



A multivariate algorithm for identifying contaminated peanut using visible and near-infrared hyperspectral imaging



Zhen Guo ^{a,b,c}, Jing Zhang ^a, Jiashuai Sun ^{a,b,c}, Haowei Dong ^{a,b,c}, Jingcheng Huang ^{a,b,c}, Lingjun Geng ^{a,b,c}, Shiling Li ^{a,b,c}, Xiangzhu Jing ^{a,b,c}, Yemin Guo ^{a,b,c,*}, Xia Sun ^{a,b,c,**}

^a School of Agricultural Engineering and Food Science, Shandong University of Technology, No. 266 Xincun Xilu, Zibo, Shandong, 255049, China

^b Shandong Provincial Engineering Research Center of Vegetable Safety and Quality Traceability, No. 266 Xincun Xilu, Zibo, Shandong, 255049, China

^c Zibo City Key Laboratory of Agricultural Product Safety Traceability, No. 266 Xincun Xilu, Zibo, Shandong, 255049, China

ARTICLE INFO

Keywords:

Hyperspectral imaging
Convolutional neural network
Genetic algorithm
Uniform manifold approximation and projection
Peanut kernel

ABSTRACT

In this study, a novel uniform manifold approximation and projection combined-improved simultaneous optimization genetic algorithm-convolutional neural network (UMAP-ISOGA-CNN) algorithm was proposed. The improved simultaneous optimization genetic algorithm (ISOGA) combined with convolutional neural network (CNN) to optimize the architecture, hyperparameters, and optimizer of the CNN model simultaneously. Additionally, a uniform manifold approximation and projection (UMAP) method was used to visualize the feature space of all feature layers of the ISOGA-CNN model. The UMAP-ISOGA-CNN algorithm combined with visible and near-infrared hyperspectral imaging was used to identify peanut kernels contaminated with *Aspergillus flavus* and to distinguish their storage time, which is essential for the food industry to monitor the freshness of products. Overall, the UMAP-ISOGA-CNN algorithm provides useful insights into the feature space of the ISOGA-CNN model, contributing to a better understanding of the model's internal mechanisms. This study has practical implications for the food industry and future research on deep learning optimization.

1. Introduction

Peanut kernels are valuable sources of protein, fat, vitamins, and dietary fiber, which can be consumed directly or used to make peanut butter and edible oil [1]. However, these kernels are highly susceptible to contamination by toxicogenic fungi, with *Aspergillus flavus* and *Aspergillus parasiticus* being the most common culprits. These fungi produce aflatoxin B₁ (AFB₁), which is a highly toxic and carcinogenic substance [2]. Therefore, identifying and selecting any similarly contaminated peanut kernels is of utmost importance to prevent the entry of AFB₁ into the food chain [3]. By studying the changes in tissue structure and chemical composition of peanut kernels that have been contaminated with *Aspergillus flavus*, it is possible to identify differences between healthy kernels and contaminated kernels, facilitating the prompt detection of contaminated peanuts and enabling the management and control of peanut kernels to ensure their quality and safety.

Hyperspectral imaging technology is capable of acquiring both

spectral and spatial information of a sample, making it an effective tool for rapid and non-destructive detection of both external and internal quality of food and agricultural products [4]. The technology has been extensively employed in identifying fungal contamination in maize kernels and wheat grains [5,6]. For instance, the near-infrared hyperspectral imaging technique has been used to identify moldy peanut kernels, with an accuracy rate of 87.14% for the training images and 98.73% for the validation images [7]. Additionally, short-wave infrared (SWIR) hyperspectral imaging has been applied to identify prepared contaminated peanut kernels, with pixel-level classification accuracy rates of 99.13% for variety A, 96.72% for variety B, and 99.73% for variety C in the training images, and 96.32%, 94.2%, and 97.51% respectively in the validation images [3]. These studies demonstrate the feasibility of using hyperspectral imaging techniques to identify peanut kernels contaminated with *Aspergillus* fungi. Furthermore, hyperspectral imaging technology still faces some challenges. Firstly, for circular, spherical, or cylindrical food items, spectral variations occur due to the

* Corresponding author. School of Agricultural Engineering and Food Science, Shandong University of Technology, No. 266 Xincun Xilu, Zibo, Shandong, 255049, China.

** Corresponding author. School of Agricultural Engineering and Food Science, Shandong University of Technology, No. 266 Xincun Xilu, Zibo, Shandong, 255049, China.

E-mail addresses: gym@sdu.edu.cn (Y. Guo), sunxia2151@sina.com (X. Sun).

curvature of their surfaces. The light reflected from circular objects seen by the camera is not uniform, and the measured distance is actually shorter than the scattering distance [8]. Similarly, the ellipsoid shape of peanuts also causes certain spectral differences. Secondly, hyperspectral data contains a considerable amount of irrelevant information [9]. Reliable feature extraction and modeling methods are essential to deeply mine useful information from the hyperspectral data while avoiding excessive dimensionality reduction. During the model development process, it is crucial to experiment with various approaches and validate the model results to prevent overfitting or underfitting. In addition, convolutional neural network (CNN) is a widely used technique in deep learning (DL), which has achieved great success in image recognition and computer vision tasks. Moreover, DL represented by CNN combined with hyperspectral imaging has also been widely used in detecting fungal contamination in peanut kernels and corn kernels. For instance, a point-based CNN with embedded feature selection combined with near-infrared hyperspectral imaging achieves a recognition accuracy rate of 97.98% for moldy peanut kernels [10]. In addition, 1-dimensional-convolutional neural network (1D-CNN) combined with visible and near infrared hyperspectral imaging technology has been used for fungal contamination detection in corn kernels, and the 1D-CNN model based on the germ-up data set shows good classification results with a classification error rate of 3.15% [11].

In addition, the appropriate architecture of the CNN model is capable of extracting complex features from the training data effectively. However, designing an appropriate architecture requires domain knowledge and human expertise [12]. Selecting optimal values for parameters is crucial for achieving the best performance of a CNN, including the number of convolutional kernels, kernel size, pool size, and fully connected layer neurons. Finding the optimal configuration may be time-consuming. Moreover, determining when to stop the training is critical because stopping too early may result in under-learning, while stopping too late can lead to over-fitting [13]. The number of epochs and batch size are also essential factors that determine the performance of the CNN model. Genetic algorithm (GA) is a method of random global search and optimization that is developed by mimicking the evolutionary mechanisms of natural organisms, incorporating both Darwin's theory of evolution and Mendel's genetic theory [14]. Essentially, GA is a highly efficient, parallel, and global search technique that automatically acquire and accumulate knowledge of the search space throughout the search process. Additionally, it flexibly adopts the search process to obtain the optimal solution. The primary function of GA is as an efficient and effective means of conducting random global search and optimization [15]. Relevant studies have been applied GA to optimize CNN models. CNN has been used to classify pavement crack types, while GA is used to optimize the hyperparameters of the CNN. The optimized GA-CNN achieves an accuracy of 99.6% on the validation set [16]. In addition, GA is used to search for the best hyperparameters of fusion feature reduction-genetic algorithm-convolutional neural network-long short-term memory (FFR-GACLN), which is used to identify control chart patterns [17]. Moreover, GA also optimizes the hyperparameters of a hybrid convolutional neural network-bidirectional-long short-term memory (CNN-Bi-LSTM) model to predict the wind speed 24 h ahead [18]. GA has also been used to optimize CNN architecture for image classification, with the core encoding strategy using skip layers and pooling layers to construct the CNN model [19]. Additionally, a hybrid classification architecture composed of CNN and GA is used for image classification. The CNN feature layer is added, deleted, and modified in the GA to achieve variability in the CNN architecture [20]. However, no research has been found to apply GA to optimize the architecture, hyperparameters, and optimizer of CNN models simultaneously, and GA combined with CNN has not been utilized to identify peanut and other grain crops contaminated with *Aspergillus flavus*.

DL models are often considered as "black box" function approximators, meaning that they map inputs to classification outputs without providing any explanation for how they make their decisions. As a

result, developing high-quality DL models can be a time-consuming process that relies on trial-and-error methods [21]. Hence, techniques that enable a deeper understanding of the mechanism and performance of DL models have become a crucial component of robust verification procedures. Uniform manifold approximation and projection (UMAP) is a non-linear dimensionality reduction algorithm that uses manifold learning to map high-dimensional feature vectors to a low-dimensional space while preserving the local and global structures of the original data [22]. UMAP is particularly suitable for visualizing the complex structure of high-dimensional feature vectors in CNN models as 3D scatter plots, providing a better understanding of the model's feature extraction ability and internal workings. Compared to other manifold learning algorithms, such as t-SNE and PCA, UMAP has better scalability and interpretability, making it one of the preferred algorithms for exploring the feature extraction ability and internal workings of CNN models [23,24].

The purpose of this work is as follows: (1) Propose a novel improved simultaneous optimization genetic algorithm-convolutional neural network (ISOGA-CNN) algorithm that uses improved simultaneous optimization genetic algorithm (ISOGA) to optimize the architecture, hyperparameters, and optimizer of the CNN model simultaneously. (2) ISOGA-CNN combined with visible and near-infrared hyperspectral imaging is used to identify peanut kernels contaminated with *Aspergillus flavus* and to distinguish the storage time of contaminated peanut kernels. (3) To verify and to explore the robustness of the ISOGA-CNN algorithm and investigate the influence of architecture, hyperparameters and optimizer on the performance of the CNN model. (4) An innovative uniform manifold approximation and projection combined-improved simultaneous optimization genetic algorithm-convolutional neural network (UMAP-ISOGA-CNN) algorithm is proposed to visualize the feature space of all feature layers of the ISOGA-CNN model.

2. Materials and methods

2.1. Samples preparation

The variety of peanut kernel samples was "Weihua", which was purchased from a supermarket in Zibo, China. A total of 1260 peanut kernels were divided into the contaminated group and control group with a ratio of 2:1 based on the quantity proportion. All samples were soaked in a 75% ethanol solution for 1 min for sterilization, rinsed with sterile water 3 times, and placed in a sterile environment. *Aspergillus flavus* (ATCC#28539) was obtained from the National Strain Center of China, inoculated on PDA medium, and cultured for 5 d at 28 °C. Then, spores were collected and diluted to a concentration of 1×10^6 CFU/mL with sterile water to contaminate the peanut kernels. The peanut kernels were stored in an incubator until the 9th day, with the incubator was set to 30 °C and 85% relative humidity, respectively [25]. A total of 168 and 84 peanut kernels were selected from the contaminated group and control group, respectively, to collect hyperspectral images on the 1st, 3rd, 5th, 7th and 9th day.

2.2. Data acquisition

A visible and near-infrared hyperspectral imaging system (Isuzu Optics Corp., Taiwan, China) was used to obtain the hyperspectral images. It consists of a line-scan spectrograph (400–1000 nm), a CCD camera, 2 lamps and a light source, a mobile platform, and a computer used to process spectral data. During hyperspectral image collection, the exposure time was 9 ms, and the speed of the mobile platform was 1.32 mm/s. The black reference image and white reference image were used to calibrate the raw hyperspectral image to reduce the influence of ambient noise. The relevant correction formula was described by the previous study [26].

The hyperspectral imaging instrument operated in a controlled environment with constant temperature and humidity. Central air

conditioning was used to maintain these environmental conditions. Before collecting samples, they were placed in the same environment for 24 h. During the instrument's usage, the instrument was preheated for 30 min to ensure that it reached a stable state. All these standard procedures were implemented to minimize the impact of environmental and instrument-related factors on the data. Fig. S1 shows the hyperspectral image of peanut kernels.

Otsu's method was used to threshold the peanut kernels from the background, which was a popular algorithm used for image thresholding. It aims to automatically determine the optimal threshold value for image segmentation. ENVI 5.3 was used to select the ROI to extract the average reflectance of every single peanut kernel. Finally, a spectral matrix was obtained for further analysis. The organization of this matrix followed a pattern where the peanut samples (1260 peanut kernels) were represented by rows, the full spectral wavelengths (890 wavelengths) were represented by columns, and the average reflectance were represented internally.

2.3. Multiple modeling methods

2.3.1. Unsupervised classification methods

Principal component analysis (PCA) was used as an initial exploratory unsupervised learning technique to aid in visualizing the samples. The visible and near-infrared hyperspectral data matrix was reduced in dimensionality and important information was extracted using principal components (PC). By utilizing the extracted spectra, PCA provided valuable information about the similarity or dissimilarity between peanut kernel samples via a 3D projection of PCA scores [27]. Additionally, UMAP was a dimensionality reduction visualization method that employed manifold learning as its fundamental principle for dimensionality reduction operations. The primary objective of manifold learning was to identify clustering in the dataset while constraining the size of the local neighborhood. Therefore, data visualization had become a significant advantage of dimensionality reduction preserving both local and global relationships between data points. This characteristic made UMAP particularly useful as a preprocessing step for various clustering techniques [28]. The important parameters of UMAP included the number of neighbors (`n_neighbors`) and minimum distance (`min_dist`), which were intuitive in balancing the preservation of local and global structures [29]. Due to the large amount of spectral data, that the data was projected onto a 2D space would result in a relatively dense data distribution, which would be difficult to effectively distinguish. Therefore, a 3D space PCA and UMAP were employed.

2.3.2. Linear classification methods

Linear discriminant analysis (LDA) was a commonly used linear classification algorithm, aiming at maximizing the separability between classes by projecting the data onto a low-dimensional space [30]. Principal component analysis-linear discriminant analysis (PCA-LDA) was a classification algorithm that used both PCA and LDA, where the data was 1st projected onto a low-dimensional space using PCA, and then classified using LDA on the projected data [31]. Partial least squares-discriminant analysis (PLS-DA) was a commonly used multivariate statistical analysis method used to establish the relationship between input and output variables, mainly applied to classification problems. PLS-DA used partial least squares regression analysis to establish the relationship between input variables and response variables, and LDA was used for classification [32]. The accuracy rate was used to evaluate the classification effect of each model.

2.4. ISOGA-CNN

2.4.1. CNN

In this study, a 1D-CNN was utilized as the classification model. Compared to 2 dimensional-convolutional neural network (2D-CNN), 1D-CNN had the advantages of faster training speed, stronger

interpretability, and was more suitable for 1 dimensional data such as spectral data. CNN was a type of feed-forward artificial neural network consisting of 3 primary components: convolutional layers, pooling layers, and fully connected layers [33]. Each convolution layer contained multiple convolution kernels that were used to extract features from the data. Following the convolution operation, pooling layers were employed to reduce the feature dimension [18]. The fully connected layer served as the classifier within the CNN. In addition, to avoid over fitting, the dropout layer was incorporated into the neural network to halt specific neurons during forward propagation with a certain probability [16]. Rectified linear units (ReLU) were used as activation function units to address the issue of vanishing gradient and expedite the training process in all the convolutional layers and fully connected layers, except for the last fully connected layer which used the "softmax" function as its activation function [34]. As shown in Fig. S2, the initial CNN model has 4 convolutional layers, 2 max pooling layers, a dropout layer, and 2 fully connected layers. The initial CNN model had sufficient depth to extract high-level features from the data [35].

2.4.2. GA

In the GA, the initial step involved generating a random population of potential solutions to the problem. Each solution was evaluated based on its fitness, which was a measure of its effectiveness in solving the problem. The fittest individuals from the population were selected as parents for the next generation, typically through methods such as roulette wheel or tournament selection. The genetic material of 2 parent individuals was combined to create new children through random selection of a crossover point and exchange of genetic material. To promote diversity and prevent convergence to a local optimum, random changes were introduced to the genetic material of some individuals in the population. The least fit individuals were then replaced with new children form the next generation. This process was repeated until a satisfactory solution was found or a termination criterion was met.

2.4.3. ISOGA

The purpose of the study was to optimize a CNN model using the ISOGA in order to classify contaminated peanut kernels. The ISOGA aimed to find the optimal architecture, hyperparameters, and optimizer for the CNN model simultaneously, using the validation accuracy as the fitness function. Each CNN model was treated as an individual participating in the evolutionary process, and its architecture, hyperparameters and optimizer were encoded as a variable-length chromosome. Each chromosome consisted of an input block, multiple middle blocks, an output block, and a hyperparameter block, with middle blocks that could be stacked to form a variable-length block. The input block contained 2 convolutional layers and a max pooling layer, while the output block consisted of a dropout layer for regularization and 2 fully connected layers. The hyperparameter block contained the hyperparameters for model compilation and training, including batch size, epochs, optimizer, and learning rate. 7 optimizers were encoded as potential genes to enhance the population's diversity, including stochastic gradient descent (SGD), adaptive moment estimation (Adam), adaptive gradient algorithm (Adagrad), adaptive delta (Adadelta), adaptive moment estimation with infinity norm regularization (Adamax), nesterov-accelerated adaptive moment estimation (Nadam), and root mean square propagation (RMSprop). Fig. 1 shows the process of optimizing a CNN model using the ISOGA, which mainly includes generating an initial population, selecting, crossing over, mutating, and finally generating offspring.

The optimization process was as follows: the initial CNN model was used as the parent and its architecture, hyperparameters and optimizer were strongly mutated to generate an initial population. In each evolutionary generation, new offspring were produced and evaluated for their fitness, with lower-fitness individuals being replaced by those from the new generation. Selection of offspring was done using elite, tournament, and roulette wheel algorithms to increase population diversity

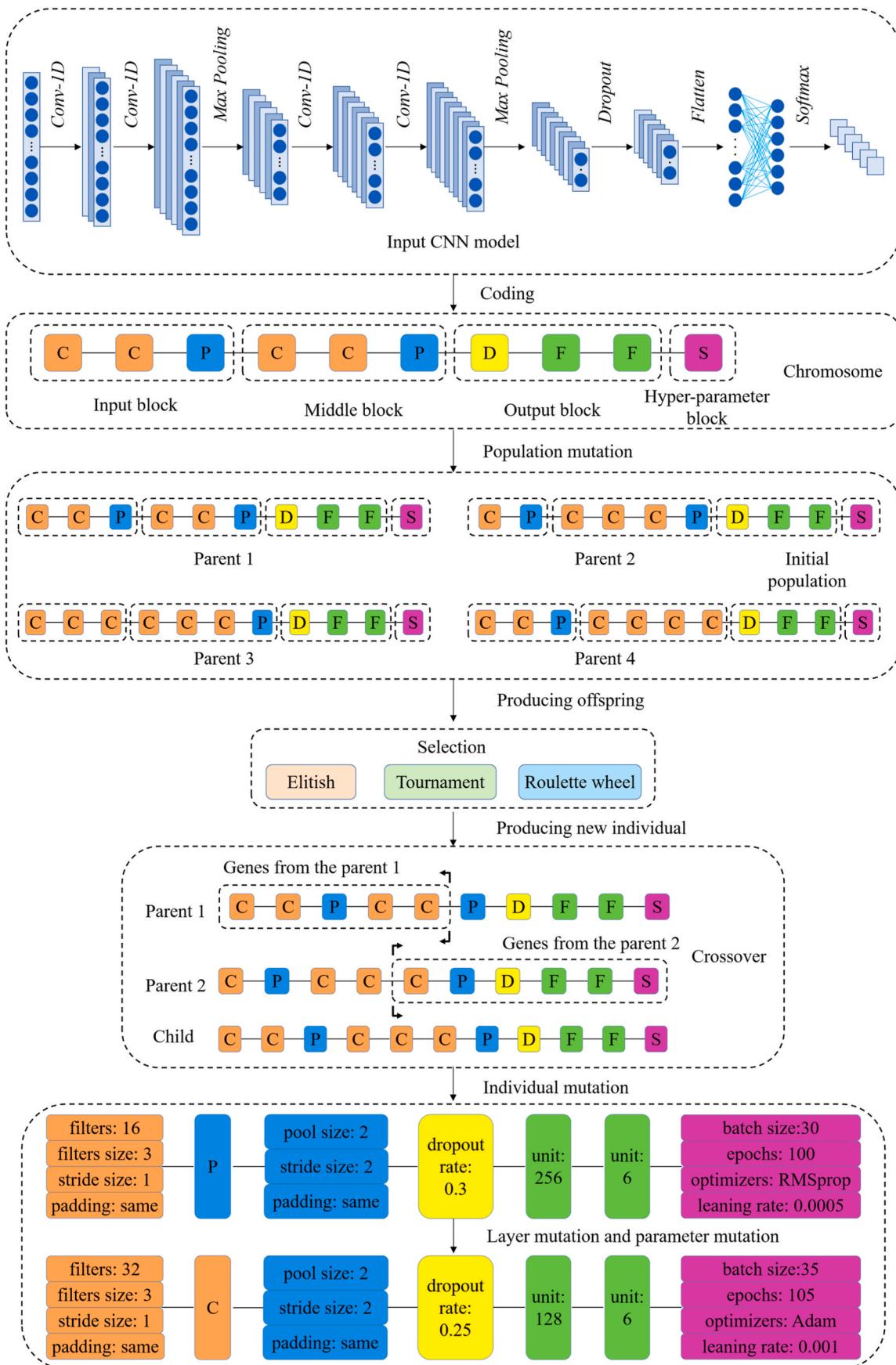


Fig. 1. Process of optimizing the initial CNN model using ISOGA.

and avoid getting stuck in local optima. Offspring was produced using the single-point crossover technique, which randomly split the coding block of each parent into 2 parts and combined them. 2 types of mutations were used, including population mutation and individual mutation. The former was used to generate the initial population and it involved block mutation, layer mutation, and parameter mutation in that order. And the latter was achieved through layer mutation and parameter mutation. Block mutation using for middle blocks randomly selected a block from an individual's coding and randomly decided to add a new layer or delete an existing one. This changed the structure of the middle blocks, allowing the CNN architecture to adjust its depth according to the problem. Layer mutation randomly selected a block from an individual's coding and mutated it based on its type, except the output block and hyperparameter block. For instance, if it was a convolutional layer, an operator divided the number of feature maps in that layer by 2 and created a new layer through deep copying. This new layer was then placed after the selected layer. And if it was a pooling layer, the operator replaced it with a new convolutional layer. Parameter mutation changed the parameter values of different layers, including all of blocks. During each iteration of generating new offspring, the operator examined a layer and made a random decision on whether to apply a mutation. When it encountered a convolutional layer, the operator could double or halve the number of convolution kernels. When the operator encountered a hyperparameter block, it could modify the values of batch size and number of epochs by adding or subtracting 5. Additionally, the optimizer might change from RMSprop to Adam, and the learning rate values of the optimizer could be doubled or halved.

2.5. Other CNN models

The multi-head attention-convolutional neural network (A-CNN) utilized a multi-head attention mechanism to learn feature representations at different levels by adding attention mechanisms to various convolutional layers. In each convolutional layer, a convolution operation was performed first, followed by a fully connected layer that generated attention weights on the output feature map. Finally, the attention weights were used to weight the output feature map of the convolutional layer. By combining multiple convolutional operations and attention mechanisms, the network could learn more discriminative feature representations. The convolutional neural network-long short-term memory (CNN-LSTM) was a neural network architecture that combined the strengths of CNN and long short-term memory (LSTM). The CNN served as a feature extractor in the CNN-LSTM, extracting meaningful features from the input data. The output of the CNN was then input to the LSTM network, which learned temporal dependencies in the data and made predictions based on those dependencies. The multi-head attention-convolutional neural network-long short-term memory (A-CNN-LSTM) was a method that combined A-CNN and CNN-

LSTM. All algorithms were implemented based on Python 3.7.8.

3. Results and discussion

3.1. Spectral analysis

In this study, Fig. 2(a) shows that 1260 spectral curves are obtained including 840 contaminated samples and 420 control samples. The spectral curves were found to overlap and could not be differentiated in the 400–600 nm range. Furthermore, the lower reflectance values of the spectral curves in the 400–600 nm range could be attributed to the light absorption of pigment components such as anthocyanins and chlorophyll in peanut skin [36]. The average spectral curves of contaminated group and control group show significant spectral differences in the 600–1000 nm range in Fig. 2(b), and the reflectance of the contaminated group exhibit significantly lower reflectance than that of the control group. Additionally, Fig. 2(c) shows that the original reflectance spectra of the control group and contaminated groups with different storage times exhibit a similar trend of changes in the 400–1000 nm wavelength range. No obvious reflectance peak was observed on the smooth spectral curve, and the reflectance intensity of the peanut kernels decreased with an increase in the fungal spore count and storage days which was similar to previous reports [37]. This phenomenon might have been related to the metabolic activity of *Aspergillus flavus* during storage, which caused a decrease in moisture and nutrients such as protein, and starch, resulting in a loss of luster on the surface of peanuts, darkening of color, and a decrease in spectral reflectance [38]. Moreover, the spectral difference between the control group and contaminated groups was small in a short storage time (day 1, day 3), suggesting it was challenging to distinguish the control kernels and contaminated kernels visually.

3.2. Unsupervised classification methods

PCA and UMAP were employed for exploratory data analysis. The n_components parameter for both PCA and UMAP had been set to 3. The n_neighbors and min_dist parameters for UMAP had been set to 15 and 0.1, respectively. Firstly, PCA was utilized to qualitatively analyze the spectral data from the control group and contaminated groups. The obtained results illustrate that the control group is distinctly separable from the contaminated group as evidenced in Fig. 3(a). However, Fig. 3(c) shows that the 5 contaminated groups are not clearly separated, and the day 1 group cannot be effectively distinguished from the control group. UMAP was then implemented to reduce the dimensionality of the spectral data using unsupervised learning. Fig. 3(b) shows that the spectral data from the control group and contaminated group present an obvious separation phenomenon after UMAP reduction, although some data are still challenging to differentiate. Moreover, Fig. 3(d) shows that the day 1 group and control group have a small local distance, leading to

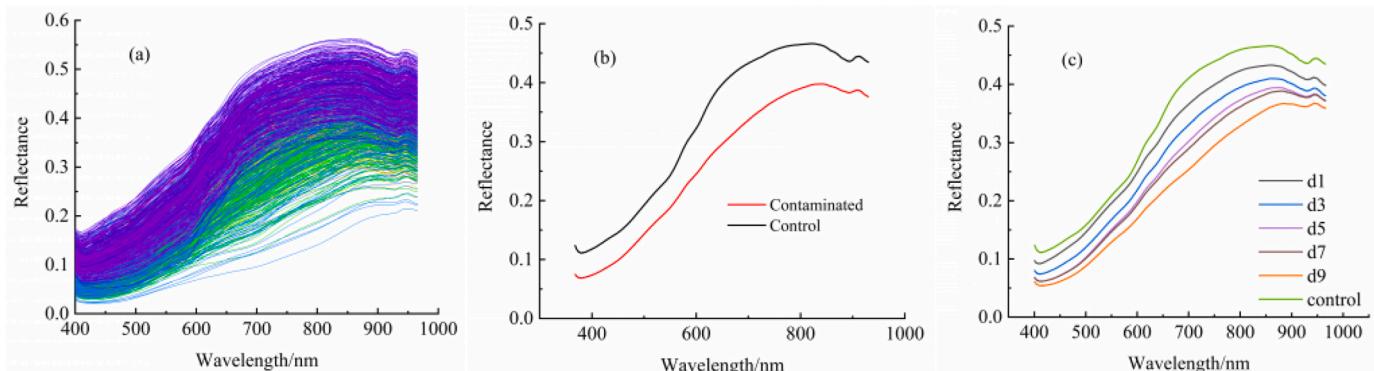


Fig. 2. Spectral curves of peanut kernels, (a) spectral curves of 1260 peanut kernels, (b) spectral curves of the contaminated group and the control group, (c) spectral curves of the contaminated group from day 1 (d1) to 9 (d9), as well as the control group.

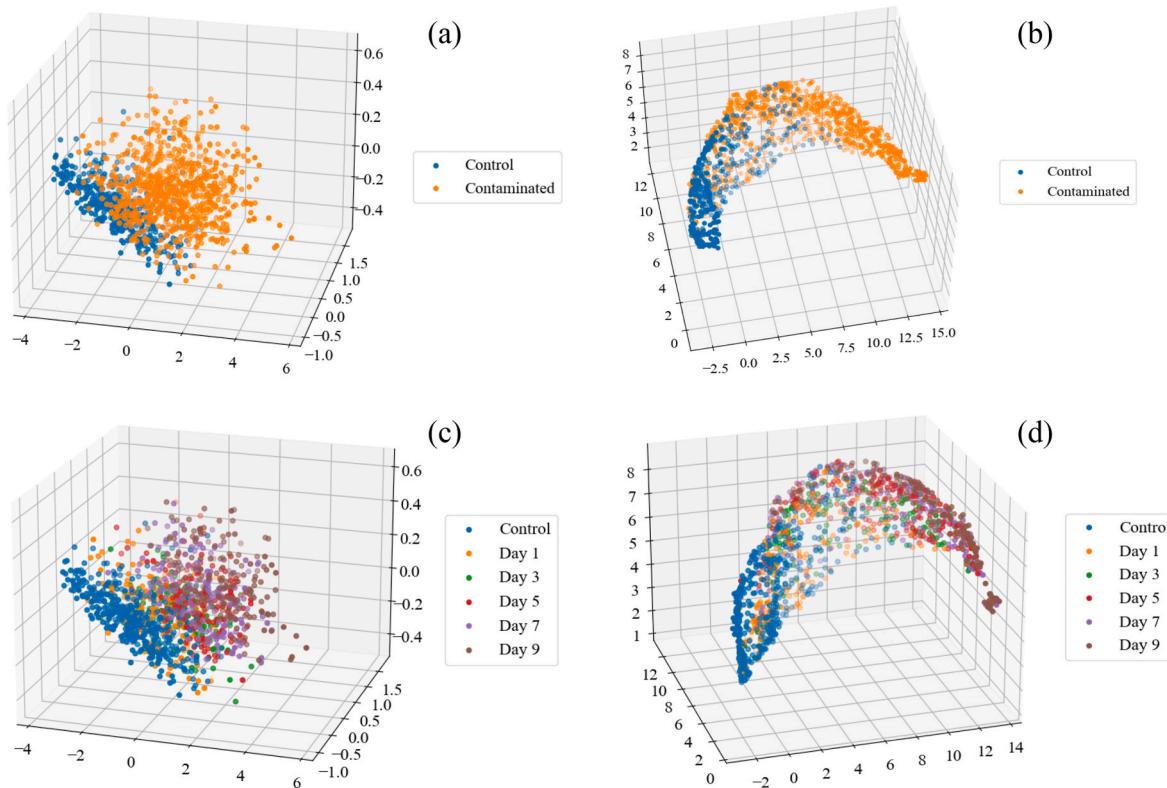


Fig. 3. Dimensionality reduction performance of PCA and UMAP, (a) PCA results of the control group and the contaminated group data, (b) UMAP results of the control group and the contaminated group data, (c) PCA results of the control group data and the contaminated group data from day 1–9, (d) UMAP results of the control group data and the contaminated group data from day 1–9.

difficulties in distinguishing them. Although both PCA and UMAP were unable to distinguish the 5 contaminated groups effectively, the UMAP algorithm was able to capture local relationships while preserving global distance. Therefore, it was utilized to visualize the feature layer of the CNN model in subsequent discussions, enhancing the CNN's interpretability.

3.3. Model performance based on different classification methods

3.3.1. Initial CNN model

The samples are classified using the initial CNN model with the architecture shown in Fig. S2. The sample set was divided into a training set and a test set in a quantity 3:1 ratio, and 5-fold cross-validation was used. The batch size and epochs were set to 30 and 100, respectively, and the model was trained using RMSprop optimizer with the learning rate of 0.0005. At first, the samples from the control group and contaminated groups are classified into 2 categories, as illustrated in Fig. 4(a) and (b). The training set and validation set exhibited classification accuracies of 96.16% and 95.77%, respectively, and the corresponding loss rates were 0.1035 and 0.1195, respectively. The model started to converge at the 95th epoch. On the test set, the model achieves a classification accuracy of 92.38% in Fig. 4(e). The CNN model performed well in classifying contaminated samples of the control group and contaminated group. Subsequently, the control group and contaminated groups with different storage times were classified into 6 categories. The classification accuracies of the training set and validation set are 64.15% and 58.73%, respectively, and the corresponding loss rates are 0.8513 and 0.9226, respectively, during 100 epochs, as observed in Fig. 4(c) and (d). The accuracy and loss curves fluctuated within the first 100 epochs, and the CNN model did not converge, indicating a poor fitting effect that could be attributed to the inappropriate design of the model architecture or hyperparameters settings. Fig. 4(f) presents the confusion matrix of the 6-class classification effect, with an overall

accuracy of 70.16%. The initial CNN model could not accurately classify contaminated peanut samples with different storage times. Therefore, optimization of the CNN model's architecture, hyperparameters and optimizer was needed to achieve better classification results.

3.3.2. ISOGA-CNN

The initial CNN model was optimized using ISOGA, the number of the population sizes and generations was set to 12 and 8, respectively. 3 offspring were produced per generation to ensure genetic diversity in the population and to find the individual with the highest fitness in a short period of time. Since the initial CNN model had good classification performance for identifying the contaminated samples in the control group and contaminated group, the ISOGA-CNN was emphatically used to classify the control group and contaminated groups with different storage times. Table 1 shows the fitness of individuals in the initial population and final population in ISOGA-CNN, which each individual is a CNN model. In the initial population, individual 1 was the initial CNN model, with a fitness equal to its validation accuracy of 0.5873. Individual 9 had the highest fitness of 0.6085, while individual 11 had the lowest fitness of 0.3333, possibly due to hyperparameter mutation, which caused its learning rate to be high. The large amplitude of parameter updates during training made the model unable to converge. In the final population, individuals 3, 8, and 10 had the highest fitness, all reaching 1.0000, indicating that their validation accuracy reached 100%. Compared to the average fitness of individuals in the initial population (0.5048), the average fitness of all individuals in the final population was significantly improved to 0.8399, demonstrating the clear optimization effect of the ISOGA on the CNN model. Table 2 shows the fitness of offspring produced in each generation during the ISOGA-CNN, and 24 offspring are produced in total. The fitness of the 2nd offspring in the 2nd generation, the 3rd offspring in the 4th generation, and the 2nd offspring in the 7th generation all reached 1.0000, while 5 offspring had low fitness of 0.3333. In addition, an individual with a

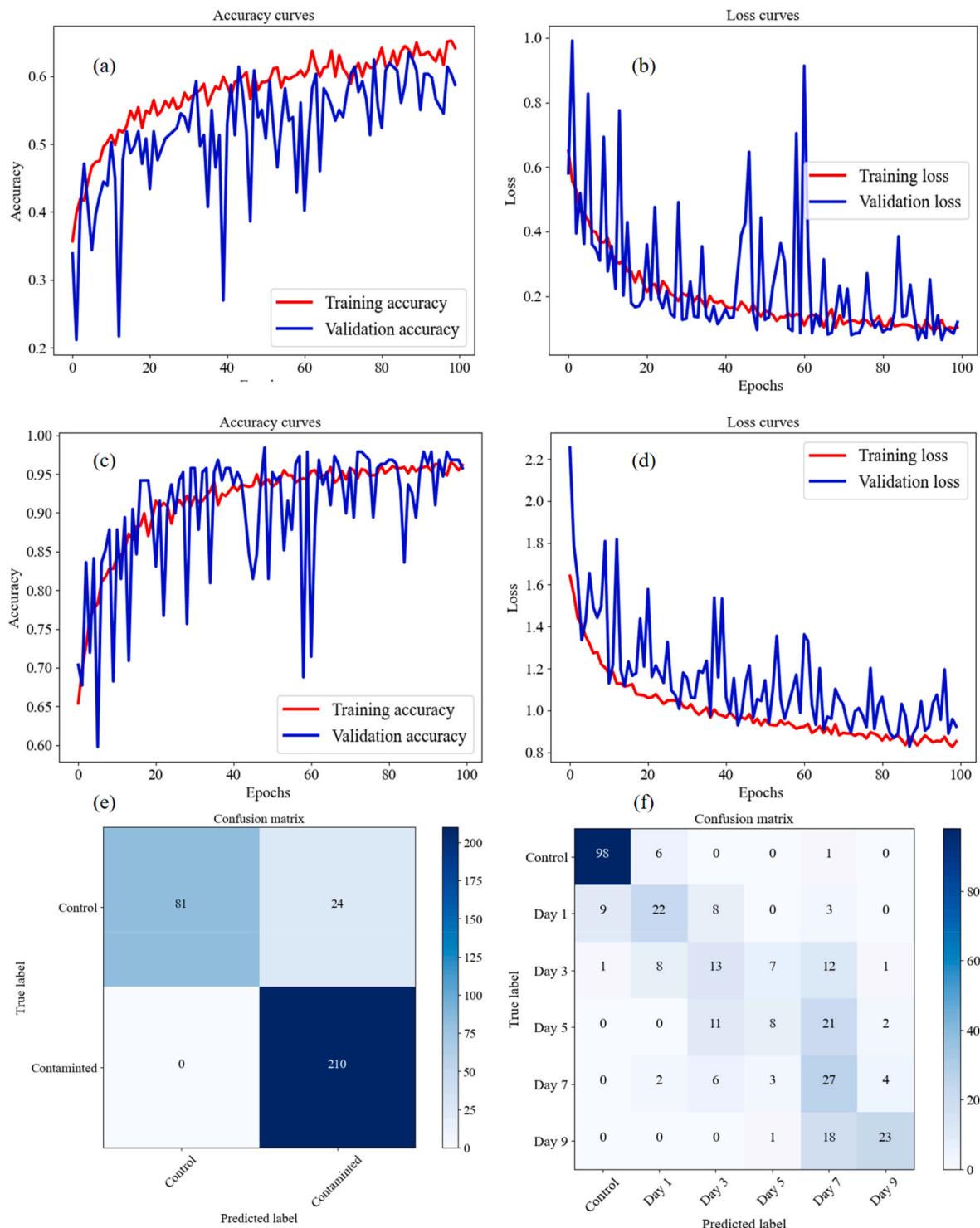


Fig. 4. Performance of the initial CNN model, (a) accuracy curves for the 2-class classification, (b) loss curves for the 2-class classification, (c) accuracy curves for the 6-class classification, (d) loss curves for the 6-class classification, (e) confusion matrix for the 2-class classification, (f) confusion matrix for the 6-class classification.

Table 1

Fitness of 12 individuals in the initial population and the final population of the ISOGA-CNN algorithm.

Individual	1	2	3	4	5	6	7	8	9	10	11	12
Initial population	0.5873	0.5661	0.5291	0.5079	0.4762	0.4180	0.5026	0.3810	0.6085	0.5979	0.3333	0.5502
Final population	0.9894	0.6349	1.0000	0.9627	0.9735	0.6243	0.6720	1.0000	0.6085	1.0000	0.9894	0.6243

Table 2

Fitness of 24 new individuals generated across 8 generations in the ISOGA-CNN algorithm.

Generation	Offspring 1	Offspring 2	Offspring 3
1	0.4762	1.0000	0.6243
2	0.3333	0.9894	0.4603
3	0.5926	0.4074	0.6720
4	0.3333	0.9630	1.0000
5	0.6031	0.3333	0.3333
6	0.4921	0.6349	0.9894
7	0.9735	1.0000	0.6243
8	0.3333	0.5026	0.5767

fitness of 1.0000 appeared in the 1st generation, indicating that the ISOGA-CNN had a fast convergence speed. All results showed that the ISOGA could globally search for the optimal solution in the solution space and effectively optimize the architecture and hyperparameters of the CNN model.

The optimal individual in the 2nd offspring of the 7th generation is selected as the ISOGA-CNN model, whose architecture is shown in [Table 3](#). Compared to the initial CNN model, the ISOGA-CNN model increased the number of convolutional layers from 4 to 8, indicating a significant improvement in its ability to extract deep features from data. A dropout layer was added between the 5th and 6th convolutional layers to further improve the model's generalization ability. In addition, the optimizer for the ISOGA-CNN model was RMSprop, the learning rate was 0.0005, the batch size was 25, and the epochs were 100. [Fig. 5\(a\)](#) and (b) show that the ISOGA-CNN model's classification accuracy on the training set and validation set both reaches 100.00%, and the loss rates are 0.0002 and 0.0020, respectively. The accuracy rates and loss rates fluctuated very little between the 80th and 100th epoch, indicating that the model had completed training and fully converged within 100 epochs. [Fig. 5\(c\)](#) shows that the ISOGA-CNN model's classification accuracy on the test set is also 100%, indicating that the model achieves accurate classification between the control group and contaminated group and accurate discrimination of the contaminated samples with different storage times.

In addition, ISOGA-CNN also demonstrated good classification performance for the control group and contaminated group, achieving classification accuracies of 99.60% and 100.00% on the training set and validation sets, respectively, and an accuracy of 99.37% on the test set. The corresponding data are presented in the supplementary information ([Fig. S3](#), [Table S1](#), [Table S2](#) and [Table S3](#)).

3.3.3. Other classification methods

ISOGA-CNN, CNN, A-CNN, CNN-LSTM, A-CNN-LSTM, LDA, PCA-LDA, and PLS-DA were used to identify contaminated peanut kernels, 2-class classifications and 6-class classifications were conducted for each

model. The classification results of all models are displayed in [Table 4](#) and [Table 5](#). The architectures of A-CNN, CNN-LSTM, and A-CNN-LSTM were similar to the initial CNN model, A-CNN added multi-head attention mechanisms to each convolutional layer, while CNN-LSTM replaced the 2 fully connected layers with LSTM layers. In A-CNN, there were 2 attention heads, and in CNN-LSTM, the 1st LSTM layer had 256 memory units, while the 2nd LSTM layer had 6 memory units with "softmax" function outputting classes. LDA used linear classification, PCA-LDA, and PLS-DA all had 20 PC. In the 2-class classification of the control group and contaminated group, ISOGA-CNN achieved the best classification performance on the test set with an accuracy of 99.37%, precision of 99.05%, recall of 99.05% and an F1-score of 99.05%. A-CNN-LSTM followed closely with an accuracy of 98.73%, precision of 96.33%, recall of 100.00% and an F1-score of 98.13%. Compared to the initial CNN model, A-CNN-LSTM showed an improvement in classification performance. In the 6-class classification of the control group and contaminated groups with different storage times, ISOGA-CNN achieved the best classification performance on the test set with an accuracy of 100.00%, precision of 100.00%, recall of 100.00% and an F1-score of 100.00%. The next best model was PLS-DA, with an accuracy of 78.10%, precision of 84.62%, recall of 82.92% and an F1-score of 83.43%. This suggested that the ISOGA-CNN model had better generalization ability than linear models and other CNN models.

3.4. Robustness of ISOGA-CNN

The ISOGA-CNN algorithm was run independently for 100 times, and the running results of the algorithm were recorded to explore its robustness and to explain the behavior and performance of the CNN model. [Fig. S4](#) shows the box plot of the fitness of the final population selected by the algorithm after running ISOGA-CNN 100 times independently. The individuals in the final population output by the algorithm had been sorted and renamed based on their fitness, and hence, the 100 individuals in individual 1 were the best models selected by the algorithm. 46 individuals have a fitness of over 0.9800, which indicated that the corresponding ISOGA-CNN models had converged, and the ISOGA had found the global optimal solution. Meanwhile, 59 individuals have a fitness of over 0.9000, which indicated that the 13 individuals with fitness values ranging from 0.9000 to 0.9800 had not yet fully converged, which might be due to the small number of epochs. However, the lowest fitness was 0.6243, which could represent some unconverged models. Overall, these data demonstrated that the ISOGA-CNN algorithm had good robustness, but it should be noted that the algorithm's optimal solution was not unique.

[Table S4](#) presents the architectures of 46 ISOGA-CNN models that have a fitness exceeding 0.9800. The architecture of output block remained unchanged as it did not undergo block mutation and layer mutation, comprising a dropout layer, a flatten layer, and 2 fully

Table 3

Architecture of the ISOGA-CNN model.

No.	Layer	Kernel size	Stride	Filter	Output shape	Activation function	Padding
1	Input				(890,1)		
2	Conv1D 1	(3,1)	1	32	(890,32)	ReLU	same
3	Conv1D 2	(3,1)	1	256	(890,256)	ReLU	same
4	Average pooling1D 1	(2,1)	2		(445,256)	ReLU	same
5	Conv1D 3	(3,1)	1	64	(443,64)	ReLU	valid
6	Conv1D 4	(3,1)	1	256	(443,256)	ReLU	same
7	Conv1D 5	(3,1)	1	256	(443,256)	ReLU	same
8	Dropout 1				(443,256)		
9	Conv1D 6	(3,1)	1	8	(441,8)	ReLU	valid
10	Conv1D 7	(3,1)	1	16	(439,16)	ReLU	valid
11	Conv1D 8	(3,1)	1	32	(437,32)	ReLU	valid
12	Dropout 2				(437,32)		
13	Flatten				(13,984,1)		
14	Fully connected 1				(128,1)	ReLU	
15	Fully connected 2				(6,1)	Softmax	

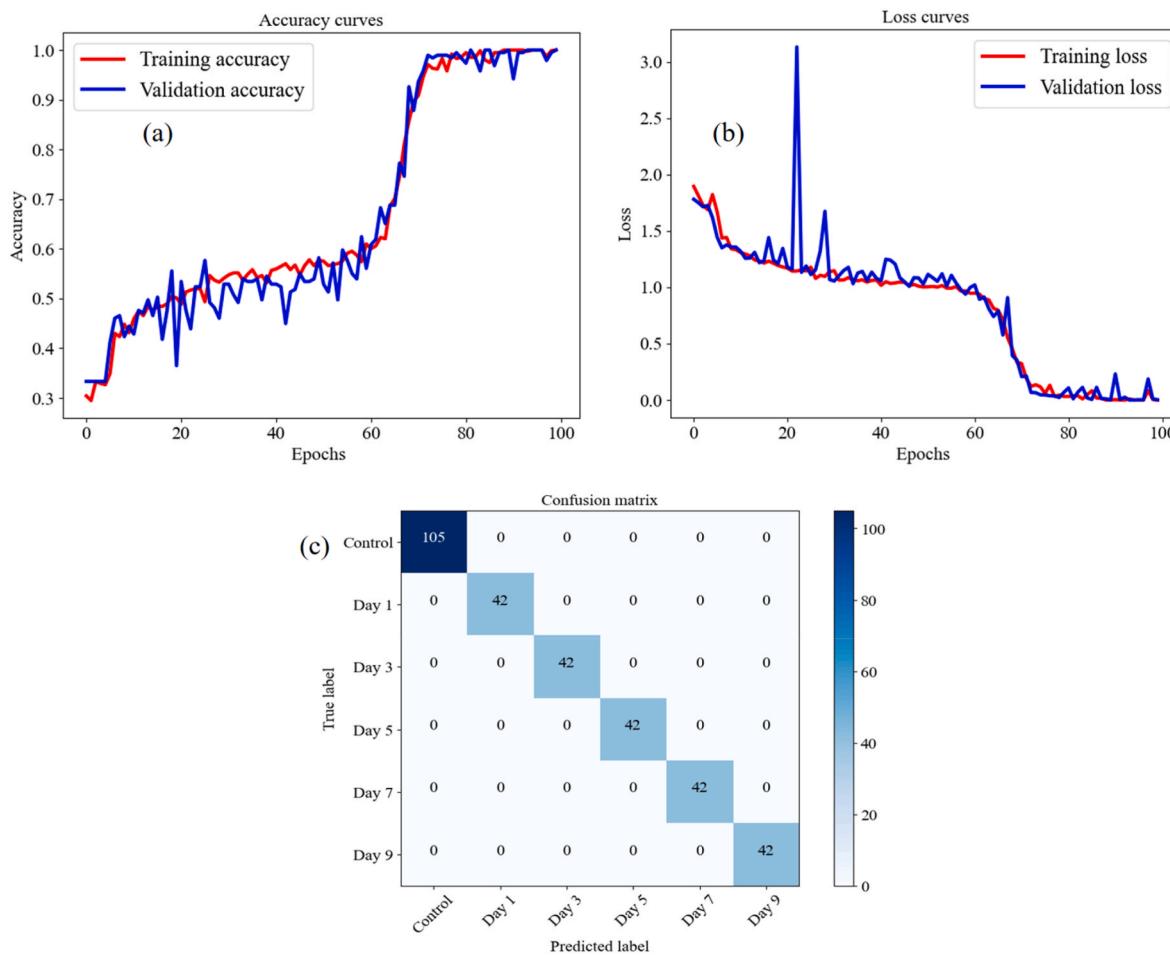


Fig. 5. Performance of the ISOGA-CNN, (a) accuracy curves for the 6-class classification of the ISOGA-CNN, (b) loss curves for the 6-class classification of the ISOGA-CNN, (c) confusion matrix for the 6-class classification of the ISOGA-CNN.

Table 4
Quantitative results of different models in identifying contamination peanut kernels.

Methods	Training set accuracy	Validation set accuracy	Test set accuracy	Precision	Recall	F1-score
ISOGA-CNN	99.60	100.00	99.37	99.05	99.05	99.05
CNN	96.16	95.77	92.38	100.00	77.14	87.10
A-CNN	97.75	99.47	98.41	96.30	99.05	97.65
CNN-LSTM	96.03	96.83	95.23	95.92	89.52	92.61
A-CNN-LSTM	97.62	98.41	98.73	96.33	100.00	98.13
LDA	100.00	16.94	69.52	25.00	64.29	36.00
PCA-LDA	92.13	91.15	86.03	47.50	45.24	46.34
PLS-DA	98.56	97.16	88.57	62.79	57.45	60.00

Table 5
Quantitative results of different models in distinguishing contamination peanut kernels storage time.

Methods	Training set accuracy	Validation set accuracy	Test set accuracy	Precision	Recall	F1-score
ISOGA-CNN	100.00	100.00	100.00	100.00	100.00	100.00
CNN	64.15	58.73	60.63	55.76	52.46	52.20
A-CNN	66.93	64.55	61.27	55.37	53.73	54.13
CNN-LSTM	59.92	60.85	61.90	58.18	54.52	55.34
A-CNN-LSTM	61.90	63.49	60.32	52.75	50.63	49.53
LDA	100.00	33.11	47.30	46.54	47.99	46.35
PCA-LDA	75.30	73.55	67.62	63.66	62.70	62.79
PLS-DA	87.32	85.90	78.10	84.62	82.92	83.43

connected layers. Therefore, Table 4 only shows the architecture of the input block and middle blocks. The results showed that the models contained a considerable number of convolutional layers, with the 99th

model having 17 convolutional layers, and on average, each model had 9.63 convolutional layers. Typically, increasing the number of convolutional layers enhanced the model's expressive power as deeper

convolutional layers could capture more features and patterns. The use of pooling layers ("M" or "A") was also widespread, and nearly all models had at least 1 pooling layer. Among them, the max pooling layer appeared 25 times, while the average pooling layer appeared 9 times, indicating that the former was more popular, likely because it preserved the essential features of an image better. Moreover, the ISOGA-CNN architectures were diverse, which made them appropriate for seeking global optimal solutions when the optimal solution was not fixed.

Fig. S5 shows the frequency distribution histograms of the optimizer, learning rate, batch size, and epochs selected by the algorithm after running ISOGA-CNN independently 100 times. Fig. S5(a) shows that RMSprop is the most commonly used optimizer, appearing 53 times. Adam and Nadam were the next most commonly used, while Adamax was used less frequently. The frequency of SGD, Adagrad, and Adadelta

was 0. In general, Adamax, Nadam, and RMSprop were more advanced and complex optimization algorithms that performed better on a wide range of problems than SGD, Adagrad, and Adadelta. They were designed to dynamically adjust the learning rate and momentum during the training process to achieve better convergence and optimization performance. Fig. S5(b) shows that the learning rate of 0.00005 is the most frequently used, appearing 32 times, followed by 0.001, 0.00025, 0.00125, and 0.0025, which are also frequently used. The frequency of other learning rates was relatively low. This suggested that a relatively small learning rate might be more suitable for training CNN models, as a learning rate that was too large may cause the model to fail to converge, while a learning rate that was too small may slow down convergence. Fig. S5(c) shows that the batch size of 25 is the most frequently used, appearing 35 times, followed by 15 and 35. Generally, a larger batch size

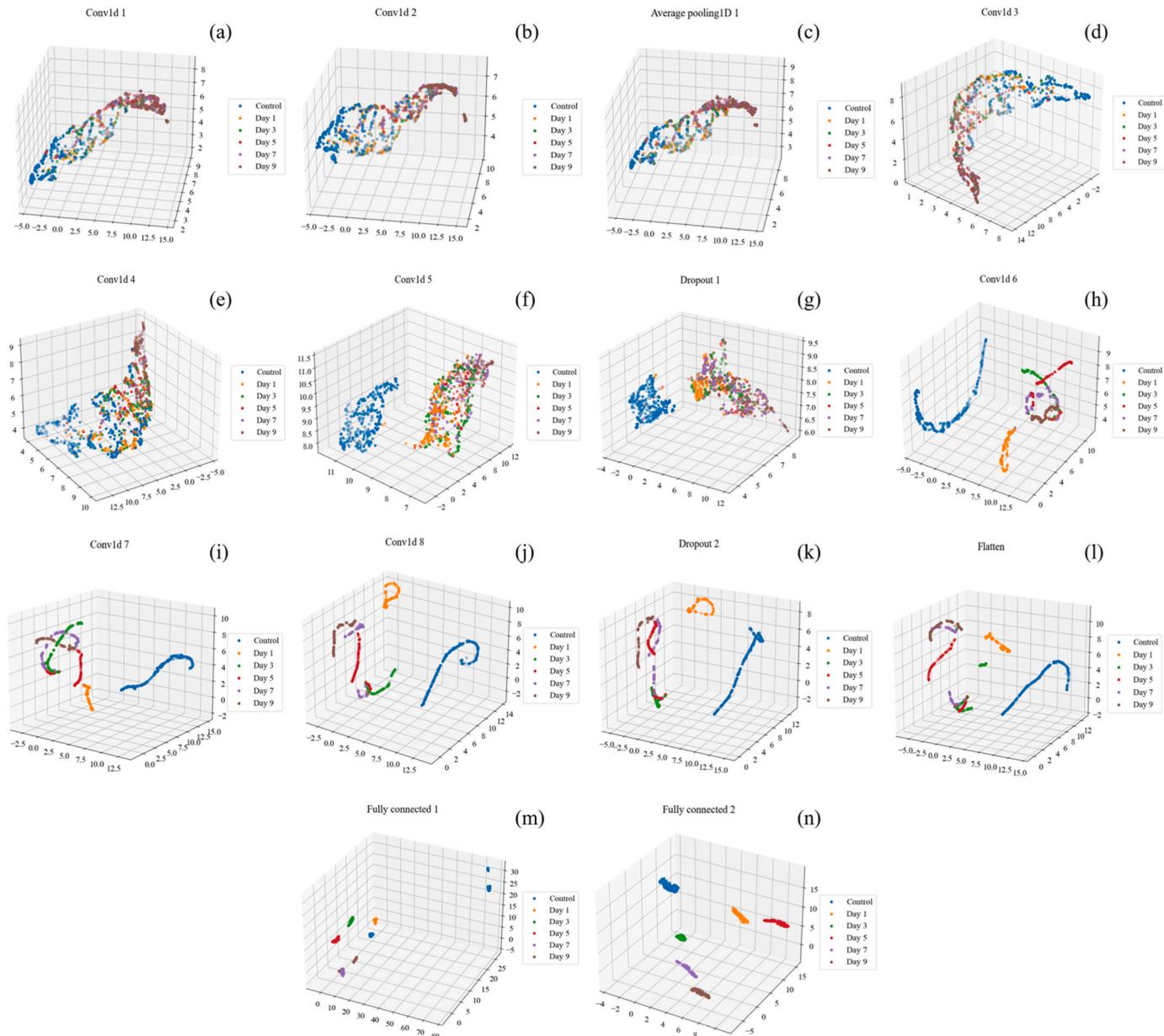


Fig. 6. Process of UMAP visualizing the ISOGA-CNN feature layers, (a) the features extracted by the 1st convolutional layer, (b) the features extracted by the 2nd convolutional layer, (c) the features extracted by the average pooling layer, (d) the features extracted by the 3rd convolutional layer, (e) the features extracted by the 4th convolutional layer, (f) the features extracted by the 5th convolutional layer, (g) the features extracted by the 1st dropout layer, (h) the features extracted by the 6th convolutional layer, (i) the features extracted by the 7th convolutional layer, (j) the features extracted by the 8th convolutional layer, (k) the features extracted by the 2nd dropout layer, (l) the features extracted by the flatten layer, (m) the features extracted by the 1st fully connected layer, (n) the features extracted by the 2nd fully connected layer.

can speed up training, but may lead to overfitting, while a smaller batch size may cause the model to be unstable. Fig. S5(d) shows that the epochs of 100 is the most frequently used, appearing 37 times, followed by 95, with the frequency of other epochs relatively low. In summary, the model with the optimizer of RMSprop, the initial learning rate of 0.0005, the batch size of 25, and the epochs of 100 had a relatively high fitness in the ISOGA-CNN, indicating its relatively high classification accuracy. From the data distribution, the skewness and kurtosis of the learning rate data set were 1.90 and 4.26, respectively, indicating a positively skewed distribution. The skewness and kurtosis of the batch size data were 0.14 and -0.34, respectively, with no obvious skewness. The skewness and kurtosis of the epochs data were 0.25 and 0.06, respectively, with a normal distribution of data. The initial hyperparameters of the input may had an impact on the statistical results, leading to results that did not conform to a normal distribution. This might be because different initial hyperparameters determined different states of the initial population, which might lead to different results in the iteration of the ISOGA-CNN. From the statistical results provided, it appeared that the initial learning rate of the input CNN model was not the optimal initial parameter. This suggested that using a ISOGA to optimize the hyperparameters of CNN models might result in finding the potential optimal solution.

3.5. UMAP-ISOGA-CNN

UMAP was a manifold learning algorithm that could map high-dimensional data to a low-dimensional space while preserving the structure and distribution information of the original data. Therefore, it could help to better understand the features of the data. By calling the ISOGA-CNN model in 3.3.2, the features of each layer of the ISOGA-CNN model were extracted by looping through them, and then UMAP visualized these features as points in a 3D space. Each point represented a sample in the dataset, and the color of the point represented the corresponding label.

Fig. 6 shows the process of UMAP visualizing the ISOGA-CNN feature layers. Fig. 6(a)–6(e) show that after the input data goes through 2 convolutional layers and 1 average pooling layer, and then 2 more convolutional layers for feature extraction, all the samples are scattered on an irregular surface. The spatial distance between the control group samples and contaminated group samples with longer storage time was far, but there was no obvious separation trend. Fig. 6(f) and (g) show that after the feature extraction of the 5th convolutional layer, the control group and the contaminated groups are clearly separated, but there are still a few samples of day 1 group that are not distinguished from the control group samples, and the dropout layer does not improve the separation of the sample data. Fig. 6(h) shows that after the feature extraction of the 6th convolutional layer, the spatial distribution of the data changes significantly, and the control group and day 1 group are clearly separated from other contaminated groups, but the remaining contaminated group samples are not separated according to storage time. Fig. 6(i) to Fig. 6(l) show that after the feature extraction of all the convolutional layers, the ISOGA-CNN model is not able to successfully distinguish between contaminated samples with different storage times. Similarly, after the data was dimensionally reduced by the flatten layer, the day 3, day 5, and day 7 groups still could not be distinguished. Fig. 6(m) and (n) show that after feature extraction and classification by the fully connected layer, samples from all groups are accurately classified. Compared with the initial CNN model, the ISOGA-CNN model had more feature layers, could extract deeper features of the data, grasp the global structure of the data, and achieve accurate classification of contaminated samples with different storage times.

Therefore, some insights of the UMAP-ISOGA-CNN algorithm in optimizing deep learning models were further expounded. Firstly, by visualizing the feature layers of ISOGA-CNN using UMAP, it provides a deeper understanding of how the model transforms the input data as it passes through each layer. This helps reveal which feature layers are

capturing important patterns and features, and which ones might need further adjustment or regularization. This is similar to the activation visualization function, which serves as the gold standard method and helps understand specific patterns or features that the model is learning. Secondly, the UMAP-ISOGA-CNN algorithm can uncover whether the model is overfitting or underfitting the data by observing the distribution and separation of data points in the lower-dimensional space. Thirdly, the distribution of predicted results might significantly differ from the true labels. The algorithm can highlight such discrepancies, indicating potential issues in the model's learning process or the quality of the dataset. Lastly, the algorithm helps validate whether the features extracted by different layers of the ISOGA-CNN are semantically meaningful and consistent with patterns relevant to the task. If the visual representations appear meaningful and well-separated, it suggests that the model is learning relevant representations. Overall, UMAP-ISOGA-CNN provides visualization for each feature layer of ISOGA-CNN, offering an additional tool for understanding the model's behavior, identifying potential flaws, and guiding optimization efforts. By gaining deeper insights into how the model processes data, informed decisions can be made regarding DL model architecture modifications, hyperparameter tuning, and data preprocessing to enhance the DL model's performance and generalization capabilities.

4. Conclusion

In this study, a novel UMAP-ISOGA-CNN algorithm was proposed, which combined ISOGA with CNN to optimize the architecture, hyperparameters, and optimizer of the CNN model simultaneously. Additionally, the UMAP method was used to visualize the feature space of all feature layers of the ISOGA-CNN model. ISOGA-CNN combined with visible and near-infrared hyperspectral imaging was utilized to identify peanut kernels contaminated with *Aspergillus flavus* and distinguish the storage time of them. The experimental results showed that ISOGA-CNN had better performance in identifying contaminated peanut kernels compared to other traditional machine learning algorithms, which achieved the highest accuracy of 99.37% and 100.00% in the 2-class classification and 6-class classification on the test set, respectively. Furthermore, ISOGA-CNN was also capable of distinguishing the storage time of contaminated peanut kernels, which was essential for the food industry to monitor the freshness of products. In order to guarantee the algorithm's robustness, ISOGA-CNN was executed independently for 100 times, of which 46 runs produced fitness scores exceeding 0.9800. The results demonstrated the model's consistent and trustworthy performance. Moreover, the influence of hyperparameters and optimizers on the performance of the CNN model was investigated, providing useful insights for future research on the optimization of DL models. Overall, the proposed ISOGA-CNN model demonstrated promising results in identifying contaminated peanut kernels and distinguishing their storage time. The UMAP-ISOGA-CNN method provides useful insights into the feature space of the ISOGA-CNN model, contributing to a better understanding of the model's internal mechanisms. This study has practical implications for the food industry and future research on deep learning optimization.

Credit author statement

Zhen Guo: Investigation, Conceptualization, Methodology, Formal analysis, Writing – original draft; Jing Zhang: Investigation, Software, Data curation, Validation, Writing–review & editing; Jiashuai Sun: Investigation, Methodology, Formal analysis, Writing–review & editing; Haowei Dong: Investigation, Methodology, Supervision, Writing – review & editing; Jingcheng Huang: Methodology, Supervision, Resources, Writing – review & editing; Lingjun Geng: Project administration, Resources, Writing – review & editing; Shiling Li: Methodology, Supervision, Resources, Writing – review & editing; Xiangzhu Jing: Investigation, Writing–review & editing; Yemin Guo:

Supervision, Resources, Software, Project administration, Funding acquisition, Writing–review & editing; Xia Sun: Supervision, Resources, Software, Project administration, Funding acquisition, Writing–review & editing.

Ethical approval

This article has no any study with human participants or animals by any of the authors.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 31772068), Funding Project for the Central Government to Guide the Development of Local Science and Technology (YDZX2022163), Shandong Province Major Applied Technology Innovation Project (SD2019NJ007), and Weifang Science and Technology Development Project (2021ZJ1103).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.talanta.2023.125187>.

References

- [1] J. Higgs, The beneficial role of peanuts in the diet—Part 2, *Nutr. Food Sci.* (N. Y.) 33 (2003) 56–64, <https://doi.org/10.1108/00346650310466637>.
- [2] R. Liu, M. Chang, Q. Jin, J. Huang, Y. Liu, X. Wang, Degradation of aflatoxin B₁ in aqueous medium through UV irradiation, *Eur. Food Res. Technol.* 233 (2011) 1007–1012, <https://doi.org/10.1007/s00217-011-1591-9>.
- [3] X. Qiao, J. Jiang, X. Qi, H. Guo, D. Yuan, Utilization of spectral-spatial characteristics in shortwave infrared hyperspectral images to classify and identify fungi-contaminated peanuts, *Food Chem.* 220 (2017) 393–399, <https://doi.org/10.1016/j.foodchem.2016.09.119>.
- [4] Y. Lu, B. Jia, S.C. Yoon, H. Zhuang, X. Ni, B. Guo, S.E. Gold, J.C. Fountain, A. E. Glenn, K.C. Lawrence, H. Zhang, X. Guo, F. Zhang, W. Wang, Spatio-temporal patterns of *Aspergillus flavus* infection and aflatoxin B₁ biosynthesis on maize kernels probed by SWIR hyperspectral imaging and synchrotron FTIR microspectroscopy, *Food Chem.* 382 (2022), 132340, <https://doi.org/10.1016/j.foodchem.2022.132340>.
- [5] F. Tao, H. Yao, Z. Hruska, R. Kincaid, K. Rajasekaran, D. Bhatnagar, A novel hyperspectral-based approach for identification of maize kernels infected with diverse *Aspergillus flavus* fungi, *Biosyst. Eng.* 200 (2020) 415–430, <https://doi.org/10.1016/j.biosystemseng.2020.10.017>.
- [6] Y. Sun, Z. Ye, M. Zhong, K. Wei, F. Shen, G. Li, J. Yuan, C. Xing, Rapid and nondestructive method for identification of molds growth time in wheat grains based on hyperspectral imaging technology and chemometrics, *Infrared Phys. Technol.* 128 (2023), 104532, <https://doi.org/10.1016/j.infrared.2022.104532>.
- [7] J. Jiang, X. Qiao, R. He, Use of Near-Infrared hyperspectral images to identify moldy peanuts, *J. Food Eng.* 169 (2016) 284–290, <https://doi.org/10.1016/j.jfoodeng.2015.09.013>.
- [8] C.-H. Feng, Y. Makino, S. Oshita, J.F. García Martín, Hyperspectral imaging and multispectral imaging as the novel techniques for detecting defects in raw and processed meat products: current state-of-the-art research advances, *Food Control* 84 (2018) 165–176, <https://doi.org/10.1016/j.foodcont.2017.07.013>.
- [9] H.-J. He, D.-W. Sun, Hyperspectral imaging technology for rapid detection of various microbial contaminants in agricultural and food products, *Trends Food Sci. Technol.* 46 (2015) 99–109, <https://doi.org/10.1016/j.tifs.2015.08.001>.
- [10] D. Yuan, J. Jiang, X. Qi, Z. Xie, G. Zhang, Selecting key wavelengths of hyperspectral image for nondestructive classification of moldy peanuts using ensemble classifier, *Infrared Phys. Technol.* 111 (2020), 13058, <https://doi.org/10.1016/j.infrared.2020.103518>.
- [11] S.M. Mansuri, S.K. Chakraborty, N.K. Mahanti, R. Pandiselvam, Effect of germ orientation during Vis-NIR hyperspectral imaging for the detection of fungal contamination in maize kernel using PLS-DA, ANN and 1D-CNN modelling, *Food Control* 139 (2022), 109077, <https://doi.org/10.1016/j.foodcont.2022.109077>.
- [12] A. Bhandare, D. Kaur, Designing convolutional neural network architecture using genetic algorithms, 6, in: 2018 World Congress in Computer Science, Computer Engineering and Applied Computing, CSCE 2018-Proceedings of the 2018 International Conference on Artificial Intelligence, ICAI, 2018, pp. 26–35, <https://doi.org/10.21307/ijanmc-2021-024>, 2018.
- [13] J. Reiling Anthony, *Convolutional Neural Network Optimization Using Genetic Algorithms*, Diss. Dayton: University of Dayton, 2017.
- [14] D.E. Goldberg, J.H. Holland, *Genetic algorithms and machine learning*, *Mach. Learn.* 3 (1988) 95–99.
- [15] D. Whitley, *A genetic algorithm tutorial*, *Stat. Comput.* 4 (1994) 65–85.
- [16] J. Li, T. Liu, X. Wang, J. Yu, Automated asphalt pavement damage rate detection based on optimized GA-CNN, *Nutr. Food Sci.* 136 (2022), 104180, <https://doi.org/10.1016/j.autcon.2022.104180>.
- [17] Y. Yu, M. Zhang, Control chart recognition based on the parallel model of CNN and LSTM with GA optimization, *Expert Syst. Appl.* 185 (2021), 115689, <https://doi.org/10.1016/j.eswa.2021.115689>.
- [18] T.H.T. Nguyen, Q.B. Phan, Hourly day ahead wind speed forecasting based on a hybrid model of EEMD, CNN-Bi-LSTM embedded with GA optimization, *Energy Rep.* 8 (2022) 53–60, <https://doi.org/10.1016/j.ejegyr.2022.05.110>.
- [19] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Completely automated CNN architecture design based on blocks, *IEEE Transact. Neural Networks Learn. Syst.* 31 (2020) 1242–1254, <https://doi.org/10.1109/TNNLS.2019.2919608>.
- [20] R. de Lima Mendes, A.H. da Silva Alves, M. de Souza Gomes, P.L.L. Bertarini, L.R. do Amaral, gaCNN: composing CNNs and GAs to build an optimized hybrid classification architecture, 2021, in: 2021 IEEE Congress on Evolutionary Computation, CEC, 2021, pp. 79–86, <https://doi.org/10.1109/CEC45853.2021.9504850>. - Proceedings.
- [21] G. Montavon, W. Samek, K.R. Müller, Methods for interpreting and understanding deep neural networks, *Digit. Signal Process.* 73 (2018) 1–15, <https://doi.org/10.1016/j.dsp.2017.10.011>.
- [22] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: uniform manifold approximation and projection, *J. Open Source Softw.* 3 (2018) 861, <https://doi.org/10.21105/joss.00861>.
- [23] M. Vermeulen, K. Smith, K. Eremin, G. Rayner, M. Walton, Application of uniform manifold approximation and projection (UMAP) in spectral imaging of artworks, *Spectrochim. Acta, Part A* 252 (2021), 119547, <https://doi.org/10.1016/j.saa.2021.119547>.
- [24] Y. Yang, H. Sun, Y. Zhang, T. Zhang, J. Gong, Y. Wei, Y.G. Duan, M. Shu, Y. Yang, D. Wu, D. Yu, Dimensionality reduction by UMAP reinforces sample heterogeneity analysis in bulk transcriptomic data, *Cell Rep.* 36 (2021), 109442, <https://doi.org/10.1016/j.celrep.2021.109442>.
- [25] Y. Long, W. Huang, Q. Wang, S. Fan, X. Tian, Integration of textural and spectral features of Raman hyperspectral imaging for quantitative determination of a single maize kernel mildew coupled with chemometrics, *Food Chem.* 372 (2022), 131246, <https://doi.org/10.1016/j.foodchem.2021.131246>.
- [26] Z. Guo, J. Zhang, C. Ma, X. Yin, Y. Guo, X. Sun, C. Jin, Application of visible-near-infrared hyperspectral imaging technology coupled with wavelength selection algorithm for rapid determination of moisture content of soybean seeds, *J. Food Compos. Anal.* 1161 (2023), 105048, <https://doi.org/10.1016/j.jfca.2022.105048>.
- [27] D. Kimuli, W. Wang, W. Wang, H. Jiang, X. Zhao, X. Chu, Application of SWIR hyperspectral imaging and chemometrics for identification of aflatoxin B₁ contaminated maize kernels, *Infrared Phys. Technol.* 89 (2018) 351–362, <https://doi.org/10.1016/j.infrared.2018.01.026>.
- [28] M.W. Dorrrity, L.M. Saunders, C. Queitsch, S. Fields, C. Trapnell, Dimensionality reduction by UMAP to visualize physical and genetic interactions, *Nat. Commun.* 11 (2020) 1–6, <https://doi.org/10.1038/s41467-020-15351-4>.
- [29] D. Milošević, A.S. Medeiros, M. Stojković Piperač, D. Cvijanović, J. Soininen, A. Milosavljević, B. Predić, The application of Uniform Manifold Approximation and Projection (UMAP) for unconstrained ordination and classification of biological indicators in aquatic ecology, *Sci. Total Environ.* 815 (2022), 152365, <https://doi.org/10.1016/j.scitotenv.2021.152365>.
- [30] E.R.K. Neo, Z. Yeo, J.S.C. Low, V. Goodship, K. Debattista, A review on chemometric techniques with infrared, Raman and laser-induced breakdown spectroscopy for sorting plastic waste in the recycling industry, *Resour. Conserv. Recycl.* 180 (2022), 106217, <https://doi.org/10.1016/j.resconrec.2022.106217>.
- [31] V.E. de Almeida, D.D. de Sousa Fernandes, P.H.G.D. Diniz, A. de Araújo Gomes, G. Véras, R.K.H. Galvão, M.C.U. Araújo, Scores selection via Fisher's discriminant power in PCA-LDA to improve the classification of food data, *Food Chem.* 363 (2021), 130296, <https://doi.org/10.1016/j.foodchem.2021.130296>.
- [32] M. Aminu, N.A. Ahmad, Complex chemical data classification and discrimination using locality preserving partial least squares discriminant analysis, *ACS Omega* 5 (2020) 26601–26610, <https://doi.org/10.1021/acsomega.0c03362>.
- [33] J. Zhu, H. Li, Z. Rao, H. Ji, Identification of slightly sprouted wheat kernels using hyperspectral imaging technology and different deep convolutional neural networks, *Food Control* 143 (2023), 109291, <https://doi.org/10.1016/j.foodcont.2022.109291>.
- [34] L. Pang, L. Wang, P. Yuan, L. Yan, Q. Yang, J. Xiao, Feasibility study on identifying seed viability of *Sophora japonica* with optimized deep neural network and hyperspectral imaging, *Comput. Electron. Agric.* 190 (2021), 106426, <https://doi.org/10.1016/j.compag.2021.106426>.
- [35] D. Wang, F. Tian, S.X. Yang, Z. Zhu, D. Jiang, B. Cai, Improved deep CNN with parameter initialization for data analysis of near-infrared spectroscopy sensors, *Sensors* 20 (2020) 874, <https://doi.org/10.3390/s20030874>.

- [36] X. He, C. Yan, X. Jiang, F. Shen, J. You, Y. Fang, Classification of aflatoxin B₁ naturally contaminated peanut using visible and near-infrared hyperspectral imaging by integrating spectral and texture features, *Infrared Phys. Technol.* 114 (2021), 103652, <https://doi.org/10.1016/j.infrared.2021.103652>.
- [37] D. Yang, J. Jiang, Y. Jie, Q. Li, T. Shi, Detection of the moldy status of the stored maize kernels using hyperspectral imaging and deep learning algorithms, *Int. J. Food Prop.* 25 (2022) 170–186, <https://doi.org/10.1080/10942912.2022.2027963>.
- [38] F. Shen, T. Zhao, X. Jiang, X. Liu, Y. Fang, Q. Liu, Q. Hu, X. Liu, On-line detection of toxicogenic fungal infection in wheat by visible/near infrared spectroscopy, *LWT* 109 (2019) 216–224, <https://doi.org/10.1016/j.lwt.2019.04.019>.