RSC Advances

PAPER

Cite this: *RSC Adv.*, 2025, **15**, 37039

DeepPHSI: attention-driven CNN-LSTM fusion for hyperspectral origin traceability across *Pogostemon cablin* batches

Xiaqiong Fan,[a] Yulin Liu,[a] Zihao Zhang,[a] Peijun Zhao,[a] Zhengyan Li,[b] Junjun Zhou,[c] Dandan Zhai,[c] Yi Hu,[a] Peng Li*[d] and Hongchao Ji *[b]

Received 2nd September 2025
Accepted 20th September 2025

DOI: 10.1039/d5ra06579h

rsc.li/rsc-advances

Pogostemon cablin (*P. cablin*) is rich in chemical compounds and is extensively utilized in the medicine, food, and fragrance industries. However, factors such as variety, regional ecology, growth conditions, harvest time, and processing methods result in differences in the traits and quality of *P. cablin* from different origins. The traditional labor-intensive identification methods require a lot of manpower and material resources, and the accuracy of identification is also affected by individual subjectivity. In this study, a deep learning network based on a pixel-level hyperspectral image was constructed to identify *P. cablin* from different origins, named DeepPHSI. DeepPHSI can be used to distinguish between the three main origins of *P. cablin* and their stems and leaves from the background. The DeepPHSI model was designed based on convolutional neural and long short-term memory networks. The hyperspectral image data collected under two experimental conditions were used for training and fine-tuning, respectively. Results showed that DeepPHSI can accurately identify the origin of *P. cablin* under different experimental conditions with transfer learning. The prediction based on DeepPHSI also enabled the fully automated identification of origins and parts, which makes the model suitable for the rapid analysis of large-scale samples. These advantages make DeepPHSI a promising method in hyperspectral applications.

1 Introduction

Pogostemoncablin (*P. cablin*), a member of the Lamiaceae family, is a prominent medicinal and edible plant with its origin in tropical regions. *P. cablin* serves as a traditional Chinese medicine (TCM) for the treatment of diarrhea, vomiting, nausea and fever.[1] In addition, the whole plant of *P. cablin* contains volatile oils, which are common ingredients in perfumes, fragrances, and cosmetics. Research has revealed that *P. cablin* contains many active components, including polysaccharides,[2] terpenoids,[3] flavonoids,[4] phytosterols, organic acids,[5] phenols,[6] alkaloids, and glycosides.[7] Notably, its volatile oils are recognized as a safe natural food flavoring by the U.S. Food and Drug Administration (FDA) and are widely incorporated into food products, including beverages, gelatin-based meats, meat products, and frozen dairy desserts. In China, *P. cablin* is mainly cultivated in Guangzhou City, Zhaoqing City, Zhanjiang City in Guangdong Province, and some areas in Hainan Province.[8] However, the composition of its volatile oils varies significantly depending on the growing area and harvest time. Based on these chemical differences, *P. cablin* is classified into two distinct chemotypes: the pogostone type and the patchouliol type.[9,10] This classification reflects the unique chemical profiles of its volatile oils, which are influenced by environmental and temporal factors. These variations can influence the pharmacological properties, fragrance quality, and commercial value of *P. cablin*, making standardization and quality assessment crucial. Therefore, quickly identifying *P. cablin* from different origins has an important practical significance.

Due to their high resolution and rich spectral information, hyperspectral images (HSIs)[11,12] provide sufficient data features within narrow bands. In addition, the advantages of non-destructive and rapid detection make HSI a promising tool for online detection and real-time monitoring.[13,14] These advantages of HSI technology offer it an important role in various fields such as environmental monitoring,[15] resource exploration,[16] agricultural management,[17] biomedical science[18,19] and chemical analysis.[20,21] However, due to changes in the measurement environment, instrument parameters, and lighting conditions, there are often differences in the data collected from different batches. Therefore, data analysis methods based on a single batch of data often cannot be directly applied to data from other batches. In addition, as

[a] School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou 450001, PR China

[b] Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen 518120, PR China. E-mail: jihongchao@caas.cn

[c] School of Biological Engineering, Henan University of Technology, Zhengzhou 450001, PR China

[d] Institute for Complexity Science, Henan University of Technology, Zhengzhou 450001, PR China. E-mail: lipeng@haut.edu.cn

hyperspectral data offer both spatial and spectral information, data analysis is relatively complex.

In recent years, researchers have proposed many methods for HSI data analysis and applications. A framework[22] for the classification of hyperspectral scenes has been proposed, which pursues the combination of multiple features. The framework can cope with the linear and nonlinear class boundaries present in the data. Structured sparsity priors based on the task-driven dictionary learning algorithm[23] can benefit from the advantages of both the simultaneous sparse representation and supervised dictionary learning. A super-pixel-based sparse representation (SSR)[24] model was proposed for hyperspectral image super-resolution. The high-resolution hyperspectral image can be reconstructed with the obtained fractional abundance coefficient matrix. Spatial-spectral hypergraph discriminant analysis (SSHGDA)[25] can effectively reveal the complex spatial-spectral structures of HSI and enhance the discriminating power of the features for land-cover classification. The active broad learning system approach[26] was applied to extract the spectral and spatial features of the image using principal component analysis and local binary patterns, respectively. Active learning was used to select high-quality samples, which can reduce the cost of sample labelling. An unfolding network with disentangled spatial-spectral representation[27] was proposed for analysing the super-resolution HSI. A variant of depth-wise separable convolution and a lightweight spectral attention mechanism were used to adequately incorporate the structure prior of the HSI. Deep learning and a constrained optimization-based approach[28] were proposed for hyperspectral image denoising and spectral compressive imaging. A Gaussian mixture model[29] was employed to cluster the HSI to achieve segmentation. It meets the requirements of high-throughput and real-time analysis.

Deep learning[30,31] has made significant breakthroughs in computer vision,[32] natural language processing,[33] speech recognition,[34] and other related fields.[35,36] The main reason is that flexible architectures and efficient algorithms can learn multilevel representational features directly from large amounts of raw data. Due to these advantages, deep learning methods have gained increasing attention in agriculture for several applications, such as gene expression prediction,[37] phenotypic prediction,[38] and crop yield prediction.[39] Specific to HSI, deep learning plays an important role, especially Convolutional Neural Networks (CNN).[40] A CNN model with 1 × 1 convolutional layers[41] has been adopted for boosting the discrimination accuracy of hyperspectral image classification, where the original data are used as the input and the final CNN outputs are the predicted class-related results. Another framework that takes advantage of both CNNs and multiple feature learning[42] was proposed to better predict the class labels for HSI pixels. Various features extracted from the raw imagery were used as input to obtain joint feature maps. The diverse region-based CNN method[43] encodes semantic context-aware representation to obtain promising features. It exploits diverse region-based inputs to learn contextual interactional features and is, thus, expected to have more discriminative power. A variant of depth-wise separable convolution[27] was used to disentangle and extract the spatial and spectral features. The results show that the difficulty and computational complexity of feature learning can be reduced. A heterogeneous network[44] can extract spectral-spatial and semantic features simultaneously. The network was established based on the semantic transformer scheme and the spectral-spatial convolution network branch. The reinforced pool-based deep active learning approach[45] was proposed to overcome the limitations of statistical selection approaches. The reinforcement learning-based agent can transfer and choose samples for annotation on other HSI dataset after being trained. A 3D grouped convolution[46] was designed as a vehicle to convey the semantic features, and it conveys the properties of HSI data well in the time and space domains. Image-level annotation is introduced to predict pixel-level classification maps[47] for HSI. The proposed method explores weakly supervised HSI classification with image-level tags, bridging the gap between image-level annotation and dense prediction.

In this study, we present DeepPHSI, a transfer learning-enabled framework for multi-origin identification of *Pogostemon cablin* (*P. cablin*) in hyperspectral imaging (HSI) data. This framework advances current hyperspectral analysis paradigms through three key innovations: (1) a hybrid architecture combining CNN's spatial feature extraction with LSTM's sequential pattern modeling, overcoming the single-modality limitations in existing methods; (2) a transfer learning protocol that freezes CNN layers to preserve cross-batch invariant features while adaptively tuning LSTM parameters, resolving the batch effect issue prevalent in prior studies; (3) pixel-level classification, enabling precise spatial mapping of chemical heterogeneity within plant tissues. The implementation of this methodology establishes a robust platform for large-scale, automated origin authentication, directly addressing the industrial demands identified in recent quality control analyses; the model is publicly accessible at **https://github.com/xiaoyulinoOO/DeepPHSI**.

2 Methods

2.1 Overview of DeepPHSI for origin and anatomical part identification

DeepPHSI serves as a deep learning model designed to accurately predict the origins and anatomical parts of *P. cablin* using HSI data. To address the challenge of batch-to-batch variability in HSI data, we implemented a transfer learning strategy, enabling the model to adapt to different acquisition conditions while maintaining a high performance. DeepPHSI integrates three key components: CNNs for spatial-feature extraction, LSTMs for sequential–spectral–pattern modeling, and attention mechanisms for adaptive-feature weighting. In HSI data, spatial information captures the local leaf structure (veins, texture, *etc.*) and branch characteristics (thickness, color, *etc.*), and spectral dimensions capture some of the intrinsic qualities of *P. cablin*. Extensive validation experiments demonstrate that DeepPHSI achieves superior identification accuracy and exhibits broad applicability in HSI data analysis. The workflow of the model,

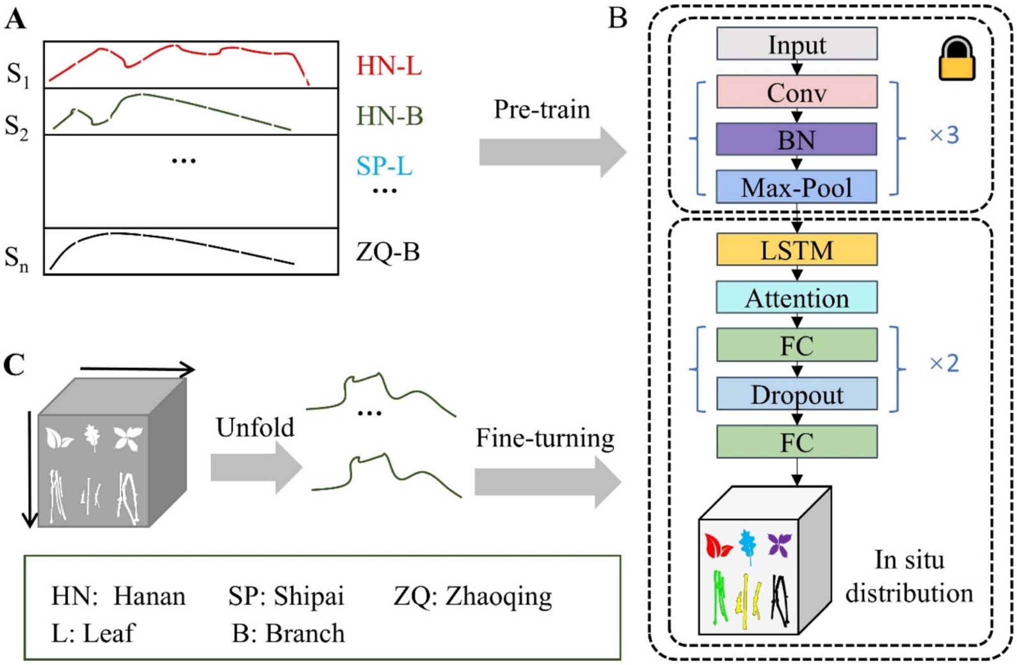

Fig. 1 Flowchart of pre-training and fine-tuning of the DeepPHSI model. (A) Standard training dataset for the DeepPHSI model. (B) Architecture of DeepPHSI. The convolutional (Conv) layer with batch normalization (BN) is frozen during fine-tuning. The LSTM layer, combined with the attention mechanism and the fully connected (FC) layers, is fine-tuned. The "lock" represents the freezing operation.

including the pre-training and fine-tuning stages, is illustrated in Fig. 1.

2.2 Architecture of DeepPHSI

2.2.1 CNN for spatial feature extraction. CNN[40] is a neural network inspired by biological vision, consisting of convolutional layers and convolutional kernels. The convolutional kernels are moved in a certain way on the convolutional layers to calculate the convolutional features. Compared with fully connected neural networks, a CNN has the characteristics of shared weights and sparse connections. Shared weights can improve learning efficiency and achieve better generalization. Sparse connections allow neural networks to produce the strongest response to local input patterns. The Batch Normalization (BN) layer is applied behind each convolution layer. The output of these convolutional layers is a high-dimensional feature map, which is normalized by the BN layer to ensure that the subsequent Rectified Linear Unit (ReLU)[48] activation function and Max-Pooling layer can work effectively. The BN layer is used to speed up the training process, improve the stability of the model, and reduce overfitting by standardizing the input of each layer.

2.2.2 LSTM for sequential pattern modeling. LSTM[49] is one of the recurrent neural networks (RNN); it can capture long-term dependencies in sequence data. For HSI spectral data, LSTM captures dependencies between adjacent bands in a spectral sequence. After feature extraction in the convolution layer, a feature map is rearranged into a form suitable for LSTM and fed into the LSTM layer. In DeepPHSI, the LSTM layer has an input size of 64 (corresponding to the feature dimension after convolution). The hidden size is 128. It is a single layer with a unidirectional structure (bidirectional = false). This architecture enables the layer to further learn temporal dependencies in spectral data, empowering the model to capture and process long-range dependencies within spectral sequences.

2.2.3 Attention mechanism for capturing important information. The attention mechanism can enhance the model's attention to different parts of the inputs. The core idea is to dynamically assign different weights to different parts of the input, allowing the model to focus on the information that is most important to the task. In this study, an attention mechanism is added behind the LSTM layer, which is an important improvement. The attention layer generates a weighted context vector by calculating the attention weights for each time step, and then, these weights are applied to the output of the LSTM. The weighted context vector can highlight those features that contribute more to the final decision. The attention mechanism allows the model to capture important information in serialized spectral data, thereby improving the classification effect. The output of the LSTM layer combined with the attention mechanism is a context vector, based on weighted summation, which represents the overall "summary" of the LSTM output. The structure of the LSTM layer, combined with the attention mechanism, provides more targeted and effective information for the subsequent fully connected layer. In HSI, the distinction between key bands and redundant bands is achieved through adaptive learning of attention weights. Specifically, the attention layer calculates weights for the output sequence of the LSTM and normalizes them *via* the softmax function, ensuring the sum of all band weights equals 1. Bands with higher weights are considered more critical, while those with lower weights are automatically suppressed as redundant features. Since the softmax mechanism ensures the relative discriminability of weights, no explicit fixed threshold needs to be set. The model dynamically learns the optimal weight distribution during training, thereby ensuring ultimate focus on the spectral features relevant to geographical origin.

2.2.4 Summary of the DeepPHSI architecture. The network architecture of the DeepPHSI model is shown in Fig. 2. The spectra extracted from each pixel in the HSI data were used as the model input. First, three one-dimensional convolution layers were used to extract local band features from the input. In the first convolutional layer, the kernel size is 11, and the number of channels is 16. The kernel size of the second convolution layer is 7, and the number of channels is 32. The kernel size of the third convolution layer is 3, and the number of channels is 64. The BN and max-pooling were added after each convolutional layer. Then, the feature maps from the CNN layers were rearranged and fed into the LSTM layer. This was achieved by treating the number of output channels from the CNN layer as the input feature dimension of LSTM while considering the spectral band dimension as the sequence dimension. Thus, we ensured that the spatial features extracted by the CNN can be fed into the LSTM sequentially according to the spectral order. In this process, the fusion of spatial and spectral features is jointly accomplished by the LSTM and attention mechanism: the spatial features extracted by the CNN layers are used as input to the LSTM, while the LSTM dynamically learns dependencies along the temporal dimension (*i.e.*, the spectral band direction). Finally, the attention mechanism assigns different weights to the spectral band features outputted by the LSTM, enabling the model to adaptively emphasize the key spectral features relevant to the classification task. The introduction of the attention mechanism also reduces the influence of redundant bands, thereby achieving a weighted fusion of spatial and spectral features. The attention mechanism is applied to the output of the LSTM, which ensures that the features extracted from the LSTM are fed into the fully connected layer more efficiently. Finally, the fully connected layers were used for classification, and the dropout was added after the first two fully connected layers, respectively. These double dropouts help to reduce the risk of overfitting. Dropout prevents the network from over-relying on specific weights by randomly discarding a subset of neurons, thereby improving the model's ability to generalize. The Rectified Linear Unit (ReLU) is used to activate the features after linear transformation, which enhances the nonlinear representation ability of the model. The last fully connected layer maps the output of the model to the final classification, generating the prediction results.

The neural network architecture of DeepPHSI was optimized for improved performance. Here, the number of convolutional

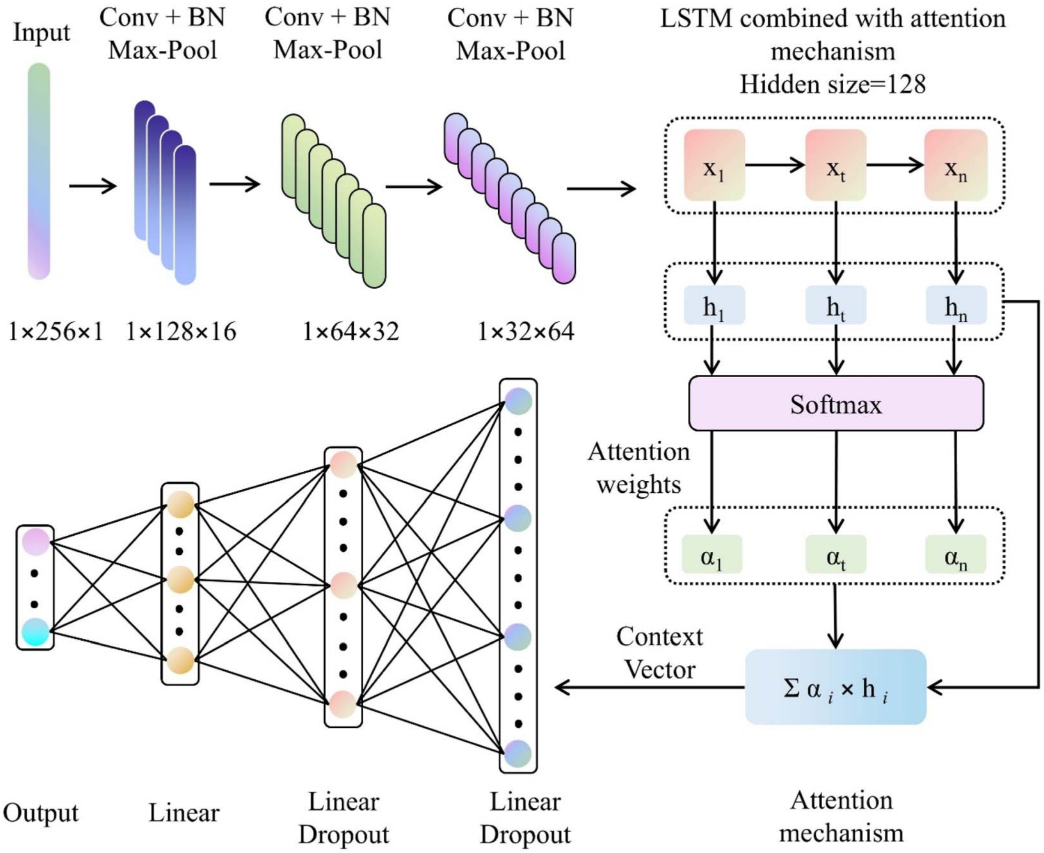

Fig. 2 Architecture of the DeepPHSI model.

layers for feature extraction, the hidden size of the LSTM layer, the number of dropout operations in the fully connected layers, and batch normalization are the key hyperparameters to be considered. These considered architectures are shown in Fig. S1. The corresponding learning curves of these considered architectures are shown in Fig. S2.

2.3 Description of the *P. cablin* datasets

Fresh leaf and stem tissues of *P. cablin* plants were collected from each cultivar, with individual leaves and stems sampled in sextuplicate. In this study, two batches of *P. cablin* hyperspectral data were acquired, covering leaves and branches from three different origins: Zhaoqing, Shipai, and Hainan. These datasets were obtained under different experimental conditions.

Both datasets were measured using the GaiaSorter-Dual Gaia full-band hyperspectral sorter, covering a spectral range of 900–1700 nm. In the first dataset, the wavelength range was 853–1701 nm, with spectral intensity values ranging from 362 to 16 383. The spatial resolutions of the hyperspectral images (HSI) were 600 × 320 and 535 × 320. In the second dataset, the wavelength range remained 853–1701 nm, but the spectral intensity values ranged from 0 to 33, with spatial resolutions of 300 × 320 and 374 × 320. Two sets of bromine tungsten light sources were used to illuminate the sample table, providing uniform irradiation through thermal radiation. The non-uniformity of the light source within a volume of 300 × 20 × 100 mm (length × width × height) was less than 5%.

2.4 Data preprocessing and augmentation

A total of 1 131 200 spectra were extracted from the first batch of the hyperspectral imaging (HSI) dataset. For the second dataset, a total of 311 680 spectra were extracted. To enhance the model's robustness and generalization ability, as well as to improve the classification performance, data augmentation was applied. The number of spectra per class after the augmentation is shown in Fig. S3. Both batches of data were divided into training set, validation set, and test set with a ratio of 8 : 1 : 1, respectively. Please see more detailed information about the data augmentation in Text S1.

Data preprocessing methods may have a significant impact on the performance of the model. In order to ensure that the pre-trained model delivers optimal performance, different data preprocessing methods were used for comparison. The comparison results show that the model trained directly using the original HSI data achieved the highest accuracy of 98.48% on the test set. Please see more detailed information about the data augmentation in Table S1.

2.5 Model training

During training, some standardized data extracted from the first batch of hyperspectral images were used for model training (see Fig. 1A). The cross-entropy loss was used as the loss function. The cross-entropy loss is a loss function for handling multiple classification tasks to calculate the gap between the probability distribution predicted by the model and the actual label. It is suitable for the current seven classification tasks in this study.

Adam with weight decay (AdamW)[50] was used as the optimizer. AdamW uses an adaptive learning rate for each parameter to help accelerate convergence and dynamically adjust the learning rate during training. Unlike the standard Adam optimizer, AdamW applies weight decay to each parameter, allowing for better control of the model's complexity and preventing overfitting.

The learning rate decay strategy was adopted to accelerate convergence and improve the model's performance. In this study, ReduceLROnPlateau adjusts the learning rate according to changes in the loss of the validation set. When the loss of the validation set does not improve significantly within a certain round, the learning rate is reduced. ReduceLROnPlateau helps the model adjust the parameters more carefully as it approaches the optimal solution.

2.6 Transfer learning

Affected by experimental conditions, there are large differences between different batches of HSI data. The method established based on one batch of HSI data is usually not applicable to other batches of data. Therefore, transfer learning is adopted to solve the problem of applying the model to different batches of data. During the model fine-tuning, freezing different layers has a significant impact on model performance. In this study, several different fine-tuning methods were compared, including fine-tuning the fully connected layer; fine-tuning the fully connected layer and the attention layer; and fine-tuning the fully connected layer, the attention layer, and the LSTM layer. The detailed comparison results are shown in Table S2. After comparison, transfer learning was performed by fine-tuning the fully connected layer, the attention layer, and the LSTM layer. As can be seen in Fig. 1B, a "lock" is used to represent the freezing operation.

Fine-tuning the LSTM, attention and fully connected layers can help the model better adapt to new tasks or datasets. Fine-tuning can enhance the model's ability to distinguish different classes, especially in a new batch of data. The flowchart of the fine-tuning of the DeepPHSI model is shown in Fig. 1B and C.

2.7 Implementation

DeepPHSI is implemented in the Python programming language and based on NumPy and PyTorch. The operating system is Windows 11, with NVIDIA GeForce RTX 3090. DeepPHSI is available at https://github.com/xiaoyulinoOO/DeepPHSI.

3 Results and discussion

3.1 Overall performance of DeepPHSI

Three deep learning models, CNN (Fig. S4A), LSTM (Fig. S4B) and multi-head attention (Fig. S4C), were established and compared to demonstrate the necessity of the DeepPHSI architecture.

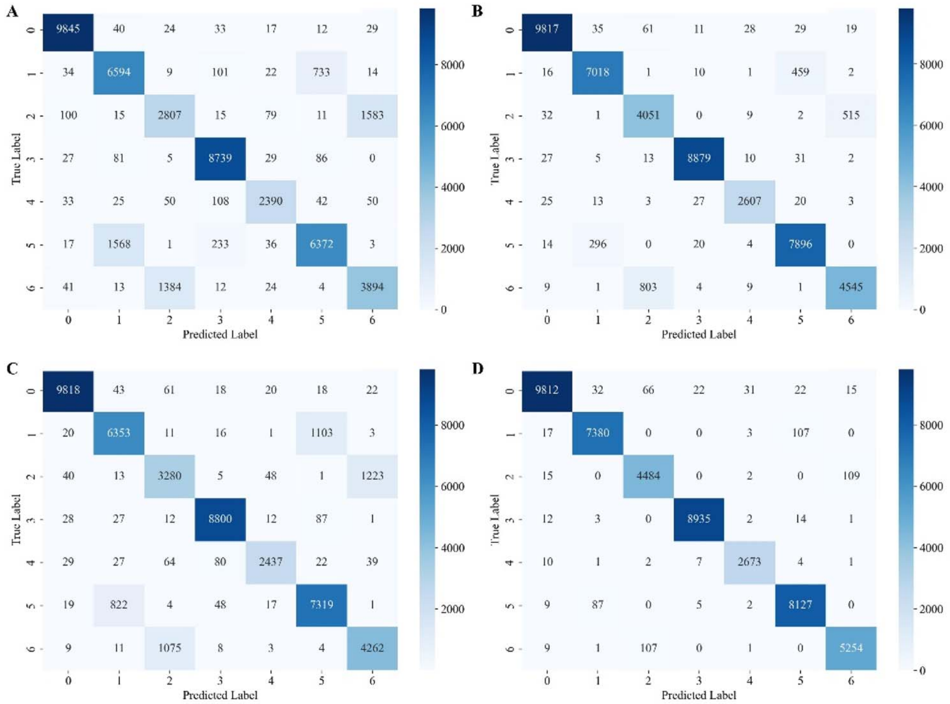

Fig. 3 Confusion matrix of different models on the test set of the first batch of data: (A) multi-head attention, (B) CNN, (C) LSTM, and (D) DeepPHSI.

The confusion matrices of these considered network architectures on the test set of the first batch of data are shown in Fig. 3. As can be seen from Fig. 3A, the multi-head attention model makes the most errors in the prediction of *P. cablin* from Zhaoqing (class 6) and Hainan (class 1 and 2). There were 1568 leaves and 1384 branches of *P. cablin* from Zhaoqing, which were not correctly identified. There were 733 leaves and 1583 branches of *P. cablin* from Hainan, which were not correctly identified. The LSTM model predicted classes 5 and 6 (the leaves and branches of *P. cablin* from Zhaoqing, respectively), slightly outperforming the multi-head attention model, but it predicts more class 1 (the leaves of *P. cablin* from Hainan) as class 5 (the leaves of *P. cablin* from Zhaoqing) (see Fig. 3C). The CNN model has a better prediction performance than the multi-head attention and LSTM model in the Hainan and Zhaoqing spectra, see Fig. 3B. There were 296 leaves and 803 branches of *P. cablin* from Zhaoqing that were not correctly identified. There were 459 leaves and 515 branches of *P. cablin* from Hainan that were not correctly identified. The DeepPHSI model shows the best prediction performance. The confusion matrix predicted by the DeepPHSI model is shown in Fig. 3D. Its confusion matrix is symmetrical along the diagonal, and the colors on both sides of the diagonal are the "cleanest". The above comparison shows that the DeepPHSI model can overcome the impact of data imbalance on model performance. In contrast, DeepPHSI can more effectively distinguish the spectra of *P. cablin* from Hainan and Zhaoqing, whether leaves or branches.

The evaluation metrics of the CNN, LSTM, multi-head attention and DeepPHSI models on the test set are shown in Fig. 4. It can be seen that DeepPHSI achieves the best performance on the test set of the first batch of data. The accuracy of DeepPHSI is 0.9848, CNN is 0.9457, LSTM is 0.8921, and multi-head attention is 0.8577. Considering the class imbalance, the accuracy may be affected to a certain extent. The Precision, Recall and *F*1 score are used to further evaluate the predictive performance of these models. Precision represents the probability that all samples predicted as positive are actually positive samples. Recall represents the probability that samples that are actually positive are predicted as positive samples. Precision and Recall are adopted to measure the performance of the model in identifying the origins and parts of *P. cablin*. The *F*1 score takes into account both Precision and Recall, and it comprehensively balances both to achieve the highest value. DeepPHSI achieved the highest *F*1 score of 0.9834, while CNN was slightly worse at 0.9376. The LSTM and multi-head attention models achieved *F*1 scores of 0.8751 and 0.8344, respectively.

The Receiver Operating Characteristic (ROC) curves of different methods are provided in Fig. 5. As can be seen from the ROC curves, the curve of DeepPHSI completely encloses the curves of the other methods. The ROC curve of CNN is closer to that of DeepPHSI, indicating that their performances are significantly better than those of the other methods. The area included in the multi-head attention model is the smallest, indicating that its prediction effect is the worst among these models, followed by LSTM. The ROC curves intuitively reveal the relationships between the True Positive Rate (TPR) and False Positive Rate (FPR). It comprehensively evaluates the identification accuracy of the models. The results show that among these models, the DeepPHSI model has the best performance, followed by the CNN, and LSTM, and multi-head attention models in order.

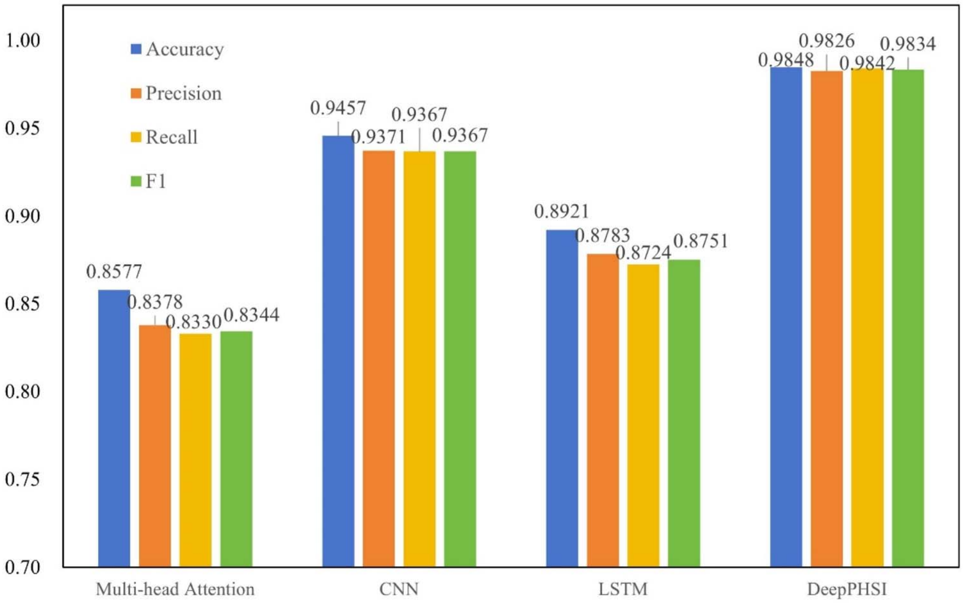

Fig. 4 Evaluation metrics of the CNN, LSTM, multi-head attention and DeepPHSI models on the test set of the first batch of data.

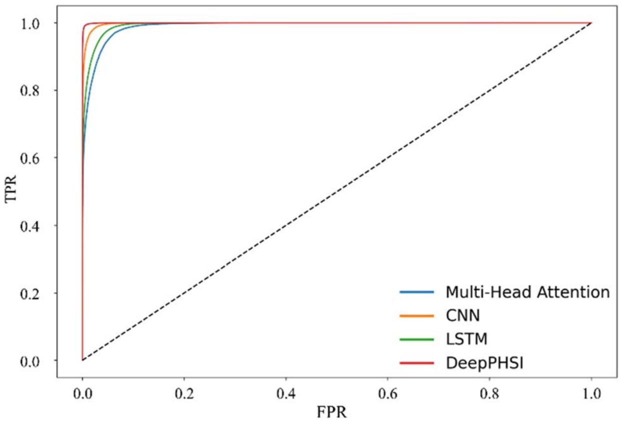

Fig. 5 Receiver operating characteristic curves of different models.

3.2 Evaluation on the first batch of HSI data

After the model was established, six sets of HSI data were used for prediction to visually demonstrate the model performance. The spectrum in each pixel of the HSI was extracted as the input of the DeepPHSI model. After pixel-by-pixel prediction, the model gives the origin, part or background of *P. cablin* represented by each spectrum. By relocating the identified category information to the original HSI, the distribution of *P. cablin* from different parts and origins on the image can be obtained. The original images of six HSI data and the corresponding prediction results of the DeepPHSI model are shown in Fig. 6. In each sub-figure of Fig. 6, the left figure represents the original image and the right figure represents the predicted result. In the prediction result figure, the white part represents the background (class 0). The red represents the leaves of *P. cablin* from Hainan (class 1), and green represents the branches of *P. cablin* from Hainan (class 2). The blue represents the leaves of *P. cablin* from Shipai (class 3), and yellow represents the branches of *P. cablin* from Shipai (class 4). Purple represents the leaves of *P. cablin* from Zhaoqing (class 5), and black represents the branches of *P. cablin* from Zhaoqing (class 6).

As can be seen in Fig. 6, DeepPHSI can accurately predict *P. cablin* from different parts and origins and can significantly distinguish it from the background. Through *in situ* reconstruction, *in situ* analysis of *P. cablin* can be achieved. DeepPHSI can achieve accurate identification, as shown in the predicted distribution image in Fig. 6A. The scattered small debris is also accurately captured and predicted to be leaves from Zhaoqing (purple).

However, there are some deviations in the identification of *P. cablin* leaves in Hainan (red) and Zhaoqing (purple), especially at the edges of the leaves. In the predicted distribution images, this is reflected as “noise” at the edge of the leaf contour.

3.3 Evaluation on the second batch of data

In the second batch of samples, *P. cablin* from different origins and parts were randomly mixed for hyperspectral data collection. When DeepPHSI is applied to these batches of data, the model needs to be fine-tuned and then used for prediction.

The prediction results of the transferred DeepPHSI model are shown in Fig. 7. The original images are shown on the left-hand side in each sub-figure. The prediction results are shown on the right-hand side in each sub-figure. In each prediction image, the white part represents the background (class 0). The red represents the leaves of *P. cablin* from Hainan (class 1), and green represents the branches of *P. cablin* from Hainan (class 2). The blue represents the leaves of *P. cablin* from Shipai (class 3), and yellow represents the branches of *P. cablin* from Shipai (class 4). Purple represents the leaves of *P. cablin* from Zhaoqing (class 5), and black represents the branches of *P. cablin* from

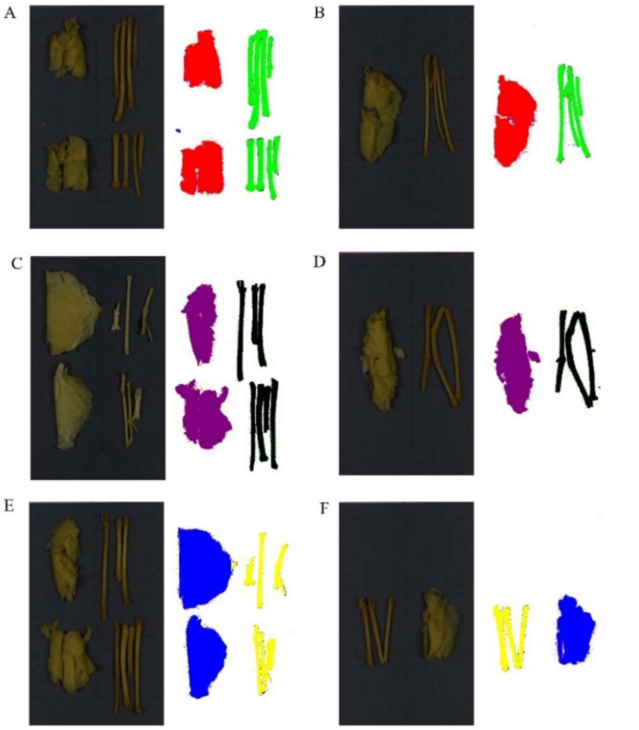

Fig. 6 Prediction results of the DeepPHSI model on the first batch of data. In each sub-figure, the left side represents the optical image of the original HSI data, and the right side represents the predicted result of the DeepPHSI model. In the prediction result figure, white represents the background (class 0). Red represents the leaves of *P. cablin* from Hainan (class 1), and green represents the branches of *P. cablin* from Hainan (class 2). Blue represents the leaves of *P. cablin* from Shipai (class 3), and yellow represents the branches of *P. cablin* from Shipai (class 4). Purple represents the leaves of *P. cablin* from Zhaoqing (class 5), and black represents the branches of *P. cablin* from Zhaoqing (class 6). (A) *P. cablin* sample from Hainan. (B) *P. cablin* sample from Hainan. (C) *P. cablin* sample from Zhaoqing. (D) *P. cablin* sample from Zhaoqing. (E) *P. cablin* sample from Shipai. (F) *P. cablin* sample from Shipai.

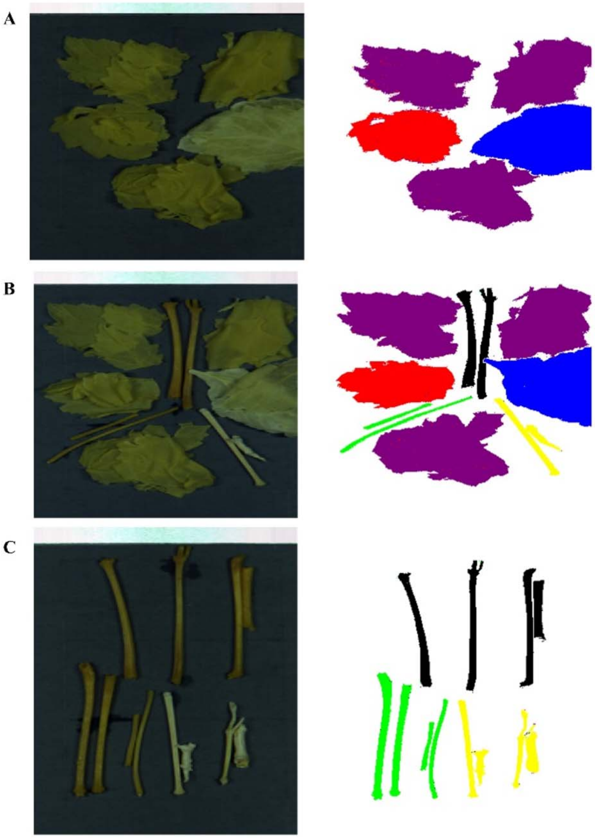

Fig. 7 Prediction results of the transferred DeepPHSI model. The original images are shown on the left side of each sub-figure. The prediction results are shown on the right side of each sub-figure. In each prediction image, white represents the background (class 0). Red represents the leaves of *P. cablin* from Hainan (class 1), and green represents the branches of *P. cablin* from Hainan (class 2). Blue represents the leaves of *P. cablin* from Shipai (class 3), and yellow represents the branches of *P. cablin* from Shipai (class 4). Purple represents the leaves of *P. cablin* from Zhaoqing (class 5), and black represents the branches of *P. cablin* from Zhaoqing (class 6). (A) Prediction results of mixed sample 1. (B) Prediction results of mixed sample 2. (C) Prediction results of mixed sample 3.

Zhaoqing (class 6). As can be seen in Fig. 7, the fine-tuned DeepPHSI model can accurately identify *P. cablin* from different origins and parts. After reconstruction, the distribution of origins and parts on the hyperspectral image can be clearly displayed.

However, by observing Fig. 7A and B, it can be found that there are some "scatter points" in the prediction image of leaves from Hainan (red) and Zhaoqing (purple). There is still room for improvement in the identification performance of the model for these two classes, which is consistent with the prediction results of the untuned model.

3.4 Cross-batch consistency verification

To validate the rationale of the "freezing CNN layers while fine-tuning LSTM and attention layers" strategy in transfer learning, a cross-batch consistency verification was performed. Specifically, 1000 samples were randomly selected from each batch, and feature vectors output by the frozen CNN layers were extracted. The cosine similarity between the mean feature vectors of the two batches was calculated, along with the intra-batch and inter-batch average Euclidean distances (Fig. 8). The

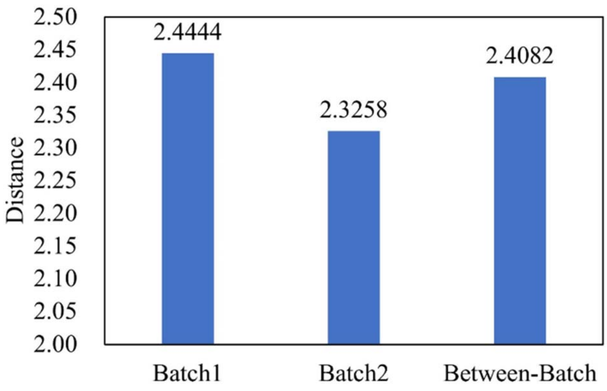

Fig. 8 Intra-batch and inter-batch average Euclidean distances of the CNN features.

cosine similarity is 0.9988 between the mean feature vectors of the two batches. The intra-batch average distances are 2.4444 (Batch 1) and 2.3258 (Batch 2). The inter-batch average distance is 2.4082. The cosine similarity of the feature vectors across batches is high. The Euclidean distances between the feature vectors between batches are similar to those within batches, and the value lies between those within two batches. These results demonstrate that the freezing strategy is reasonable and that the features are consistent across batches.

3.5 Advantages

3.5.1 Ideal performance. In this study, the DeepPHSI model realizes innovative applications in HSI data classification tasks. It integrates multi-level feature extraction and draws on the advantages of CNN, LSTM and Attention, and it can capture important features from raw data. After systematic comparison, the DeepPHSI model achieved ideal analysis results.

3.5.2 Friendly *in situ* analysis tool. The use of hyperspectral imaging to collect *P. cablin* samples can afford rich spatial–spectral information, and this process is non-destructive and fast. After pixel-by-pixel prediction by DeepPHSI, the prediction results are reconstructed to achieve *in situ* analysis of HSI. In addition, a simple graphical user interface tool is developed for ease of use and visualization. It supports visualization, prediction and result presentation of HSI data. The interface is shown in Fig. S5.

3.5.3 Transferability. In this study, transfer learning is used to address the differences between different batches of HSI data. First, the model is trained using standard data sets. Then, the pre-trained model can be fine-tuned and applied using a small amount of data from different batches of data. In practical applications, the model retraining requires a large amount of data, which usually requires tedious data collection and data labeling. In addition, model retraining takes a lot of time for optimization, and computing equipment and computing resources also play decisive roles. The strategy of transfer learning simplifies the process of data labeling and model retraining, which can save a lot of manpower and material resources.

3.5.4 Universality. In HSI data processing, complex preprocessing steps are usually considered to be necessary to improve the model's performance. However, this study shows that even without data preprocessing, the DeepPHSI model still exhibits excellent classification performance. This result shows that the DeepPHSI model can extract some key features that are helpful for classification, which may be partially lost during the preprocessing stage. In addition, modelling the analysis based on raw data (without preprocessing) avoids the impact of the selection of the preprocessing methods. In practical applications, the dependence of the model's performance on data preprocessing can be greatly reduced. To a certain extent, the universality of the model is ensured, which is in line with the trend of big data analysis.

3.6 Limitations and future work

As can be seen from the prediction results in Fig. 5 and 6, some predicted "noise" pixels tend to appear at the edge of the leaves. One possible reason is that the leaves were not completely flat and stretched during data collection. The refraction and scattering effects of light lead to differences in the spectra collected at the edge and in the center of the leaves. In addition, the standard training data used by the pre-trained model cannot strictly fit the distribution of leaves when the data are manually labeled (human eye). Spatial neighboring spectral information will be considered to solve the problem of inaccurate predictions at the edges in future work. Alternative and more accurate data labeling processes also deserve further study.

4 Conclusions

In this study, the DeepPHSI model was developed based on CNN, LSTM, and attention mechanisms. DeepPHSI predicts the pixel-by-pixel along the spatial resolution direction of the HSI, and it identifies the origin and anatomical part information of *P. cablin*. Two batches of HSI data of *P. cablin* were collected for model training, validation and testing. These *P. cablin* originated from Hainan, Zhaoqing and Shipai, including leaves and branches. Compared with the traditional CNN, LSTM and Attention models, DeepPHSI exhibited better identification performance. Different preprocessing methods were performed on the same test dataset, and the results showed that the pre-trained model exhibited high accuracy without data preprocessing. As for transfer learning, the performances of the transferred models with fine-tuning applied at different layers are also compared to select the optimal transfer strategy. Cross-batch consistency verification results demonstrated that the freezing strategy is reasonable and the features are consistent across batches. With pre-training and fine-tuning strategies, the prediction results of the DeepPHSI model on different datasets were obtained. The results showed that the model could accurately identify the origins and parts of *P. cablin* in the HSI. Although there is minor prediction "noise" at the edge of the leaves, the visualization results are still of great reference significance. Combined with transfer learning, the DeepPHSI method is expected to be applied to quality control and the online detection of *P. cablin*, and it is a transferable, universal and highly promising method.

Author contributions

Xiaqiong Fan: validation, writing – review and editing, funding acquisition. Yulin Liu: writing – original draft, visualization. Zihao Zhang: software, visualization. Peijun Zhao: visualization. Zhengyan Li: validation. Junjun Zhou: data curation. Dandan Zhai: data curation. Yi Hu: validation. Peng Li: methodology, funding acquisition. Hongchao Ji: funding acquisition, methodology.

Conflicts of interest

There are no conflicts to declare.

Data availability

Data will be made available on request.

Supplementary information: Text S1: data preprocessing and augmentation. Fig. S1: some considered network architectures. Fig. S2: learning curves of the considered architectures. Fig. S3: the data number of each class after augmentation. Fig. S4: some considered network architectures for comparison. Fig. S5: a screenshot of the graphical user interface tool. Table S1: accuracy of the model on the test set of first batch of data with different preprocessing methods. Table S2: accuracy of fine-tune different layers on the test set of second batch of data. See DOI: https://doi.org/10.1039/d5ra06579h.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (grant no. 2023YFA0915800); Natural Science Project of Science and Technology Department of Henan Province (grant no. 252102211033); high-level Talents Fund of Henan University of Technology (grant no. 2022BS075); Agricultural Science and Technology Innovation Program (grant no. CAAS-ZDRW202503); and Open Research Topics of Henan University of Technology (grant no. CSKFJJ-2024-18).

References

1 C. Junren, X. Xiaofang, L. Mengting, X. Qiuyun, L. Gangmin, Z. Huiqiong, C. Guanru, X. Xin, Y. Yanpeng and P. Fu, *Chin. Med.*, 2021, **16**, 1–20.

2 Y. Chen, Q. Luo, S. Li, C. Li, S. Liao, X. Yang, R. Zhou, Y. Zhu, L. Teng and H. Chen, *J. Ethnopharmacol.*, 2020, **259**, 113009.

3 X. Chen, J. Li, X. Wang, L. Zhong, Y. Tang, X. Zhou, Y. Liu, R. Zhan, H. Zheng and W. Chen, *BMC Plant Biol.*, 2019, **19**, 1–18.

4 M. Miyazawa, Y. Okuno, S.-i. Nakamura and H. Kosaka, *J. Agric. Food Chem.*, 2000, **48**, 642–647.

5 B. Xie, X.-F. Wu, H.-T. Luo, X.-L. Huang, F. Huang, Q.-Y. Zhang, X. Zhou and H.-Q. Wu, *J. Pharm. Biomed. Anal.*, 2022, **209**, 114526.

6 F. Liu, W. Cao, C. Deng, Z. Wu, G. Zeng and Y. Zhou, *Chem. Cent. J.*, 2016, **10**, 1–11.

7 M. K. Swamy and U. R. Sinniah, *Molecules*, 2015, **20**, 8521–8547.

8 H. Guozhen, L. Jinkun, G. Wei, S. Jing and C. Weiwen, *Chin. Agric. Sci. Bull.*, 2012, **28**, 288–292.

9 T. A. van Beek and D. Joulain, *Flavour Fragrance J.*, 2018, **33**, 6–51.

10 G. Hu, C. Peng, X. Xie, S. Zhang and X. Cao, *J. Evidence-Based Complementary Altern. Med.*, 2017, **2017**, 4850612.

11 Y. Li, J. He, H. Liu, Y. Zhang and Z. Li, *Expert Syst. Appl.*, 2025, **260**, 125453.

12 Y. Chu, J. Cao, W. Ding, J. Huang, H. Ju, H. Cao and G. Liu, *Inf. Sci.*, 2025, **689**, 121504.

13 Y. T. Chan, S. J. Wang and C. H. Tsai, *Inform. Fusion*, 2017, **39**, 154–167.

14 S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre and C. A. Mücher, *Remote Sens. Environ.*, 2012, **126**, 222–231.

15 J. Pontius, M. Martin, L. Plourde and R. Hallett, *Remote Sens. Environ.*, 2008, **112**, 2665–2676.

16 T. Akgun, Y. Altunbasak and R. M. Mersereau, *IEEE Trans. Image Process.*, 2005, **14**, 1860–1875.

17 D. Wang, Y. Tan, C. Li, J. Xin, Y. Wang, H. Hou, L. Gao, C. Zhong, J. Pan and Z. Li, *Soil Tillage Res.*, 2025, **248**, 106397.

18 Z. Xing, G. Ma, L. Wang, L. Yang, X. Guo and S. Chen, *IEEE Internet Things J.*, 2025, **12**, 21328–21338.

19 Z. Xing, Z. Meng, G. Zheng, G. Ma, L. Yang, X. Guo, L. Tan, Y. Jiang and H. Wu, *Front. Comput. Neurosci.*, 2025, **19**, 1543643.

20 S. Gamage, S. Manna, M. Zajac, S. Hancock, Q. Wang, S. Singh, M. Ghafariasl, K. Yao, T. E. Tiwald and T. J. Park, *ACS Nano*, 2024, **18**, 2105–2116.

21 G. Zhao, Z. Xu, L. Tang, X. Li, Z. Dai, Z. Xie, Y. Jiang, Y. Wu, P. Zhang and Q. Wang, *J. Food Compos. Anal.*, 2025, **138**, 107028.

22 J. Li, X. Huang, P. Gamba, J. M. B. Bioucas-Dias, L. Zhang, J. A. Benediktsson and A. Plaza, *IEEE Trans. Geosci. Remote Sens.*, 2015, **53**, 1592–1606.

23 X. Sun, N. M. Nasrabadi and T. D. Tran, *IEEE Trans. Geosci. Remote Sens.*, 2015, **53**, 4457–4471.

24 F. Leyuan, Z. Haijie and L. Shutao, *Neurocomputing*, 2017, **273**, 171–177.

25 F. Luo, B. Du, L. Zhang, L. Zhang and D. Tao, *IEEE Trans. Cybern.*, 2019, **49**, 2406–2419.

26 H. Huang, Z. Liu, C. L. P. Chen and Y. Zhang, *Appl. Intell.*, 2022, **53**, 15683–15694.

27 D. Liu, J. Li, Q. Yuan, L. Zheng, J. He, S. Zhao and Y. Xiao, *Inform. Fusion*, 2023, **94**, 92–111.

28 Y. Chen, H. Zhang, Y. Wang, Y. Yang and J. Wu, *IEEE Trans. Image Process.*, 2024, **33**, 1211–1226.

29 A. Kartakoullis, N. Caporaso, M. B. Whitworth and I. D. Fisk, *Chemom. Intell. Lab. Syst.*, 2025, **259**, 105341.

30 Y. Lecun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436–444.

31 J. Schmidhuber, *Neural Netw.*, 2015, **61**, 85–117.

32 Y. Wu, X. Han, Y. Su, M. Glidewell, J. S. Daniels, J. Liu, T. Sengupta, I. Rey-Suarez, R. Fischer and A. Patel, *Nature*, 2021, **600**, 279–284.

33 Y. Tom, H. Devamanyu, P. Soujanya and C. Erik, *IEEE Comput. Intell. Mag.*, 2018, **13**, 55–75.

34 G. Hinton, L. Deng, D. Yu, G. E. Dahl and B. Kingsbury, *IEEE Signal Process. Mag.*, 2012, **29**, 82–97.

35 D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. Mckeown, G. Yang, X. Wu and F. Yan, *Cell*, 2018, **172**, 1122–1131.

36 G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio and C. I. Sánchez, *Med. Image Anal.*, 2017, **42**, 60–88.

37 Z. Wang, Y. Peng, J. Li, J. Li, H. Yuan, S. Yang, X. Ding, A. Xie, J. Zhang and S. Wang, *Plant Commun.*, 2024, **5**, 100985.

38 Y. Ren, C. Wu, H. Zhou, X. Hu and Z. Miao, *Plant Commun.*, 2024, **5**, 101002.

39 J. Li, D. Zhang, F. Yang, Q. Zhang, S. Pan, X. Zhao, Q. Zhang, Y. Han, J. Yang and K. Wang, *Plant Commun.*, 2024, **5**, 100975.

40 A. Krizhevsky, I. Sutskever and G. E. Hinton, *Adv. Neural Inf. Process.*, 2012, **25**, 1097–1105.

41 S. Yu, S. Jia and C. Xu, *Neurocomputing*, 2016, **219**, 88–98.

42 Q. Gao, S. Lim and X. Jia, *Remote Sens.*, 2018, **10**, 299.

43 M. Zhang, W. Li and Q. Du, *IEEE Trans. Image Process.*, 2018, **27**, 2623–2634.

44 Y. Liu, X. Wang and L. L. Jiang, *Pattern Recognit. Lett.*, 2024, **179**, 1–8.

45 U. Patel and V. Patel, *J. Supercomput.*, 2024, **80**, 2461–2486.

46 J. Sun, J. Yang, W. Chen, S. Ding, S. Li and J. Hu, *Neural Process. Lett.*, 2024, **56**, 181.

47 J. Yang, B. Du, D. Wang and L. Zhang, *IEEE Trans. Image Process.*, 2024, **33**, 257–272.

48 X. Glorot, A. Bordes and Y. Bengio, 2011.

49 S. Hochreiter and J. Schmidhuber, *Neural Comput.*, 1997, **9**, 1735–1780.

50 I. Loshchilov and F. Hutter, Decoupled weight decay regularization, *arXiv*, 2017, preprint, arXiv:1711.05101, DOI: 10.48550/arXiv.1711.0510.