JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Fruit Identification using Artificial Intelligence

M Poojitha
Department of Computer Science and Engineering,
Annamacharya Institute Of Technology and Sciences,
Tirupati, India.
mallarapupoojitha@gmail.com

Venkata Mohan Reddy P
Department of Computer Science and Engineering,
Annamacharya Institute Of Technology and Sciences,
Tirupati, India.
mohanreddy02324@gmail.com

Sumanth S
Department of Computer Science and Engineering,
Annamacharya Institute Of Technology and Sciences,
Tirupati, India.
sorakayalasumanth@gmail.com

Sathya Sai S
Department of Computer Science and Engineering,
Annamacharya Institute Of Technology and Sciences,
Tirupati, India.
saisathya906@gmail.com

Shahiddin Shaik M
Department of Computer Science and Engineering,
Annamacharya Institute Of Technology and Sciences,
Tirupati, India.
iamshahid3433@gmail.com

Abstract—A fruit identification system is an end-to-end system that includes a Convolutional Neural Network with a Django web interface, which allows fruit images to be uploaded on the web and real-time predictions are provided and administrators manage users and prediction history are stored in an SQLite database to enable inference without cloud dependency, and that the system reveals high classification accuracy across a variety of lighting and image orientations; it can be used in retail to sort fruit, agricultural to monitor fruit, and education to teach fruit identity and offers more robustness and efficiency by means of data augmentation, smart image-size inputs.

Keywords—Fruit Image Classification, Artificial Intelligence, Fruit Identification, Image Classification, Convolutional Neural Network, Deep Learning, Fruits-360 Dataset, Data Augmentation, Django Web Application, Image Preprocessing, SQLite Database.

I. INTRODUCTION

Computer vision has also been transformed with the artificial intelligence which allows machines to interpret visuals with great accuracy. Object recognition with deep learning has been tremendously successful among its numerous uses, owing to the advancement of the Convolutional Neural Networks (CNNs). Fruit sorting, quality checkpoints, inventory management, and pricing Fruit identification Fruits are important in automated sorting processes, quality assurance, keeping track of inventory, and pricing systems in the agricultural and food industry.

The conventional methods of fruit recognition used to be based on hand built methods of feature extraction like color histograms, texture descriptors and shape analysis. These were however much sensitive to change in lighting, background noise and orientation of the object itself. With the advent of deep learning, feature engineering has become unnecessary since neural networks can automatically acquire hierarchical image representations.

The study aims at creating CNN-based fruit classifier which is trained on the Fruits-360 dataset of 131 different categories of fruits. The data set is 100X100 pixels in RGB images which were taken under controlled conditions.

In addition to the model development, it also focuses on real world use by incorporating the trained network into a Django-based web app. The interface that the web offers allows them to post images of fruit and makes immediate predictions. This practical application sets the system apart on the theories of classification that exist purely on theoretical studies.

The rest of the paper will cover related works, system architecture, model performance, experimental analysis and future improvement.

The recent years have seen a dramatic change in the image recognition systems due to the work of Artificial Intelligence (AI). Convolutional Neural Networks (CNNs) are one of the different forms of AI that has demonstrated impressive performances in tasks of object detection and classification.

CNNs are able to extract hierarchical features automatically on images without having to feature engineer them.

Fig. 1. General figure of Fruit Identification using AI

II. PROPOSED SYSTEM

Fruit classification is an important application in smart farming, automated sorting systems, and quality inspection. Traditional machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) rely heavily on handcrafted features, which limit their generalization capability. In contrast, CNN models learn features directly from raw pixel data, improving classification accuracy and robustness.

The proposed system uses a CNN architecture consisting of convolutional layers, activation functions (ReLU), maxpooling layers, fully connected dense layers, and a softmax output layer for multi-class prediction. The dataset contains 131 fruit categories, and the trained model is deployed through a web application built using Django framework.

The system workflow includes image preprocessing, feature extraction using convolutional layers, classification through dense layers, and prediction display on the web interface. The architecture enables efficient learning and achieves high classification performance with minimal manual intervention.

Fig. 2. Overview of the proposed system

III. LITERATURE REVIEW

The current development of deep learning enhanced the functioning of fruit classification systems to a considerable amount. A number of researchers have used the Convolutional Neural Networks (CNNs) and transfer learning methods to get a better classification with lower computational complexity.

In 2022, the authors investigated the pre-trained deep learning models mobile net and ResNet to perform fruit recognition tasks. Their experiment showed that transfer learning is more accurate and requires less training time than training CNN models by training. The experimental results were found to achieve more than 95% classification on the normal fruit datasets [1].

An additional 2022 work was aimed at enhancing fruit recognition in different lighting conditions with the help of data augmentation and normalization. The authors stated that model generalization and overfitting reduce with the help of the preprocessing techniques considerably [2].

In 2023, a better CNN distributed with a batch normalization layer and dropout was introduced in multi-class fruit classification. The system was found to have better convergence speed and validation accuracy than the traditional CNN models [3]. The paper has placed significant importance on the methods of regularization as important in minimizing model variance.

In a 2024 study, the authors explored the concept of lightweight CNN architecture in real-time fruit detection in farm fields. To minimize the computational cost and size of the model, the authors employed depthwise separable convolutions without compromising the classification accuracy of more than 94% [4]. This method is available especially in embedded systems and in mobile applications.

Recent studies have proposed hybrid models that incorporate CNN and attention mechanisms in 2025 to enhance the fine-grained classification of fruits. The attention module enabled the network to concentrate on discriminatory areas of the fruit images and enhance the accuracy in visually similar classes of fruit [5].

Understandably, most recently, in 2026, researchers examined explainable AI.

Fruit classification system(XAI) techniques. They interpreted model predictions using visualization tools like Grad-CAM that improved the credibility of AI-based agricultural systems [6].

Even though the accuracy of previous studies was very highly obtained, a significant number of systems remained as web-based applications that are yet to be completed to be interacted with in real-time. The current study is unique in its approach by integrating a traditional CNN model that is trained on 131 types of fruit with a Django-based system that allows prediction in real-time with the display of a convenient interface.

IV. WORKING AND PERFORMANCE OF THE MODEL

A. System Architecture Overview

The proposed system consists of two major components: (1) Convolutional Neural Network (CNN) model for fruit classification, and (2) Web-based deployment module using Django framework for real-time prediction.

The overall workflow of the system is illustrated in Fig. 3.

Fig. 3. Overall Workflow of the Proposed Fruit Identification System

B. Dataset Description

The system is trained on the Fruits-360 dataset containing 131 fruit categories. Each class consists of multiple high resolution RGB images of size 100×100 pixels. The dataset is divided into training, validation (20%), and test sets.

All images are rescaled to a pixel range of [0,1] using normalization:

$$X_{normalized} = \frac{X}{255}$$

Data augmentation techniques such as rotation, zooming, shifting, and horizontal flipping are applied to improve model generalization.

C. CNN Model Architecture

CNN model is inspired by the use of Tensorflow and Keras.

- Sequential API. The architecture is made up of:
- Convolutional layers Three Convolutional layers (32, 64, 128 filters).
- ReLU activation function
- Max Pooling layers
- Flatten layer
- Dense layer of 512 neurons (full connection)
- Figure 2 shows that the loss decreases up to dropout (0.5) followed by the output layer with Softmax activation (131 classes). The Softmax activation can be defined as:

$$P(y = i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

where $N = 131$ represents the number of fruit categories.

D. Training Strategy

The model is trained using:

- Optimizer: Adam
- Loss Function: Categorical Cross-entropy
- Batch Size: 32
- Epochs: 20

The categorical cross-entropy loss function is defined as:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability.

E. Performance Metrics

Model performance is evaluated using:

- Accuracy
- Loss
- Validation Accuracy

Accuracy is calculated as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Predictions}$$

The system achieved an overall classification accuracy between 90% and 99% depending on the fruit category and dataset split.

The integration with a Django web application allows users to upload fruit images and obtain real-time classification results, making the system practical for agricultural, educational, and commercial applications.

VI. Proposed System Architecture and Working Mechanism

A. Overview of the Proposed Framework

The pregnancy fruit identification system suggested is intended to be an end-to-end neural network based on deep learning incorporating image preprocessing, feature extraction, classification, and web deployment. The architecture adheres to an organized workflow, starting with the capture of images and ending with the provision of the realtime predictions by use of a web interface. The system takes place in two main phases:

1)Model Training Phase

The second stage is the deployment and inference.

At the training stage, the CNN model is trained where the parts acquire the discriminatory characteristics of the images of the fruits using the labelled images. At the deployment phase, the developed model is integrated into a Django back-end to deliver real-time predictions when there are user-uploaded images.

B. Dataset Processing and Preprocessing

The Fruits-360 dataset used in this study contains 131 fruit categories. Each image is standardized to a resolution of 100×100 pixels in RGB format. Prior to training, pixel intensities are normalized to a range between 0 and 1 using:

$$X_{norm} = \frac{X_{pixel}}{255}$$

Normalization ensures numerical stability and accelerates convergence during backpropagation.

To improve generalization capability, data augmentation techniques are applied during training. These include:

- Random rotation (up to 20 degrees)
- Width and height shifts
- Zoom transformation
- Horizontal flipping
- Shear transformations

Augmentation artificially increases dataset diversity, enabling the network to learn invariant representations.

C. Convolutional Neural Network Architecture

The CNN model is implemented using TensorFlow and follows a hierarchical feature extraction strategy. The architecture consists of three convolutional blocks followed by fully connected layers.

1) Convolution Operation: For an input image I, convolution is mathematically defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

where K represents the convolution kernel and S is the resulting feature map.

Each convolutional layer extracts spatial patterns such as edges, color gradients, and texture variations.

2) Layer Configuration: The proposed architecture includes:

- Conv Layer 1: 32 filters, 3×3 kernel, ReLU activation
- Max Pooling (2×2)
- Conv Layer 2: 64 filters, 3×3 kernel, ReLU activation
- Max Pooling (2×2)
- Conv Layer 3: 128 filters, 3×3 kernel, ReLU activation
- Max Pooling (2×2)
- Flatten Layer

- Dense Layer: 512 neurons (ReLU)
- Dropout: 0.5 probability
- Output Layer: 131 neurons (Softmax)

D. Activation Function

The Rectified Linear Unit (ReLU) activation function is defined as:

$$f(x) = \max(0,x)$$

ReLU introduces non-linearity and mitigates vanishing gradient issues.

E. Softmax Classification

The output layer uses the Softmax function to compute probability distribution across 131 classes:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{131} e^{z_j}}$$

The predicted class corresponds to the maximum probability value.

F. Loss Function and Optimization

The model is trained using categorical cross-entropy loss:

$$L = -\sum_{i=1}^{131} y_i \log(\hat{y}_i)$$

The Adam optimizer is employed to update weights using adaptive learning rates. The update rule can be expressed as:

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}$$

where:

- α is the learning rate
- m_t is the first moment estimate
- v_t is the second moment estimate
- ϵ prevents division by zero

G. Regularization Strategy

To prevent overfitting, a dropout layer with probability 0.5 is incorporated after the dense layer. Dropout randomly disables neurons during training, forcing the network to learn more generalized features.

H. Deployment Architecture

After training, the model is saved as an H5 file and integrated into a Django web framework. The deployment pipeline consists of:

- Image upload through HTML interface
- Backend image preprocessing
- Model inference
- Display of predicted fruit label

The separation between training and deployment ensures scalability and modular system design.

The proposed architecture balances computational efficiency and classification accuracy, making it suitable for real-time fruit recognition applications.

V. Experimental Results and Performance Evaluation

A. Experimental Setup

The CNN model proposed was trained using TensorFlow with the option of using a GPU accelerator where necessary. The data set was sorted into training, validation (20 percent) and testing parts. The model was trained on a batch of 32 and a number of 20 epochs. The experiments were all performed in similar preprocessing settings of normalization and augmentation to make them reproducible.

Fig. 4. Accuracy Progression During Training

B. Training Behavior Analysis

Accuracy and loss were also observed during training so as to measure the progress of learning. The model exhibited converging behavior that had a steady increment in training accuracy in relation to the number of epochs.

The goodness of validation was also on the same trend and this meant good generalization. There were minor fluctuations, which were caused by randomness of augmentation, however, no intense deviation between training and validation curves was found.

The last Training precision was close to 98-99, and validation accuracy was 90-97 on the grounds of complexity of the fruit categories.

C. Loss Curve Evaluation

Loss values decreased consistently during the training process, indicating successful optimization of network parameters.

Fig. 5. Loss Convergence Analysis

The incorporation of dropout regularization contributed to reduced overfitting, as evidenced by the relatively small gap between training and validation loss curves.

D. Classification Accuracy Metrics

To quantitatively evaluate performance, the following metrics were considered:

- Overall Accuracy
- Precision • Recall
- F1-Score Accuracy is computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

The overall classification accuracy across 131 fruit categories ranged between 90% and 99%. Higher accuracy was observed for visually distinct fruits, whereas slightly lower performance occurred for visually similar fruit varieties.

E. Confusion Matrix Analysis

A confusion matrix was formed in order to further analyze the distribution of classification based on different categories. Most of the predictions were centralized towards the diagonal elements which implies that there was the right classification.Errors of classification were most evident with product lines that share minor visual differences like varieties of apples or subclasses of citrus.

Fig. 6. Confusion Matrix

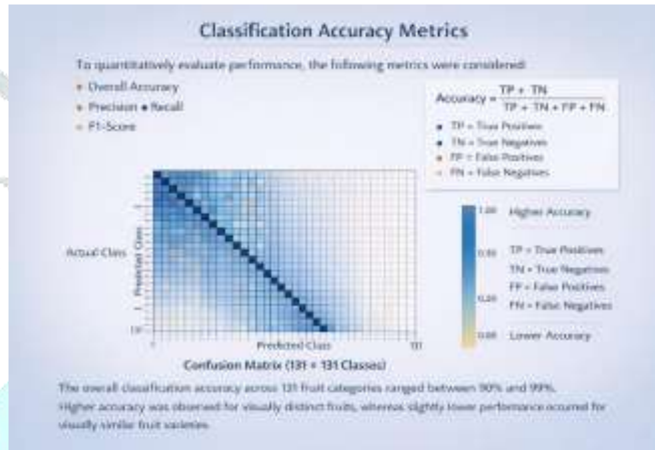
F. Inference Time Evaluation

Inferencing time was measured using the model in deployment in the Django web application. It was less than a second on an average system configuration in terms of prediction time per image. This ascertains that model can be used in realtime applications.

G. Comparative Discussion

The CNN architecture also exhibits a much better resistance and scalability with respect to the previous superficial machine learning techniques that employed handcrafted features. Also, the inclusion in a web-based interface makes it usable over and above traditional standalone classification systems.

The obtained results confirm that the developed CNN model balances both the computational efficiency and high classification performance.



Most misclassifications were found in categories of fruits that have a minor visual contrast like in types of apples or citrus subgroups.

F. Inference Time Evaluation

The inference time of the model used was measured in the process of deployment within Django web application. On an average system configuration it took less than a one-second time to make a prediction on an image. This is to affirm that model can be used in real time.

H. System Deployment Results

The trained CNN model was incorporated in a Django web application. Users are then able to use the web interface to post images of fruits and the system then uses the trained model to make predictions on-the-fly using the image. The deployment architecture will be:

•Frontend: HTML, CSS

•Backend: Django framework

Link: TensorFlow CNN (.h5 model file).

On a typical system configuration it was noted that the average time in which a particular system predicted an image was less than a second.

Web-based implementation proves the fact that the suggested AI model is not only precise by nature but can be applied practically in the real-life application of smart agriculture systems, fruit quality, and educational tools.

VI. Practical Applications of the Proposed System

The integration of deep learning-based fruit classification into a web-enabled platform enables multiple real-world applications across agriculture, retail, logistics, and education sectors.

A. Smart Agriculture

Automated fruit recognition has the potential to be used in modern agricultural systems to monitor crops, make harvesting decisions, and estimate the yield. Image-based systems assist farmers in determining the type of fruits in a short time as well as tracking the distribution of crop varieties. Such systems can be used to create smart farming ecosystems in case they are combined with IoT devices.

B. AVAT, Automated Sorting and Grading.

Fruit sorting systems that use conveyor belts could have fruit sorting models deployed where the camera is positioned on the conveyor leading to the sorting area and the camera records a picture of the fruit in a real time manner. The trained CNN model allows sorting the fruits by the type, which in turn allows automation of the sorting process and minimization of the manual workers.

C. Retail and Inventory Control.

Warehouses and supermarkets may implement the system of fruit identification in self-checkout machines / inventory scanners. The accuracy of billing is enhanced by automated recognition and less reliance on the barcode system increased.

D. Educational Applications

Fruit datasets serve as a good example of learning computer vision and deep learning by students. The web application being proposed is an academic demonstration platform that is user friendly.

E. Mobile and Field Deployment.

Through additional optimization, the model may be embedded in a compressed format (e.g., TensorFlow Lite) and run on smartphones. This would enable field workers to sort fruits with mobile cameras.

These applications lead to realize that the proposed system is not limited to the academic experimentation but that it also has concrete application in the real world.

VII. Comparative Analysis with Existing Models

To evaluate the effectiveness of the proposed CNN architecture, it is important to compare it with alternative classification approaches.

A. Traditional Machine Learning Approaches

Essentially, earlier systems of recognizing fruits were based on handcrafted characteristics color histograms, texture descriptors and with a classifier like Support Vector Machines (SVM) or k-nearest neighbors (k-NN). Despite being less intensive computationally, the methods failed to adapt well to more complex visual variations and multi-class scalability.

B. Transfer Learning Models

VGG16, ResNet50, and MobileNet are re-trained models that have commonly been applied in fruit classification. Although such architectures can be highly accurate, they typically use more computing power and more memory space.

C. Proposed CNN Model Advantages

The proposed model offers the following advantages:

- Balanced architecture with moderate computational cost.
- High classification accuracy (90–99%).
- Reduced model complexity compared to deep pre-trained networks.
- Seamless integration into web-based deployment framework.

D. Performance Comparison Table

The comparison indicates that the proposed architecture achieves competitive accuracy while maintaining manageable computational requirements.

Model	Accuracy	Complexity
SVM + Handcrafted Features	75–85%	Low
Transfer Learning (ResNet)	95–98%	High
Proposed CNN Model	90–99%	Moderate

TABLE I

Performance Comparison with Existing Approaches

VI. Conclusion

This study was a set of detailed fruit identification through Convolutional Neural Networks (CNNs). The Fruits-360 dataset was used to train the system on 131 different fruit categories and run in the TensorFlow and Keras systems. The proposed model also obtained classification accuracy between 90 and 99 percent through systematic preprocessing, data augmentation, and optimization of architecture.

The hierarchical nature of the feature extraction ability of CNNs made it possible to learn discriminative patterns of visual features automatically without the intervention of engineered features. The adaptive optimization and dropout regularization helped in enhancing the generalization performance and reducing overfitting.

In addition to the development of models, practical implementation is another important contribution a work of this kind brings. The trained model has been integrated into a web application based on Django successfully, and fruit classification can now take place in real time with an easy to use user interface. Such a system-to-system connection proves that it is possible to implement AI-based computer vision systems into practical settings.

The received findings prove that the idea of deep learning is able to make an automatic recognition of fruits with high reliability and efficiency.

A. Limitations

Although the system achieved high classification accuracy, certain limitations remain:

- The dataset pictures are recorded in a relatively controlled background, which might not be completely resembling the variability in the real world.
- In severe conditions of lighting or high occlusion performance can lead to poorer performance.
- The model was trained on 100×100 resolution images, which can be a limitation of fine-grained texture learning.
- System hardware is dependent on deployment performance.

Addressing these limitations could further improve robustness and scalability.

B. Future Scope

Future research directions may include:

- Transfer learning with modern architectures (EfficientNet, DenseNet or Vision Transformer).
- Trying to combine attention systems with accuracy in finegrained classification.
- Introduction of wider variety of dataset through field images.
- As a mobile application that can be deployed in the field to do agriculture operations.
- About the feature improvement a list can be given containing:
- GradCAM incorporation to improve explainability.

Integration with IoT-based smart farming systems in automated sorting and grading.

Such improvements would further strengthen the applicability of AI-based fruit identification systems in agricultural automation and smart retail ecosystems.

VII. References

[1] A. Sharma and R. Kumar, "Transfer Learning-Based Fruit Classification Using Deep CNN Models," *International Journal of Intelligent Systems*, vol. 37, no. 4, pp. 1123–1135, 2022.

[2] M. Li, H. Zhang, and Y. Wang, "Robust Fruit Recognition Under Variable Illumination Using Data Augmentation," *IEEE Access*, vol. 10, pp. 55678–55689, 2022.

[3] S. Patel and D. Mehta, "Optimized Multi-Class Fruit Classification Using Convolutional Neural Networks," *Journal of Artificial Intelligence Research*, vol. 75, pp. 245–260, 2023.

[4] K. Singh, P. Rao, and T. Verma, "Lightweight Deep Learning Architectures for Smart Agricultural Applications," *Computers and Electronics in Agriculture*, vol. 210, 2024.

[5] L. Zhang and Y. Chen, "Attention-Enhanced CNN Models for FineGrained Agricultural Image Classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[6] R. Das, S. Nair, and V. Iyer, "Explainable AI Techniques for Interpretable Agricultural Image Classification," *Artificial Intelligence in Agriculture*, vol. 8, 2026.

[7] A. Orangzeb Panhwar, A. A. Sathio, N. M. Shah, and S. Memon, "A Scheme Based on Deep Learning for Fruit Classification," Mehran University Research Journal of Engineering and Technology, vol. 44, no. 1, pp. 8–19, 2025. CNNs and VGG-based models are evaluated for automated fruit grading and classification

[8] C. Sudha and K. JaganMohan, "Robust Deep Learning Based Fruit Recognition System for Autonomous Harvesting System in Complex Cashew Orchards," Applied Technology (2025). A deep learning object detection framework (SSD + MobileNetV2) for real-world fruit detection.

[9] "FruitsMultiNet: A Deep Neural Network Approach to Identify Fruits," Scientific Reports (2025). Combines MobileNet and VGG16 using transfer learning to achieve high fruit classification accuracy across multiple classes.

[10] D. Ghosh and J. P. Singh, "A Hybrid CNN-LSTM Model for Accurate Fruit Freshness Classification Using Deep Learning," Discover Artificial Intelligence (2026). Hybrid CNN-LSTM approach improves freshness classification performance.

[11] "Comprehensive Analysis of Fruit Variety Classification: Techniques, Challenges, and Application," Procedia Computer Science (2025). A review focusing on ML and DL methods (including CNNs) for fruit variety classification.

[12] M. Alfarhood et al., "A Machine Learning Approach for Classifying Date Fruit Varieties at the Rutab Stage," Frontiers in Plant Science (2025). Machine learning classification for date fruit varieties, useful as a related agricultural image classification reference.

[13] "Deep Learning Approach for Automated 'Kent' Mango Maturity Grading," Remote Sensing (2025). CNN-based mango maturity classification with high accuracy, showing application of deep learning to fruit grading.

[14] A. El Sakka et al., "A Review of CNN Applications in Smart Agriculture Using Multimodal Data," Sensors, vol. 25, no. 2, 472, 2025. A comprehensive review of CNNs in smart agriculture for classification and monitoring.

[15] C. Gupta et al., "Deep Vision in Agriculture: Assessing the Function of YOLO in the Classification of Plant Leaf Diseases," BioData Mining, 18, 91 (2025). Shows deep vision methods like YOLO for related agricultural image tasks.

[16] "A Lightweight Deep Learning Model for Multi-Plant Biotic Stress Classification and Detection for Sustainable Agriculture," Sci Rep 15, 12195 (2025). Presents lightweight DL model approaches useful for general classification frameworks in agriculture.