum	L2Regularization
1	Best	0.019976	23.588	0.019976	0.019976	3	0.037896	0.9709	0.0029114
2	Accept	0.023548	18.135	0.019976	0.021716	2	0.41852	0.93015	6.2296e-05
<b>3</b>	<b>Best</b>	<b>0.011113</b>	<b>18.003</b>	<b>0.011113</b>	<b>0.011114</b>	<b>2</b>	<b>0.040348</b>	<b>0.88583</b>	<b>1.6317e-10</b>
4	Accept	0.014949	17.499	0.011113	0.011113	2	0.041195	0.8475	0.0030639
5	Accept	0.018521	18.199	0.011113	0.011217	2	0.5679	0.85755	1.0152e-10
6	Accept	0.012171	18.445	0.011113	0.011115	2	0.038224	0.9006	1.097e-07
7	Accept	0.013229	18.069	0.011113	0.011115	2	0.053222	0.89024	0.00086938
8	Accept	0.016272	18.047	0.011113	0.011115	2	0.027491	0.87909	1.6043e-10
9	Accept	0.014288	18.661	0.011113	0.012981	2	0.043773	0.88987	1.0361e-10
10	Accept	0.013229	18.096	0.011113	0.012987	2	0.042696	0.9223	4.6594e-07
11	Accept	0.012303	17.819	0.011113	0.012727	2	0.041818	0.91222	0.00034195
12	Accept	0.012436	18.437	0.011113	0.012599	2	0.040302	0.90969	0.0020171
13	Accept	0.018786	18.107	0.011113	0.01261	2	0.10979	0.92448	0.0030042
14	Accept	0.032941	18.315	0.011113	0.012561	2	0.010212	0.80056	1.3194e-05
15	Accept	0.053049	14.671	0.011113	0.012623	1	0.044569	0.90905	1.1936e-06
16	Accept	0.024342	22.386	0.011113	0.012629	3	0.35213	0.92954	7.0516e-06
17	Accept	0.01786	22.052	0.011113	0.012586	3	0.010335	0.8007	0.0031439
18	Accept	0.015875	18.677	0.011113	0.012725	2	0.035865	0.9761	0.000228
19	Accept	0.011509	22.443	0.011113	0.012732	3	0.050761	0.80023	1.3466e-10
20	Accept	0.012039	21.827	0.011113	0.012753	3	0.097811	0.80168	7.7094e-08
21	Accept	0.014949	22.504	0.011113	0.012871	3	0.058799	0.80025	0.0008059
22	Accept	0.013891	21.871	0.011113	0.012844	3	0.097644	0.8028	1.1774e-10
23	Accept	0.015214	17.955	0.011113	0.012885	2	0.14488	0.80265	2.775e-10
24	Accept	0.017066	22.921	0.011113	0.012877	3	0.87727	0.80053	2.4065e-07
25	Accept	0.022887	22.314	0.011113	0.012856	3	0.24218	0.80168	2.4954e-08
26	Accept	0.013097	22.444	0.011113	0.01285	3	0.030408	0.80163	7.924e-09
27	Accept	0.011509	22.597	0.011113	0.012021	3	0.049207	0.80091	1.0803e-08
28	Accept	0.014552	21.68	0.011113	0.012857	3	0.058919	0.80452	1.1874e-10
29	Accept	0.024871	18.575	0.011113	0.012821	2	0.97767	0.80453	2.2267e-10
30	Accept	0.011642	22.352	0.011113	0.012803	3	0.032979	0.80029	0.00013078

**Table 3.** The results from the BO process for hyperparameter tuning in a deep learning model on dataset B. Bold values indicate the optimal model B.

it can converge to a near-optimal solution and the stability of its estimations throughout the process. The X-axis is the number of function evaluations, representing how many times the optimization algorithm has tested different hyperparameter configurations. The Y-axis is the minimum objective value, which indicates the value of the performance metric achieved up to each evaluation in the optimization process, lower values indicate better model performance. The blue line (Min observed objective) represents the actual objective value observed at each evaluation of the optimization, it essentially tracks the lowest error achieved during the optimization process. This line tends to decrease or remain flat over time, indicating that the optimization process is either finding better solutions or maintaining the best solution found so far. The green line (Estimated min objective) represents the estimated minimum objective value that the Bayesian model predicts based on the data it has gathered from previous evaluations. It provides a prediction of the potential minimum error that can be achieved with the given hyperparameters. The fluctuations in this line reflect the BO's exploration of the hyperparameter space and its estimation of where the lowest error might be found. These graphs help illustrate the dynamics between observed changes of deep learning model performance and predicted by the Bayesian model.

Specifically, in Fig. 3, during 0–9 evaluations, there is significant fluctuation in the estimated min objective, which is expected as the optimization process explores the hyperparameter space in initial stage. The min observed objective (blue line) remains relatively flat, suggesting that the optimization process has not found significantly better hyperparameters during this period. During 9–18 evaluations, the blue and green lines remain similar, they both have a decrease at the beginning. Among them, the green line shows several ups and downs, indicating continued exploration and refinement of the hyperparameter space. The blue line remains flat, indicating that the hyperparameters at the tenth evaluation were the optimal hyperparameters during this period. During 18–30 evaluations, the estimated min objective drops significantly, the minimum observed objective also decreases and subsequently remains stable, thus reflecting the actual improvement in the model's performance. Overall, Fig. 3 indicates a successful BO process that progressively improves the model's performance on the verification set and finally obtained the optimal model A.

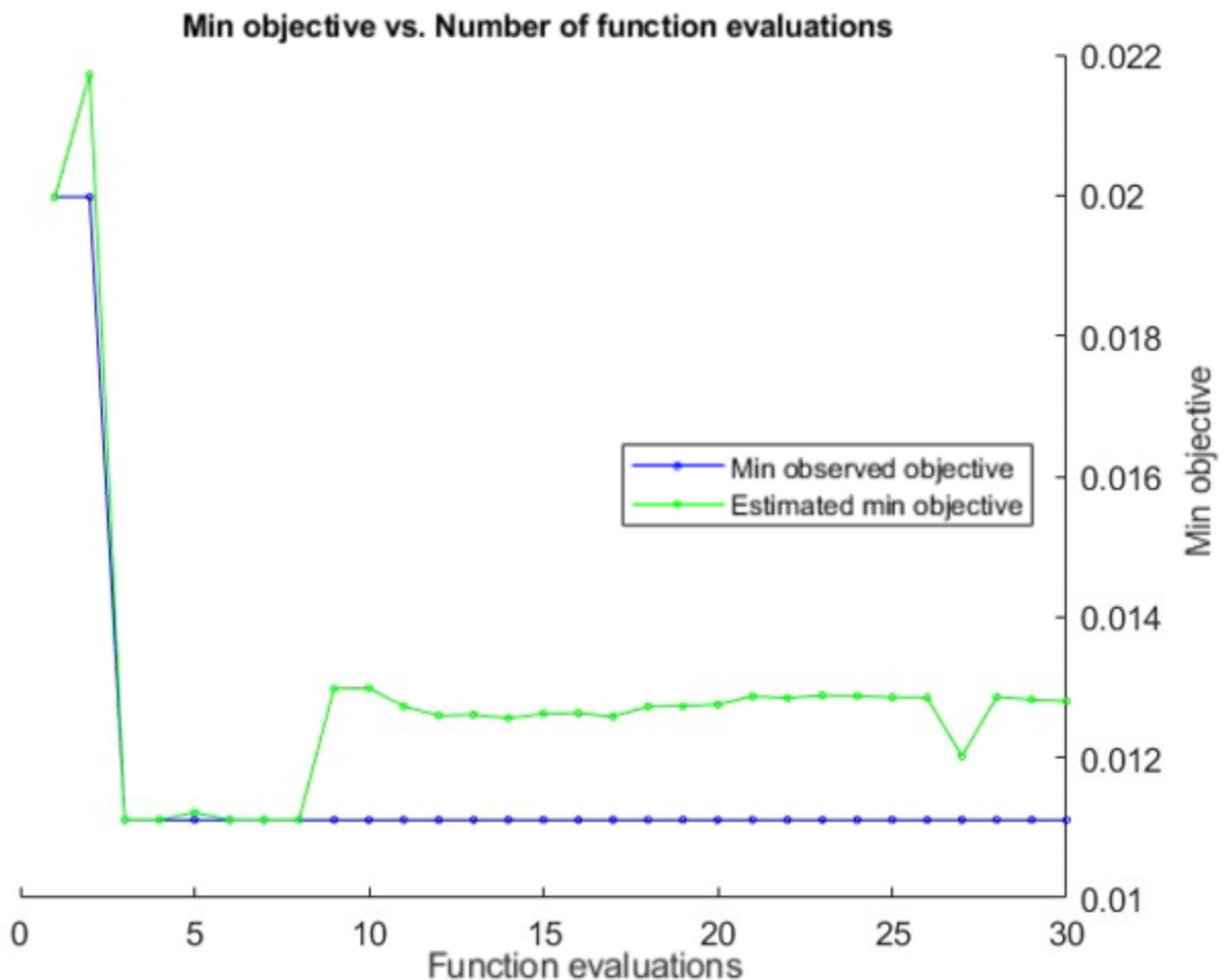
In Fig. 4, the changes of the green and blue lines are simpler than those in Fig. 3. After the two lines declined rapidly in the initial 0–3 evaluations, then, the green line entered a fluctuating state, and the blue line remained



**Fig. 3.** Minimum objective versus Number of function evaluations on dataset A.

flat until the end of the evaluation. The green line's fluctuations show the BO's exploration and refinement process, trying to estimate and find even better hyperparameters. However, the lack of significant improvement in the blue line indicates that the process is primarily exploring around a local optimum. It means the BO process quickly identified a set of hyperparameters that significantly reduced the model error on the verification set, and were not find significantly better hyperparameters in subsequent evaluations. This stability is a positive sign and indicates consistent performance. Overall, Fig. 4 also indicates a successful optimization process where a high quality set of hyperparameters was found early, with subsequent evaluations confirming its effectiveness and finally get the optimal model B.

The confusion matrices of the two optimal models for the testing sets in dataset A and dataset B are shown in Figs. 5 and 6 respectively. Correct predictions for each category are located on the diagonal of the confusion matrix and marked in blue, while incorrect predictions are marked in pink. In Figure.5, the optimal model (model A) trained using training dataset A (30% of which is validation data) demonstrates significantly higher accuracy, achieving 97.29%. In contrast, in Fig. 6, the optimal model (model B) trained with training dataset B (70% of which is validation data) shows increased misclassifications across all categories, resulting in an accuracy of 90.39%. In addition, the Precision, recall and F1-Score of the optimal model A and B are shown in Table 4. Figures 5 and 6, and Table 4 indicate that the dataset configuration has a substantial impact on the classification results of Bayesian optimized deep learning models. On the one hand, relatively more training data and less test data help to improve the overall accuracy of the optimal model. The ratios of training and test data used by the model A and model B are 21:10 and 1:10 respectively. In real applications, the test set is often infinitely large<sup>47</sup>, so Dataset B is more consistent with real applications. On the other hand, a lot of noise is added to the data. As shown in the Fig. 2, when the noise intensity is high, our human eyes even cannot recognize that the picture contains pears. Even so, the model B still achieved an overall accuracy of more than 90%. In addition, we tested the deep learning based on BO using the original data of pear in "Fruit-360" (all parameters are consistent, and



**Fig. 4.** Minimum objective versus Number of function evaluations on dataset B.

the verification set accounts for 20% of the training set), and the optimal model accuracy is 100%. We also tested different proportions of the validation set within the training set, ranging from 10 to 80%. As shown in Table 5, on Dataset A, the optimal model's accuracy fluctuated between 95.68% and 97.29%, while on Dataset B, the optimal model's accuracy varied between 86.30% and 90.39%. This indicates that in this study, the more training data available, the less the optimal model is affected by the size of the validation set.

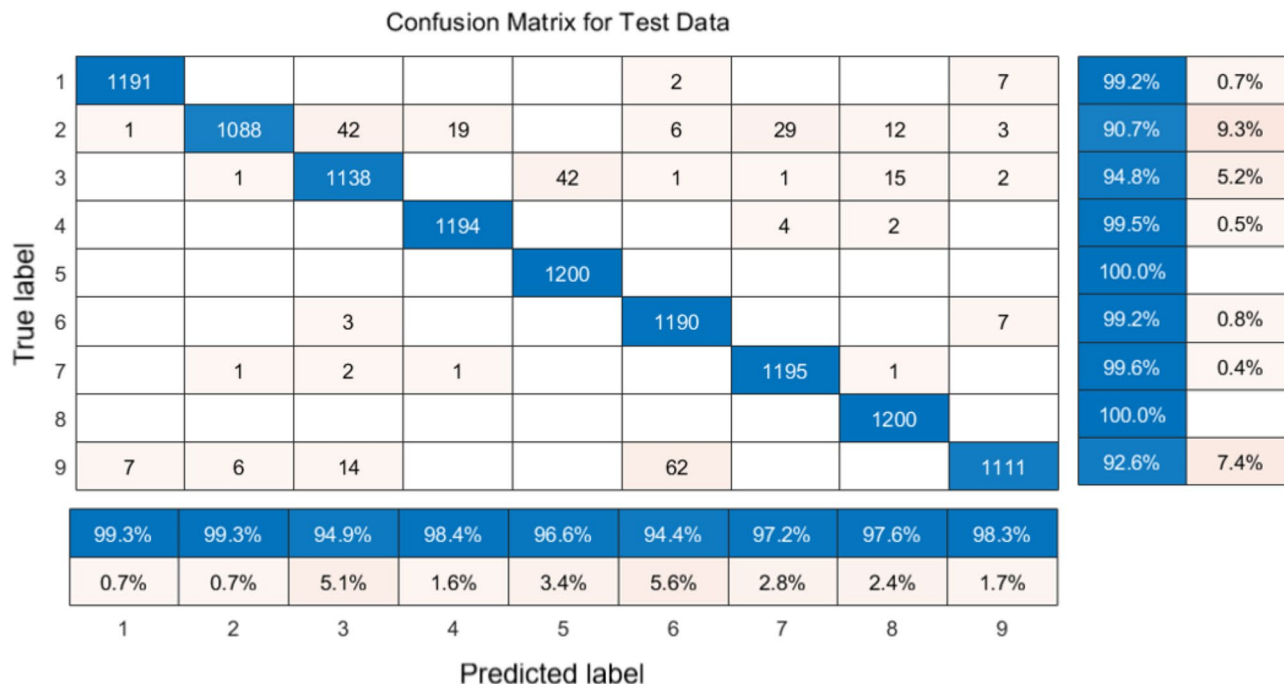
### Model interpretability analysis

#### Feature visualization

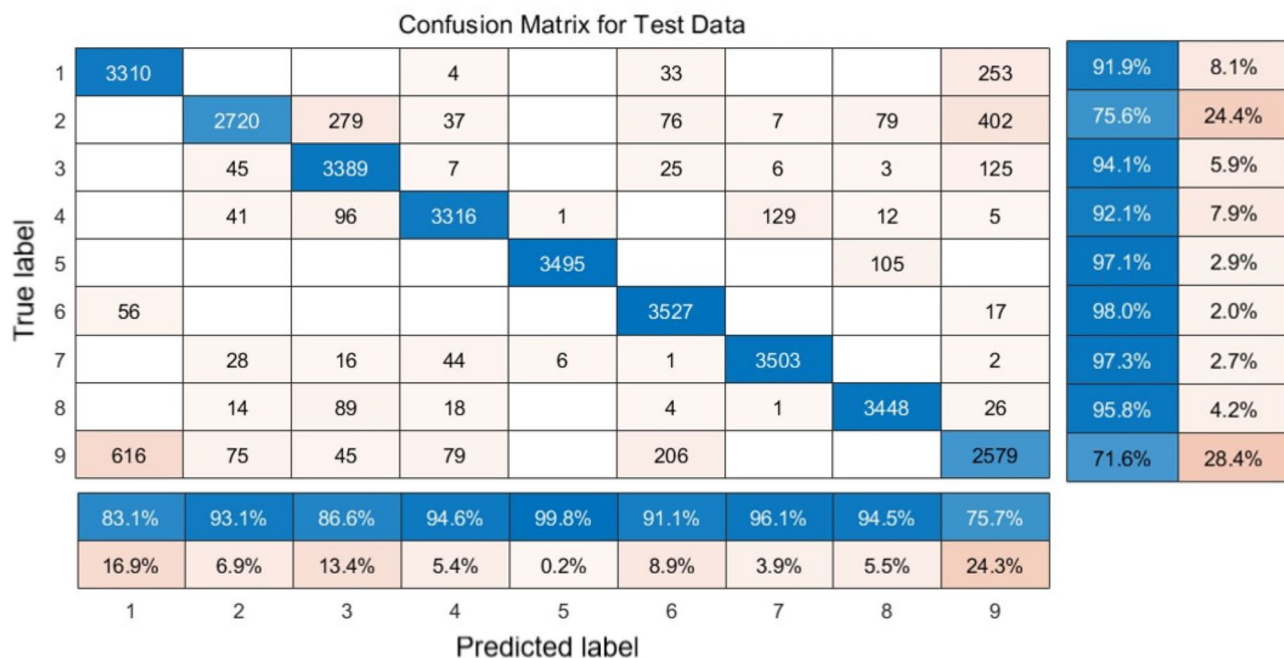
In this study, the feature visualization of the last fully connected layer of the two optimal models was used to explain to us how the optimal models obtained through BO under different dataset configurations build understanding of pear images, i.e. the common and high-level features of each type of pear learned by the optimal models from the training set, as shown in Fig. 7<sup>48</sup>.

It is evident that the feature visualization images of the two optimal models exhibit distinct patterns or styles, even for the same class of pears. This suggests that different models interpret the same class of pear in varied ways. Additionally, although some classes of pears appear very similar in appearance, their corresponding feature visualization images generated by different models remain distinct. This indicates that two optimal models have successfully learned the true differences between classes of pears<sup>41</sup>. For example, the first (Pear) and sixth (Pear Monster) types, as well as the fourth (Pear Forelle) and seventh (Pear Red) types in Fig. 1 have certain similarities in shape, size and color. But their corresponding feature visualization images are obviously different, which is similar to our previous research results<sup>11</sup>. Furthermore, it should be noted that the color of the sixth category in the two feature visualization images is significantly brighter than first category, and the seventh category is more reddish than the fourth category, which is consistent with the color of these pears to a certain extent.

In addition, the two optimal models are both series network but different in depths, model A has 34 layers and model B has 25 layers. The feature visualization images of the two models are abstract and difficult to understand. In particular, Model A appears more abstract and intricate than Model B. This phenomenon is due



**Fig. 5.** Confusion matrix of the optimal model A for test set in dataset A.



**Fig. 6.** Confusion matrix of the optimal model B for test set in dataset B.

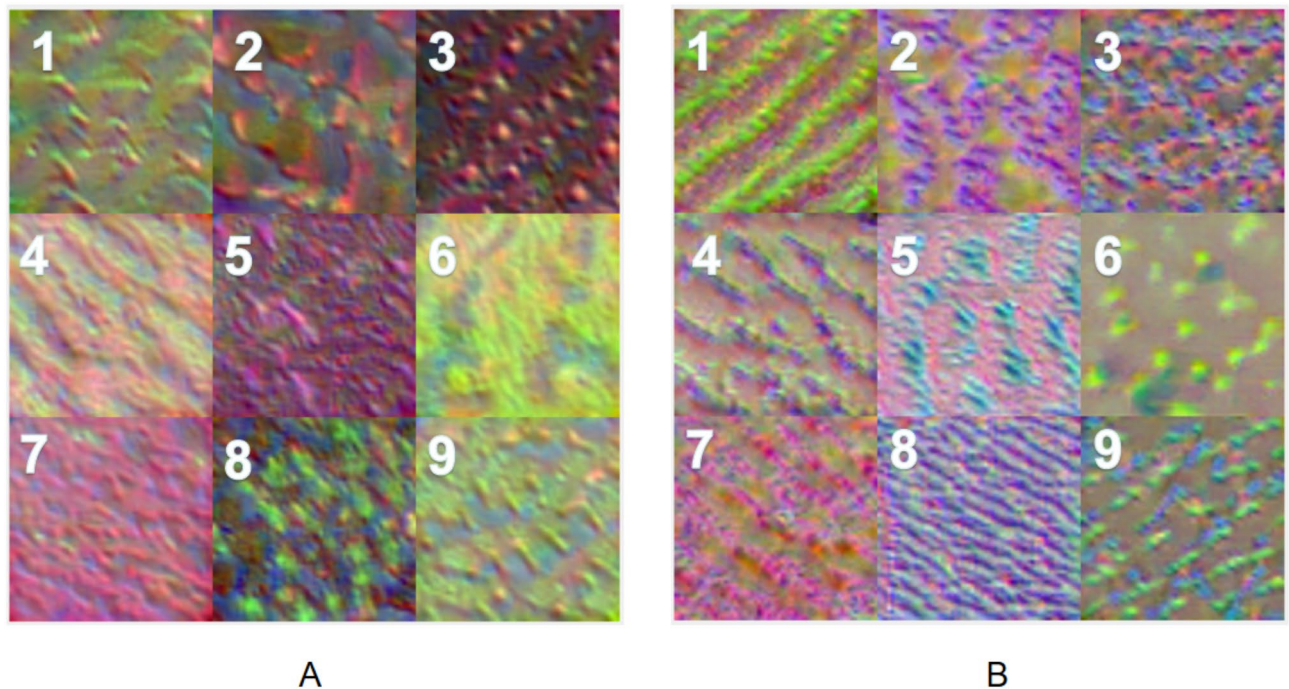
to the different depths of the models because CNNs typically build understanding of images in a hierarchical way over many layers, where earlier layers learn basic and low-level features such as colours, edges, textures, or shapes, and later layers learn and integrate simple features (learned by earlier layers) into increasing complex and abstract features such as patterns, parts or objects, so that the last fully connected layer learns the high-level features of each class and used for classification, but sometimes the high-level features are too abstract to be interpreted<sup>45,49</sup>. Based on this, since deeper layers can learn the combinations of features learned by the previous layers, the deeper model implies more convolutional layers, which can extract more advanced and complex features than the relatively shallow model<sup>11,50</sup>.

Dataset A Optimal model A				Dataset B Optimal model B			
Class	Precision (%)	Recall (%)	F1-Score (%)	Class	Precision (%)	Recall (%)	F1-Score (%)
1	99.33	99.29	99.25	1	83.12	87.31	91.94
2	99.27	94.77	90.67	2	93.06	83.40	75.56
3	94.91	94.87	94.83	3	86.59	90.20	94.14
4	98.35	98.92	99.50	4	94.61	93.34	92.11
5	96.62	98.28	100	5	99.80	98.42	97.08
6	94.37	96.71	99.17	6	91.09	94.41	97.97
7	97.23	98.39	99.58	7	96.08	96.69	97.31
8	97.56	98.77	100	8	94.54	95.16	95.78
9	98.32	95.36	92.58	9	75.65	73.59	71.64

**Table 4.** Precision, recall and F1-Score of the optimal model A and B.

Proportion of validation set in training set	Dataset A	Dataset B
	Accuracy (%)	Accuracy (%)
10%	97.24	86.30
20%	96.06	88.42
30%	97.29	87.50
40%	96.07	88.63
50%	95.90	90.15
60%	96.44	88.03
70%	96.53	90.39
80%	95.68	88.59

**Table 5.** Different proportion of validation set in training set and optimal model accuracy.

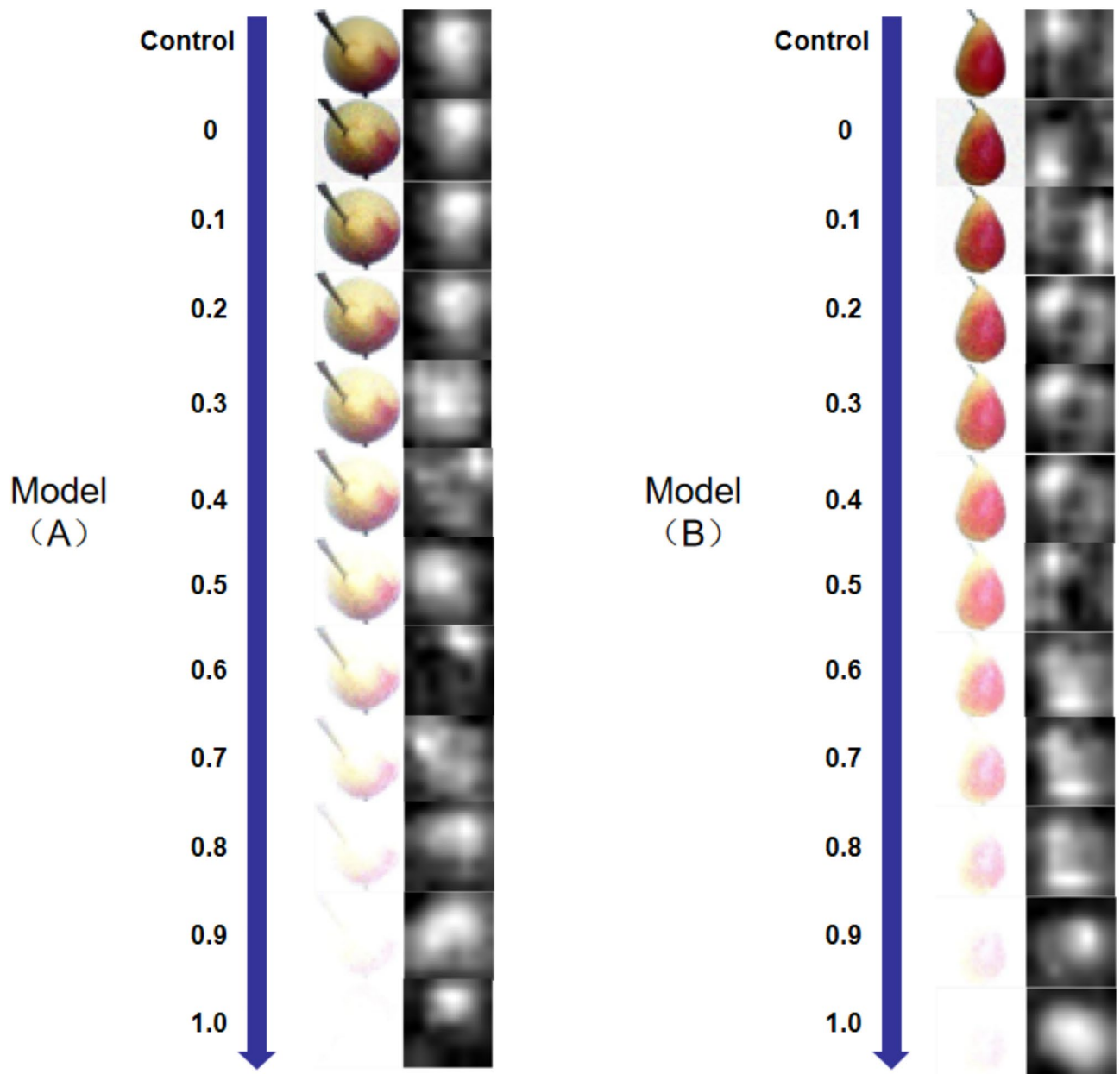


**Fig. 7.** Feature visualization of the last fully connected layer of the optimal models (1. Pear, 2. Pear 2, 3. Pear Abate, 4. Pear Forelle, 5. Pear Kaiser, 6. Pear Monster, 7. Pear Red, 8. Pear Stone and 9. Pear Williams.)

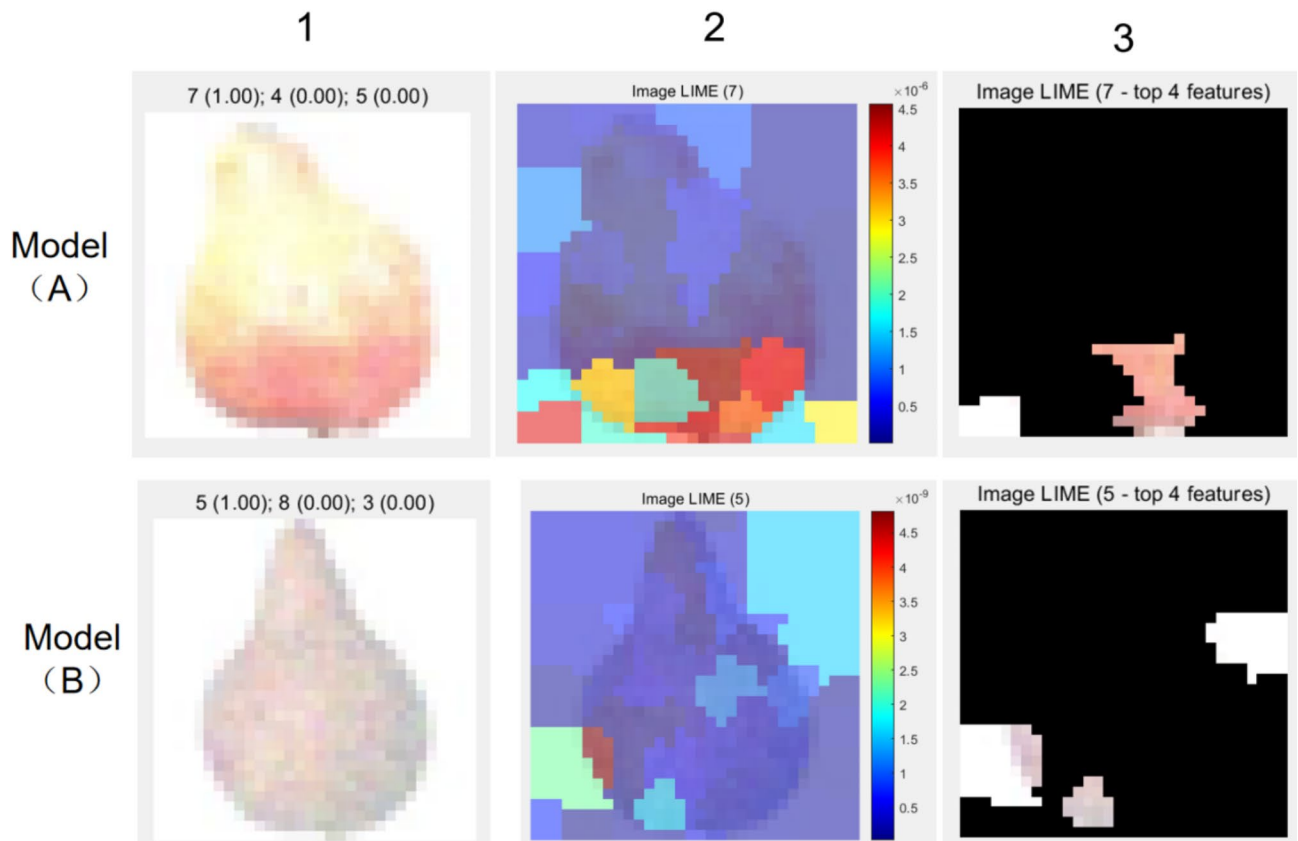
*Strongest activations*

Figure 8 shows a series of images of the same type of pears randomly selected from different test sets and these images containing different levels of added noise, and the corresponding strongest activations generated by the last convolutional layer of the two optimal models. In previous research<sup>11</sup>, we revealed how a CNN-based model classifies different fruits, these results suggest that models with different frameworks and depths recognize fruits in different ways. In this study, we further examine the impact of Gaussian white noise in images on the strongest activation of the last convolutional layer of the optimal model, and analyze patterns or differences in the way the model responds to noise.

Specifically, compared with the strongest activation of Model B, Model A performs more consistently across images with different noise levels, indicating that Model A have better feature extraction capabilities, especially when processing noisy images achieves stronger feature detection. While Model B requires higher quality and less noisy images to remain effective, which also corresponds to the accuracy of the two optimal models. Even so, this does not mean that model B is unacceptable in practical applications, because we are not always exposed to low signal-to-noise ratio images in the real world. For example, when Gaussian white noise intensity more than 0.5, it becomes challenging for the human eye to discern image content. In addition, Model B achieved an accuracy of greater than 90% on the challenging test set.



**Fig. 8.** Images of pears with different Gaussian white noise added and their corresponding strongest activations in the last convolutional layer of the two optimal models.



**Fig. 9.** Understanding of two optimal models by using LIME technique.

### LIME

Figure 9 shows the feature importance maps corresponding to model A and B as determined by the LIME. Specifically, the first column shows the classification results of the two optimal models from randomly selected images of pears from different test sets, i.e. the three categories that received the highest classification probabilities are displayed at the top of the image. The second column shows the recognition region of the image that the model used to classify. And the third column shows the most important features determined by each model<sup>50</sup>. For instance, in row 1 column 1, model A classified the pear image as Class 7 (Pear Red) with 100% probability, and Class 4 (Pear Forelle) and Class 5 (Pear Kaiser) with 0 probability. In row 1 column 2, the feature map shows which regions of the image were important for the classification of the Pear Red (Class 7). According to the chromaticity bar, the red regions have a high importance, i.e. model A focuses on the lower part of the pear to predict as Class 7, and the prediction accuracy decreases when these regions are removed<sup>46</sup>. For row 1 column 3, it is a masked image and the visible regions need to be focused on as it indicates the most important features identified by model A, it corresponding to the important regions in the row 1 column 2 image.

Compared with the LIME map of model A, model B has fewer important areas (warm tone areas). In the top4 features images, the overlapping area between the features of model B and pear is less than that of model A. Therefore we can infer that if the application requires high accuracy and detail (e.g., quality control in pears processing), Model A might be preferred. However, for applications needing faster processing with reasonable accuracy (e.g., tasks with less noise in images), Model B could be more suitable, although its accuracy is 90.39%, which is lower than Model A's 97.29%, but Model B has fewer layers.

Based on the above, Sect. 3.2 provides an insight into the two optimal models through three visualization methods to explore the models' working mechanisms in this task. Specifically, feature visualization images show the different understanding of pear images by two optimal models, strongest activations and LIME images show how and why optimal models make classification decisions. These results help us to explain model predictions and build trust in deep learning for practical applications<sup>45</sup>. In addition, it can also help us optimize the deep learning based on BO and further improve the performance of the optimal model.

### Conclusion

Automated and efficient fruit variety recognition and classification systems are essential in agricultural and food practices as they can significantly reduce labor costs and enhance the economic benefits throughout the fruit supply chain, from harvesting to sales. In this study, BO deep learning were employed to identify and classify images of nine pear varieties with added noise on two challenging datasets, which are close to real agricultural applications. Based on the figure of minimum objective vs. number of function

evaluations, and the table of the results from the BO process for hyperparameter tuning in a deep learning model, two optimal models were identified. Important findings are as follows.

- (1) Dataset configuration significantly impacts the classification accuracy of the BO optimal models, i.e. the optimal model A achieved an accuracy of 97.29% on dataset A (training-to-testing ratio = 21:10), the optimal model B achieved an accuracy of 90.39% on dataset B (training-to-testing ratio = 1:10), and on the original Fruit360 dataset, the accuracy of the BO optimal model reached 100% (training-to-testing ratio = 12:5).
- (2) Although the BO process explores the hyperparameter space based on model error on the validation set, the proportion of the validation set within the training set has a relatively minor effect on the performance of the optimal models, especially when the training set is large. Specifically, we set the proportion of the validation set within the training set from 10% to 80%, and the accuracy of the optimal models optimized using different validation sets fluctuated between 95.68% and 97.29% on dataset A, and between 86.30% and 90.39% on dataset B.
- (3) Feature visualization revealed that the two optimal models have different understandings of different pears, but for certain types of pears (Pear, Pear Forelle, Pear Monster, Pear Red), the color might influence the classification results. The strongest activations demonstrated the two optimal models used which areas of the images to classify pear images with different noise levels. LIME showed the important features used by the two optimal models for making classification decisions. And the results indicated that the number of features might correlate with the classification accuracy of the optimal models. That is the more warm-toned features, the higher the accuracy of the corresponding optimal model.

These results not only showcase the excellent performance of BO deep learning in classifying noisy pear images but also address the challenges faced by deep learning in agricultural applications, promoting the widespread application of deep learning in the food field. Our future work will focus on the following aspects, (1) Increasing the variety and number of fruits used for classification, not limited to a single type of fruit, aiming to develop a general model for the fruit and even the food sector. (2) Increasing the number of hyperparameters for BO and integrating the data augmentation process with the training of BO deep learning models. (3) Continuing to study the interpretability of deep learning-based models to reveal the feature evolution mechanisms of black-box models, thereby enhancing the trust of users in the food sector in deep learning.

### Data availability

The datasets during the current study are available from Fruits-360 (<https://www.kaggle.com/moltean/fruits>, Version: 2020.05.18.0, accessed on 10 October 2022).

Received: 27 May 2024; Accepted: 11 April 2025

Published online: 30 September 2025

### References

1. Saquet, A. Storage of Pears. *Sci. Hortic.* **246**, 1009–1016 (2019).
2. Colavita, G. M. et al. *Pear[M]//Temperate Fruits* 107–182 (Apple Academic, 2021).
3. Li, X. et al. Chemical composition and antioxidant and anti-inflammatory potential of peels and flesh from 10 different Pear varieties (*Pyrus* spp). *Food Chem.* **152**, 531–538 (2014).
4. EFSA. The EFSA Comprehensive European Food Consumption Database; EFSA: Parma, Italy, (2011).
5. Ulaszewska, M. et al. Food intake biomarkers for Apple, Pear, and stone fruit. *Genes Nutr.* **13**, 29 (2018).
6. Cortés, V., Cubero, S., Blasco, J., Aleixos, N. & Talens, P. In-line application of visible and near-infrared diffuse reflectance spectroscopy to identify Apple varieties. *Food Bioprocess. Technol.* **12**, 1021–1030 (2019).
7. Stănică, F., Cean, I. & Peticilă, A. G. Parallel trident—an efficient planting system for Pear orchards[C]//XXIX international horticultural Congress on horticulture: sustaining lives, livelihoods and landscapes (IHC2014) **1130**, 157–162. (2014).
8. Biswas, B., Ghosh, S. K. & Ghosh, A. A. Robust multi-label fruit classification based on deep convolution neural network. In *Computational Intelligence in Pattern Recognition*, Springer: 105–115. (2020).
9. Jakobek, L. et al. Indigenous Apple varieties, a fruit with potential for beneficial effects: Their quality traits and bioactive polyphenol contents. *Foods* **9**, 52 (2020).
10. Ghazal, S., Munir, A. & Qureshi, W. S. *Computer Vision in Smart Agriculture and Precision Farming: Techniques and applications*[J] (Artificial Intelligence in Agriculture, 2024).
11. Yu, F., Lu, T. & Xue, C. Deep learning-based intelligent Apple variety classification system and model interpretability analysis[J]. *Foods* **12**(4), 885 (2023).
12. An, S. et al. Predicting physicochemical properties of Papayas (*Carica Papaya* L.) using a convolutional neural networks model approach. *J. Food Sci.*, (2024).
13. Anjali, J. A. et al. *State-of-the-art non-destructive Approaches for Maturity Index Determination in Fruits and Vegetables: Principles, Applications, and Future directions*[J] 656 (Food Production, Processing and Nutrition, 2024). 1.
14. Akter, T. et al. A comprehensive review of external quality measurements of fruits and vegetables using nondestructive sensing technologies. *J. Agric. Food Res.*, 101068. (2024).
15. Amarasinghe, C. & Ranasinghe, N. *Digital Food Sensing and Ingredient Analysis Techniques To Facilitate Human-Food Interface Designs*[J] (ACM Computing Surveys, 2024).
16. Yu, G. et al. Quality detection of watermelons and muskmelons using innovative nondestructive techniques: A comprehensive review of novel trends and applications[J]. *Food Control*, 110688. (2024).
17. Sajitha, P. et al. A review on machine learning and deep learning image-based plant disease classification for industrial farming systems. *J. Industrial Inform. Integr.*, : 100572. (2024).
18. Deng, Z. et al. Deep Learning in Food Authenticity: Recent Advances and Future trends, 104344 (Trends in Food Science & Technology, 2024).
19. Pratondo, A. & Novianty, A. Pear classification using machine learning[C]//2022 IEEE 10th Conference on Systems, Process & Control (ICSPC). IEEE, 186–190. (2022).

20. Ismail, N., Owais, A. & Malik Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Inform. Process. Agric.* **9**(1), 24–37 (2022).
21. Rojas-Aranda, J. et al. Fruit classification for retail stores using deep learning. Pattern Recognition: 12th Mexican Conference, MCPR 2020, Morelia, Mexico, June 24–27, Proceedings 12. Springer International Publishing, 2020. (2020).
22. Gill, H. et al. Fruit type classification using deep learning and feature fusion. *Comput. Electron. Agric.* **211**, 107990 (2023).
23. Naranjo-Torres, J. et al. A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* **10**(10), 3443 (2020).
24. Sharafudeen, M. & SS, V. C. Multimodal siamese framework for accurate grade and measure estimation of tropical fruits, (IEEE Transactions on Industrial Informatics, 2023).
25. Siddiqi, R. July. Effectiveness of transfer learning and fine tuning in automated fruit image classification. In Proceedings of the 2019 3rd International Conference on Deep Learning Technologies, Xiamen, China, 5–7, pp. 91–100. (2019).
26. Kodors, S., Lakis, G., Zhukov, V. & Bartulsons, T. Pear and apple recognition using deep learning and mobile. *Eng. Rural Dev.* **20**, 1795–1800 (2020).
27. Choudhary, K. et al. Recent advances and applications of deep learning methods in materials science. *Npj Comput. Mater.* **8**(1), 59 (2022).
28. Rahman, M. M. et al. A deep CNN approach to detect and classify local fruits through a web interface. *Smart Agric. Technol.* **5**, 100321 (2023).
29. Hernández-García, A. & König, P. Further advantages of data augmentation on convolutional neural networks. In International Conference on Artificial Neural Networks. 95–103Springer, (2018).
30. Shorten, C. & Khoshgofaar, T. M. A survey on image data augmentation for deep learning. *J. Big Data* **6**, 60 (2019).
31. Hui, J. GAN-Why It Is So Hard to Train Generative Adversarial Networks! (2018). <https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b>
32. Willner, A. Optical Fiber Telecommunications Vol. 11, Academic Press, London, (2019).
33. Xue, G. & Liu, S. A hybrid deep learning-based fruit classification using attention model and Convolution autoencoder. *Complex. Intell. Syst.* (2020).
34. Chen, J., Han, J., Liu, C., Wang, Y. & Shen, H. A Deep-Learning Method for the Classification of Apple Varieties Via Leaf Images from Different Growth Periods in Natural Environment 14, 1671 (Symmetry, 2022).
35. Duong, L. T., Nguyen, P. T., Di Sipio, C. D. & Ruscio, D. Automated fruit recognition using EfficientNet and MixNet. *Comput. Electron. Agric.* **171**, 105326 (2020).
36. Yu, F., Lu, T. & Han, B. A quantitative study of aggregation behaviour and integrity of spray-dried microcapsules using three deep convolutional neural networks with transfer learning. *J. Food Eng.* **300**, 110515 (2021).
37. Haghghi, S., Jasemi, M., Hessabi, S. & PyCM Multiclass confusion matrix library in python. *J. Open. Source Softw.* **3**, 729 (2018).
38. Bhargava, A. Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Univ. Sci.* (2018).
39. Koirala, A., Walsh, K. B. & Wang, Z. M. C. Deep learning–Method overview and review of use for fruit detection and yield Estimation. *Comput. Electron. Agric.* **162**, 219–234 (2019).
40. Zuo, F. et al. Multilevel fine-grained features-based General Framework for Object detection (IEEE transactions on cybernetics, 2024).
41. Lu, T., Han, B., Chen, L. & Yu, F. A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning. *Sci. Rep.* **11**, 15824 (2021).
42. Giri, R. N. et al. Enhanced hyperspectral image classification through pretrained CNN model for robust Spatial feature extraction. *J. Opt.* **53**(3), 2287–2300 (2024).
43. Lu, T., Yu, F., Xue, C. & Han, B. Identification, classification, and quantification of three physical mechanisms in oil-in-water emulsions using AlexNet with transfer learning. *J. Food Eng.* **288**, 110220 (2021).
44. Chen, J., Wang, H. & He, E. A transfer learning-based CNN deep learning model for unfavorable driving state recognition. *Cogn. Comput.* **16**(1), 121–130 (2024).
45. Molnar, C., Interpretable & Learning Machine Jan, (2023). <https://christophm.github.io/interpretable-ml-book/>, Accessed on 19.
46. Ribeiro, M. T., Singh, S. & Guestrin, C. Why Should I Trust You? Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. San Francisco, United States, 13–17 Aug 2016.
47. Lu, T. & Han, B. Detection and classification of marine mammal sounds using AlexNet with transfer learning. *Ecol. Inf.* **62**, 101277 (2021).
48. Brahimi, M. & Boukhalfa, K. Deep learning for tomato diseases: Classification and symptoms visualization. *Appl. Artif. Intell.* **31**, 1–17 (2017).
49. Zurowicz, M. An interactive visualization for feature localization in deep neural networks. *Front. Artif. Intell.* **3**, 49 (2020).
50. Mathworks (R2020b). <https://ww2.mathworks.cn/help/deeplearning>, Accessed on 19 Dec 2020.

## Author contributions

T. L. and F. Y. wrote the main manuscript text, Y. Y. and L. Z. modified the manuscript, T. L., F. Y. and Y. Y. prepared Figs. 1, 2, 3, 4, 5, 6, 7, 8 and 9; Tables 1, 2, 3, 4 and 5. All authors reviewed the manuscript.

## Funding

This work was supported by National Natural Science Foundation of China [Grant number 52205109], China Postdoctoral Science Foundation [2023M742156], key project funding of Key Lab of Industrial Fluid Energy Conservation and Pollution Control (Qingdao University of Technology), Ministry of Education and Laoshan Laboratory Science and Technology Innovation Project [2021WHZZB0304].

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to F.Y.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025