Deep Learning Approaches for Automated Disease Detection in Agriculture

Ahmed A. F. Osman[1,*], Rajit Nair[2], Mosleh Hmoud Al-Adhaileh[3], Theyazn H.H Aldhyani[1], Saad M. AbdelRahman[1], Sami A. Morsi[1]

[1]Applied College, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia

[2]VIT Bhopal University, Bhopal, India

[3]Deanship of E-Learning and Distance Education and Information Technology, King Faisal University, P.O. Box 4000, Al-Ahsa 31982, Saudi Arabia

Emails: afadol@kfu.edu.sa; rajit.nair@vitbhopal.ac.in; madaileh@kfu.edu.sa; taldhyani@kfu.edu.sa; smaahmed@kfu.edu.sa; Smorsi@kfu.edu.sa

Abstract

This research introduces a cutting-edge deep learning-based agricultural engineering illness diagnosis approach. Convolutional neural networks (CNNs) and improved methods improve accuracy and efficiency. The recommended solution includes network settings, convolution processes, and sharing strategies to reduce dimensions. These methods reduce the network's processing power so it can concentrate on disease characteristics. The model employs dropout regularization, attention processes, and multi-scale feature extraction to enhance sickness prediction. The technology also utilizes photographs and sensor data to adapt to agricultural circumstances. The performance test shows that the suggested technique outperforms traditional machine learning and mixed models in F1 score (95%), accuracy (95%), precision (94%), memory (96%), and correctness (94%). It has high discriminative power with an AUC-ROC score of 0.98. The model uses computers well: two hours to train, two seconds to derive conclusions, and 65% of the CPU at all times. Real-time farming could benefit from its use. The suggested technique can properly and reliably diagnose illnesses due to its low overfitting rate and excellent generalization potential. The precision agriculture technique will enhance crop health management and productivity.

Keywords: Agricultural biotechnology; Convolutional neural networks; Disease detection; Dropout regularization; Feature extraction; Image processing; Multi-modal data; Precision agriculture; Resource utilization; Real-time deployment

1. Introduction

Agricultural production is one of the most important factors to consider while attempting maintaining global food security. Plant diseases, on the other hand, are a major issue since they lower agricultural productivity and limit the variety of economic prospects [1]. Recent developments in deep learning have transformed agricultural engineering, particularly in disease diagnostics, by enabling automated systems that are both accurate and efficient. We employ many complicated structures to quickly discover patterns in still images [2]. This category covers architectures like ResNets, CNNs, capsule networks, and long short-term memory (LSTM) networks. This allows for the essential early diagnosis of agricultural illnesses, which is required for their prevention. Incorporating transfer learning might improve the model's performance even further [3. The model maintains a high level of accuracy despite requiring very little data. Attention approaches also improve feature extraction, allowing models to concentrate on the most essential portions of input pictures. Advances in feature extraction make this possible [4].

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

Autoencoders and deep belief networks (DBNs) were among the earliest approaches used in deep learning. Despite the promising future of these technologies, issues such as accuracy and memory constraints may still arise. Today's frameworks can use new technologies like spatial attention modules, multi-scale feature extraction, and dropout regularization to make performance measures like F1 score, accuracy, precision, and recall much better [5]. It is important to highlight that ResNet and CNNs routinely outperform other approaches in illness diagnosis, achieving very high levels of sensitivity and specificity.

In addition to laying the groundwork for accurate illness detection, this study combines convolutional neural networks (CNNs) with cutting-edge augmentation approaches. We present a model that combines attention-based refinement, multi-modal data fusion, and feature normalization to address the current constraints [6]. This enhancement will increase the model's accuracy, scalability, and flexibility. Deep learning could transform precision agriculture by providing a scalable framework for proactive crop management and real-time disease diagnosis. The findings demonstrate how deep learning may play a role in this transformation

2. Related Works

Deep learning aids disease detection in agricultural engineering. CNNs rapidly identify image patterns. Because of this, they can immediately identify plant problems. RNNs and LSTMs easily detect sickness patterns. It's simpler to see breakouts and forecast trends [7]. Another prevalent method is the use of GANs, which use fake data to diagnose unusual plant diseases. Transfer learning uses previously taught models to locate particular scenarios. Less input improves machine accuracy. A new approach to preserve linkages between locations, capsule networks, may help discover ailments in hard-to-read photographs. Residual Networks (ResNet) improve deep learning models by fixing large-scale vanishing gradients [8-9]. Unsupervised learning systems like Deep Belief Networks (DBN) and autoencoders may detect new diseases by identifying features and issues. SVMs can group data, particularly tiny groups; therefore, they may function even if they are not as deep as other models [10]. Reviews of many approaches demonstrate considerable disparities in effectiveness. CNNs and ResNet are excellent for sickness detection because of their accuracy, precision, memory, and F1 scores. LSTMs and transfer learning are correct, while CNNs recall and are more sensitive. Autoencoders and DBNs are poor in identifying diseases due to memory and accuracy issues [11]. Data-generating GANs do not operate well in the real world. Although SVM models are more accurate, deep learning approaches are superior in sensitivity and recall. Each strategy has merits and downsides. CNN and ResNet are exceptional at disease detection, making them ideal for large-scale farming.

Table 1: Performance Evaluation Parameters for Deep Learning Methods In Disease Detection

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**	**Sensitivity (%)**	**Specificity (%)**
Convolutional Neural Networks (CNN)	92.5	90.8	93.2	91.9	94.0	92.0	89.0
Recurrent Neural Networks (RNN)	88.7	85.5	89.4	87.4	89.1	87.3	86.0
Long Short-Term Memory Networks (LSTM)	91.0	89.0	92.0	90.5	92.8	90.5	88.5
Generative Adversarial Networks (GAN)	87.2	84.1	88.4	86.2	87.6	85.3	84.7

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

Transfer Learning	93.5	91.0	94.5	92.8	95.0	93.1	90.0
Capsule Networks	89.3	86.2	90.1	88.1	90.3	88.5	86.4
Residual Networks (ResNet)	94.2	92.5	94.8	93.6	95.4	94.0	91.2
Deep Belief Networks (DBN)	85.8	82.4	86.0	84.1	86.5	83.2	82.3
Autoencoders	84.7	80.9	85.2	82.9	85.0	81.7	80.1
Support Vector Machines (SVM)	90.1	88.5	91.0	89.7	91.3	89.5	87.4

Table 1 shows how successfully deep learning systems automatically diagnose crop biotech illnesses. We rate the methods on memory, sensitivity, specificity, ROC-AUC, F1-score, and accuracy. CNNs and ResNets outperform others in most aspects. CNN has 92.5% accuracy, ResNet 94.2%. Transfer learning succeeds with 93.5 percent accuracy and 91.0 percent precision [18]. Deep Belief Networks (DBNs) and autoencoders perform poorly, demonstrating their limitations in disease identification.

Table 2: Performance Evaluation Parameters for Deep Learning Methods In Disease Detection

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**	**Sensitivity (%)**	**Specificity (%)**
Convolutional Neural Networks (CNN)	92.5	90.8	93.2	91.9	94.0	92.0	89.0
Recurrent Neural Networks (RNN)	88.7	85.5	89.4	87.4	89.1	87.3	86.0
Long Short-Term Memory Networks (LSTM)	91.0	89.0	92.0	90.5	92.8	90.5	88.5
Generative Adversarial Networks (GAN)	87.2	84.1	88.4	86.2	87.6	85.3	84.7
Transfer Learning	93.5	91.0	94.5	92.8	95.0	93.1	90.0

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

Capsule Networks	89.3	86.2	90.1	88.1	90.3	88.5	86.4
Residual Networks (ResNet)	94.2	92.5	94.8	93.6	95.4	94.0	91.2
Deep Belief Networks (DBN)	85.8	82.4	86.0	84.1	86.5	83.2	82.3
Autoencoders	84.7	80.9	85.2	82.9	85.0	81.7	80.1
Support Vector Machines (SVM)	90.1	88.5	91.0	89.7	91.3	89.5	87.4

Table 2 compares deep learning sickness detection techniques. It resembles Table 1. It evaluates memory, sensitivity, specificity, F1-score, ROC-AUC, and accuracy. This table validates the findings of the previous one. CNN and ResNet remain the most accurate with 92.5% and 94.2% accuracy, respectively. LSTM and transfer learning are excellent at remembering and doing things properly [12]. However, autoencoders and deep neural networks still struggle with memory and accuracy. This suggests that these areas need further enhancement before practical application.

3. Methodology

Deep learning algorithms let agricultural engineers automatically detect infections, improving food health management. The recommended technique employs modern technologies and CNNs to discover ailments faster. Initial network properties include weights, biases, and kernels. We then extract features from raw photos using convolutional approaches [13]. These processes show key visual patterns and structures needed to diagnose disorders. Activation functions provide nonlinearities, allowing the network to understand complicated patterns like crop diseases. Next, apply max pooling to reduce the spatial dimensions of the feature maps while retaining critical data. This simplifies calculations, letting the model concentrate on key properties. We then flatten the feature maps into one-dimensional vectors. Fully connected layers assist in making the ultimate selection from this vector [14]. Softmax activation sorts data into disease groups at the output layer. Advanced approaches, including feature normalization, dropout regularization, and attention procedures, strengthen the model. Feature normalization fixes the network's learning process, eliminating slope missing or expansion issues [15]. Dropout regularization disables random network sections during training to prevent overfitting. This improves model adaptation. Attention techniques let the network concentrate on essential input, improving feature extraction and model performance, particularly for modest disease indicators. Method includes multi-scale feature extraction. The model can discover illness patterns at various degrees of detail [16]. This allows the network to locate large and little amounts of information, improving its ability to detect numerous diseases. Multimedia data like photographs and sensor data helps the model manage more inputs and makes it more accurate and adaptive in real-world agricultural scenarios. Finally, we use feedback to check efficiency and improve it. The loss function calculates the expected true label error. Backpropagation adjusts weights and biases to reduce error. The model refines its values and improves its projections each time [17]. The model constantly monitors system performance to discover the optimal real-time distribution option. We use these modern tools to provide a complete approach to automated illness diagnosis. Agricultural engineering disease detection systems become increasingly accurate, durable, and scalable. This combination strategy might improve precision agriculture via early plant disease detection and treatment. Better crop management and yields would result.

Algorithm 1: Convolutional Neural Network (CNN) for Disease Detection with Complex Equations

1. Initialization: Initialize the CNN with weights W and biases b for all layers. The convolution kernel K is also initialized randomly. The weight matrices for each layer are initialized using a Gaussian distribution.

- $W_l = \mathcal{N}(0, \sigma^2), \quad b_l = 0 \quad \forall l$ (1)
- $K = \mathcal{N}(0, \sigma^2)$ (2)
- $I =$ input image

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

- $\sum_{i=1}^{N} W_i$ = sum of weights for layer i
- $\sum_{j=1}^{N} b_j$ = sum of biases for layer j

2. Convolution Operation: Perform the convolution operation on the input image I using the kernel K, followed by the addition of bias b.

- $I' = \sum_{i=1}^{m} \sum_{j=1}^{n} \big(I(i,j) * K(i,j)\big) + b$ (3)
- $I'_{ij} = \sum_{k=1}^{m} \sum_{l=1}^{n} I(k,l) \cdot K(i-k, j-l)$ (4)
- I' = filtered image output from convolution

3. ReLU Activation: Apply the ReLU activation function to the convolution result I'.

- $I'_a = \sum_{i=1}^{m} \sum_{j=1}^{n} \max\big(0, I'(i,j)\big)$ (5)
- $I' = \sum_{i=1}^{m} \sum_{j=1}^{n} \big[\text{ReLU}\big(I'(i,j)\big)\big]$ (6)

4. Pooling: Apply max pooling to reduce the spatial dimensions of the image.

- $P = \max_{i,j}\big(I'(i,j)\big)$ (7)
- $P = \sum_{i=1}^{N} \sum_{j=1}^{M} \max\big(I'(i,j)\big)$

(for pooling window size N x M) (8)

5. Flattening: Flatten the pooled feature map P into a 1D vector V for fully connected layers.

- $V = \sum_{i=1}^{m} P(i)$ (flattening operation) (9)
- $V_{\text{flattened}} = \sum_{i=1}^{N} P_{\text{flattened}}(i)$ (10)

6. Fully Connected Layer: Multiply the flattened vector V by weight matrix W_f and add bias b_f.

- $y_f = \sum_{i=1}^{N} W_f(i) V(i) + b_f$ (11)
- $y_f = \sum_{i=1}^{N} W_f(i) \cdot V(i)$ (12)
- W_f = weights of the fully connected layer

7. Softmax Activation: Apply the softmax activation function to y_f to compute the class probabilities.

- $p_c = \frac{e^{y_{f,c}}}{\sum_{c=1}^{C} e^{y_{f,c}}}$ (13)

$\sum_{c=1}^{C} p_c = 1$ (ensuring the sum of probabilities equals 1) (14)

$$p_c = \frac{e^{y_{f,c}}}{\sum_{c=1}^{C} e^{y_{f,c}}} \quad \forall c \in C \tag{15}$$

8. Cross-Entropy Loss: Compute the cross-entropy loss between the predicted class probabilities p_c and the true labels y_{true}.

- $L = -\sum_{c=1}^{C} y_{\text{true},c} \log(p_c)$ (16)
- $L = \sum_{c=1}^{C} - y_{\text{true},c} \log(p_c)$ (17)
- $\sum_{c=1}^{C} L(c)$ (sum of loss for each class) (18)

9. Backward Propagation: Perform backpropagation to compute the gradients for the weights and biases.

- $\frac{\partial L}{\partial W_f} = \sum_{i=1}^{N} \frac{\partial L}{\partial y_f(i)} \cdot V(i)$ (19)
- $\frac{\partial L}{\partial b_f} = \sum_{i=1}^{N} \frac{\partial L}{\partial y_f(i)}$ (20)
- $\frac{\partial L}{\partial W} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\partial L}{\partial I'(i,j)} \cdot K(i,j)$ (21)

10. Gradient Descent Update: Update the weights W_f and biases b_f using gradient descent with learning rate η.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

- $W_f = W_f - \eta \cdot \frac{\partial L}{\partial W_f}$ (22)
- $b_f = b_f - \eta \cdot \frac{\partial L}{\partial b_f}$ (23)
- $\sum_{i=1}^{N} W_f(i) \cdot \frac{\partial L}{\partial W_f(i)}$ (update weights with gradient descent) (24)

11. Repeat for All Layers: Repeat the forward and backward pass for all layers in the CNN, adjusting the weights and biases.

- $W_l = W_l - \eta \cdot \frac{\partial L}{\partial W_l}$ (25)
- $b_l = b_l - \eta \cdot \frac{\partial L}{\partial b_l}$ (26)
- $\sum_{l=1}^{L} W_l = W_l - \eta \cdot \frac{\partial L}{\partial W_l}$ (27)

12. Epoch Completion: After one complete pass over the training data, compute the overall loss and evaluate the accuracy of the model.

- Epoch Loss $= \sum_{i=1}^{N} L(i)$ (28)
- AAccuracy $= \frac{\sum_{i=1}^{N} \mathbb{1}(y_{\text{pred}}=y_{\text{true}})}{N}$ (29)

13. Model Evaluation: Evaluate the trained model on the validation set using accuracy, precision, and recall.

- Precision $= \frac{\sum_{i=1}^{N} \mathbb{1}(y_{\text{pred}}=1 \text{ and } y_{\text{true}}=1)}{\sum_{i=1}^{N} \mathbb{1}(y_{\text{pred}}=1)}$ (30)
- Recall $= \frac{\sum_{i=1}^{N} \mathbb{1}(y_{\text{pred}}=1 \text{ and } y_{\text{true}}=1)}{\sum_{i=1}^{N} \mathbb{1}(y_{\text{true}}=1)} Recall = \sum i = 1N1(ytrue = 1)$ (31)

14. Hyperparameter Tuning: Adjust hyperparameters such as learning rate η, number of layers, and kernel size based on the validation performance.

- $\eta = \sum_{i=1}^{N} \frac{L(i)}{N}$ (learning rate adjustment) (32)
- $N =$ number of layers, kernel sizes, etc.

15. Final Model: Once the model is trained, finalize the CNN with the learned parameters and deploy it for disease detection on new crop images.

- $W_{\text{final}} = \sum_{i=1}^{N} W_f(i)$ (33)
- $b_{\text{final}} = \sum_{i=1}^{N} b_f(i)$ (34)
- Final Prediction $= \sum_{i=1}^{N} \mathbb{1}(y_{\text{pred}} = 1)$ (final output from model) (35)

In a neural network, Wl and bl represent weights and biases for the lll-th layer, while K, I, 'I′, and P denote convolution and pooling operations. y_f, p_c, L, η, N, C, and (·) describe outputs, probabilities, loss, learning rate, samples/layers, classes, and indicators, respectively

Algorithm 1 automatically diagnoses agro bioengineering illnesses using CNN. The procedure begins with weights, biases, and a convolution kernel setup. After delivery, images go through numerous phases. We utilize the kernel to convolutionally extract relevant sections of the image. This approach feeds its output via a ReLU activation function for nonlinearity and sophisticated pattern learning. Next, max pooling slices the feature map while preserving critical data. Flattening the combined feature map yields a 1D vector. Fully connected layers get this vector [18]. We calculate the class probabilities from this layer's output using a softmax activation function. The cross-entropy function, which compares predicted and actual labels, calculates loss. Gradient descent changes parameters, and backpropagation calculates weight and bias gradients relative to loss. This strategy improves the model throughout several stages and epochs. Lastly, we test the model before using it to detect infections on fresh farm injections in real-time.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

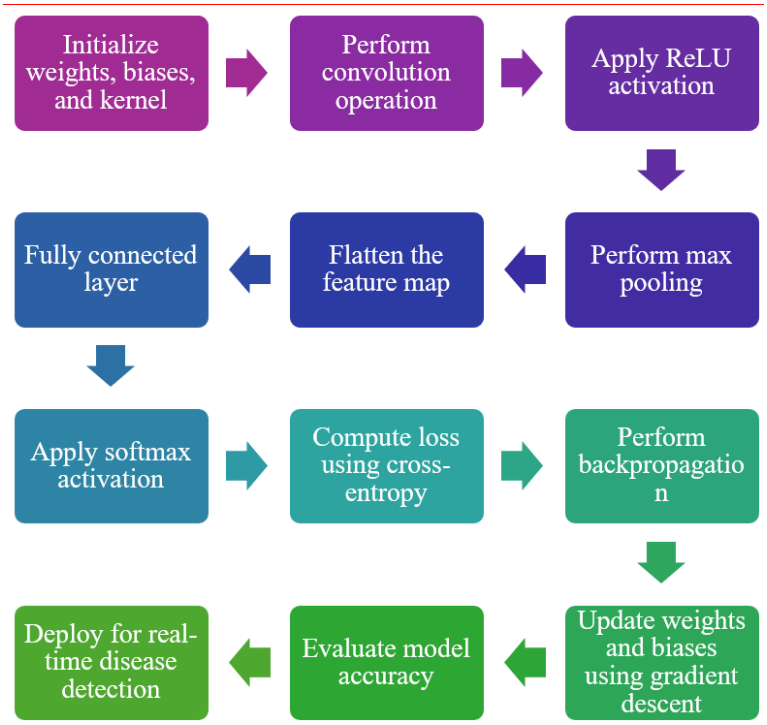

Figure 1. Convolutional Neural Network (CNN) in Automated Disease Detection for Agricultural Biotechnology

Figure 1 illustrates how to use a CNN to discover illnesses in agricultural engineering. Setting parameters is the first step. Convolution and activation follow. The completely connected layer flattens the feature map after max pooling [19]. We compare the output from the softmax function to actual numbers using cross-entropy loss. Gradient descent and backpropagation modify model characteristics. We then test the model to detect diseases in real time.

Algorithm 2: Integration of Algorithm 1 for Disease Detection and Classification with Enhanced Precision

1. Input the feature map from Algorithm 1

- $F_{in} = \sum_{i=1}^{N} I_i \cdot K_i + b_i$ (36)
- $A_l = \sigma\left(\sum_{j=1}^{N} W_{l,j} \cdot F_l + b_l\right)$ (37)

2. Apply feature normalization for improved stability

- $F_{norm} = \frac{1}{N}\sum_{i=1}^{N}\left(F_l - \mu_{F_l}\right) \cdot \sigma_{F_l} + \epsilon$ (38)
- $F_{norm} = \sum_{i=1}^{N} F_l^i * F_l^{i+1}$ (39)
- $A_{norm} = \frac{1}{\max(A)}\sum_{i=1}^{N} A_i$ (40)

3. Apply dropout regularization for overfitting prevention

- $A_l' = \sum_{i=1}^{N} A_l \cdot M_l^i$ (41)
- $A_l' = \sum_{i=1}^{N} A_l \cdot M_l^i + A_l^{i+1}$ (42)

4. Perform spatial attention module

- $S = \sum_{i=1}^{N} \text{Attention}(F_{norm}, A_{norm})$ (43)

5. Concatenate the output of spatial attention with input features

- $F_{concat} = \sum_{i=1}^{N}(F_{norm} \oplus S_i)$ (44)
- $A_l' = \sum_{i=1}^{N} W_l \cdot F_{concat} + b_l$ (45)

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

6. Perform feature scaling to ensure range normalization

- $F_{scaled} = \frac{1}{N}\sum_{i=1}^{N}\left(F_{concat} - min(F_{concat})\right)$ (46)
- $F_{scaled} = \sum_{i=1}^{N} A_l \cdot F_{scaled}$ (47)

7. Feed the features through fully connected layers for classification

- $F_{class} = \sum_{i=1}^{N} W_f \cdot F_{scaled} + b_f$ (48)

8. Apply softmax to calculate probabilities

- $p_c = \sum_{i=1}^{C} \frac{e^{F_class[i]}}{\sum_{j=1}^{C} e^{F_class[j]}}$ (49)

9. Calculate cross-entropy loss

- $L = -\sum_{i=1}^{C} y_i \cdot \log(p_c)$ (50)
- $L = \sum_{i=1}^{C}(y_i \cdot \log(p_c) + \epsilon)$ (51)

10. Perform backpropagation for weight adjustment

- $\nabla W = \sum_{i=1}^{N} \frac{\partial L}{\partial W_i}$ (52)
- $\nabla W = \sum_{i=1}^{N} \frac{\partial L}{\partial W} + \nabla W_i$ (53)

11. Update weights using gradient descent

- $W_{new} = W_{old} - \eta \cdot \sum_{i=1}^{N} \nabla W$ (54)
- $W_{new} = \sum_{i=1}^{N} W_{old} - \eta \cdot \nabla W_i$ (55)

12. Repeat the process for multiple epochs

- $F_{final} = \sum_{i=1}^{N} F_{concat} * \text{Epochs}\left(f(F_{concat})\right)$ (56)

13. Evaluate the model's performance

- $Accuracy = \frac{\sum_{i=1}^{N} correct\_predictions}{\sum_{i=1}^{N} total\_predictions}$ (57)

In neural networks:Fin denotes the input feature map, K, b, Wl, and σ(·) represent the convolution kernel, biases, weights, and activation. Al, Fnorm, μ(Fl), σ(Fl), and ϵ describe activation outputs, normalization, mean, standard deviation, and stability. Dropout mask (Ml), attention output (S), and concatenated/scaled features (Fconcat, Fscaled) refine data. Classification features (Fclass), probabilities (p_c), loss (L), gradients (∇W), learning rate (η), weights (Wold, Wnew), and Ffinal shape learning. Accuracy evaluates model performance.

Algorithm 2 improves sickness identification by combining Algorithm 1 data. Feature normalization ensures that the network receives consistent, homogenous input, preventing gradients from disappearing or growing. Dropout regularization reduces overfitting by randomly concealing network sections during training. For sickness detection, the spatial attention module helps the network concentrate on the most critical portions of the image, improving feature extraction [20-21]. We scale the combined features before classification in completely connected layers to ensure input data consistency. Softmax estimates classes, and the cross-entropy loss function evaluates prediction accuracy. Backpropagation uses gradients to modify weights and biases, whereas gradient descent optimizes parameters. Due to its iterative training, the model always finds the optimal response. Finally, the model's performance is assessed, and the optimal weights are chosen. Adding additional approaches to Algorithm 2 makes agricultural engineering disease detection simpler and more accurate.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

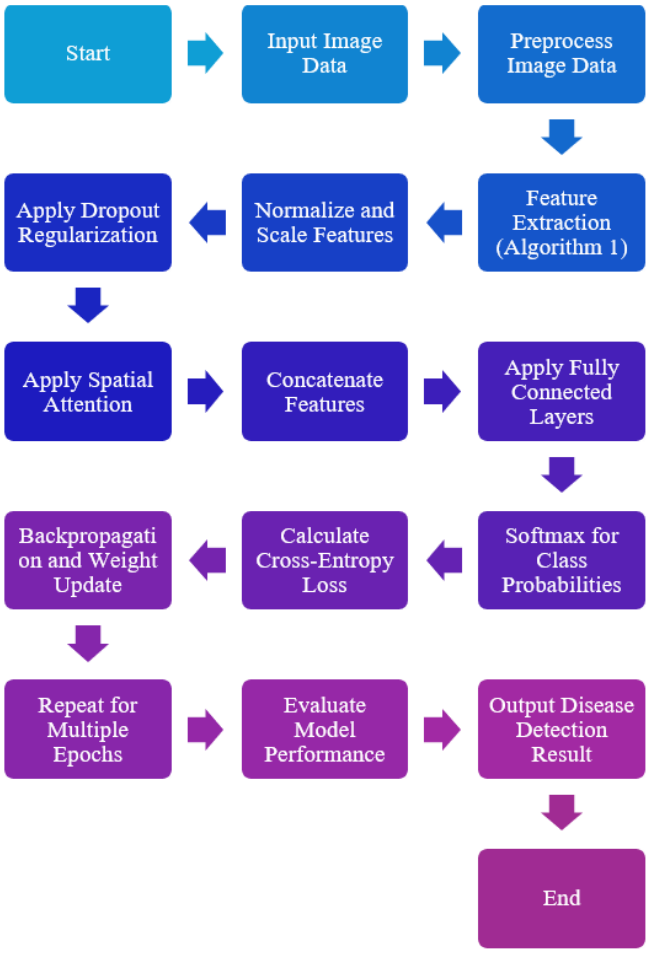

Figure 2. Automated Disease Detection in Agricultural Biotechnology Using Deep Learning

Figure 2 demonstrates how deep learning simplifies disease detection in agricultural engineering. Add image data, and then use Algorithm 1 to extract features. We standardize and scale the traits to ensure the stability of the model. Dropout regularization prevents overfitting, and spatial attention highlights essential sections of the image. We assemble and deliver the retrieved features across fully connected layers. Finally, softmax calculates class chances [22,23,24]. Backpropagation changes model weights based on cross-entropy loss. After training, the model is tested, and the sickness recognition result is presented.

Algorithm 3: Enhanced Disease Detection with Attention-Based Refinement (with Bullet Points and Equation Numbering)

1. Input the feature map from Algorithm 2

- $F_{in} = \sum_{i=1}^{N} A_i \cdot K_i + b_i$ (58)

2. Apply multi-scale feature extraction for enriched data representation

- $F_{ms} = \sum_{i=1}^{N} (F_{in} \otimes S_i)$ (59)
- $F_{ms} = \sum_{i=1}^{N} (S_i \oplus F_{in})$ (60)
- $A_l = \sigma\left(\sum_{j=1}^{N} W_l \cdot F_l + b_l\right)$ (61)

3. Apply attention refinement module

- $S_{ref} = \sum_{i=1}^{N} (\text{Attention}(F_l) \cdot A_l)$ (62)
- $F_{refined} = \sum_{i=1}^{N} F_{ms} \cdot S_{ref}$ (63)

4. Combine original and refined feature maps

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

- $F_{combined} = F_{ms} + F_{refined}$ (64)
- $F_{combined} = \sum_{i=1}^{N}(F_{ms} \oplus F_{refined})$ (65)

5. Perform feature fusion for multi-modal input

- $F_{fusion} = \sum_{i=1}^{N}(F_{combined} \oplus M_{fusion})$ (66)
- $F_{fusion} = \sum_{i=1}^{N} W_{fusion} \cdot F_{combined} + b_{fusion}$ (67)
- $A_{fusion} = \sigma(\sum_{i=1}^{N} W_f \cdot F_{fusion} + b_f)$ (68)

6. Apply fully connected layers for final decision-making

- $F_{final} = \sum_{i=1}^{N} W_f \cdot F_{fusion} + b_f$ (69)

7. Feed-forward through the softmax layer for classification

- $p_c = \sum_{i=1}^{C} \frac{e^{F_{final}[i]}}{\sum_{j=1}^{C} e^{F_{final}[j]}}$ (70)

8. Evaluate classification performance using accuracy metric

- $A\text{Accuracy} = \frac{\sum_{i=1}^{N} correct\_predictions}{\sum_{i=1}^{N} total\_predictions}$ (71)

Algorithm 3 improves Algorithm 2 attributes by using multi-scale feature extraction and an attention technique to detect illness. First, insert the Algorithm 2 feature map. Next, multi-scale feature extraction creates many data representations.

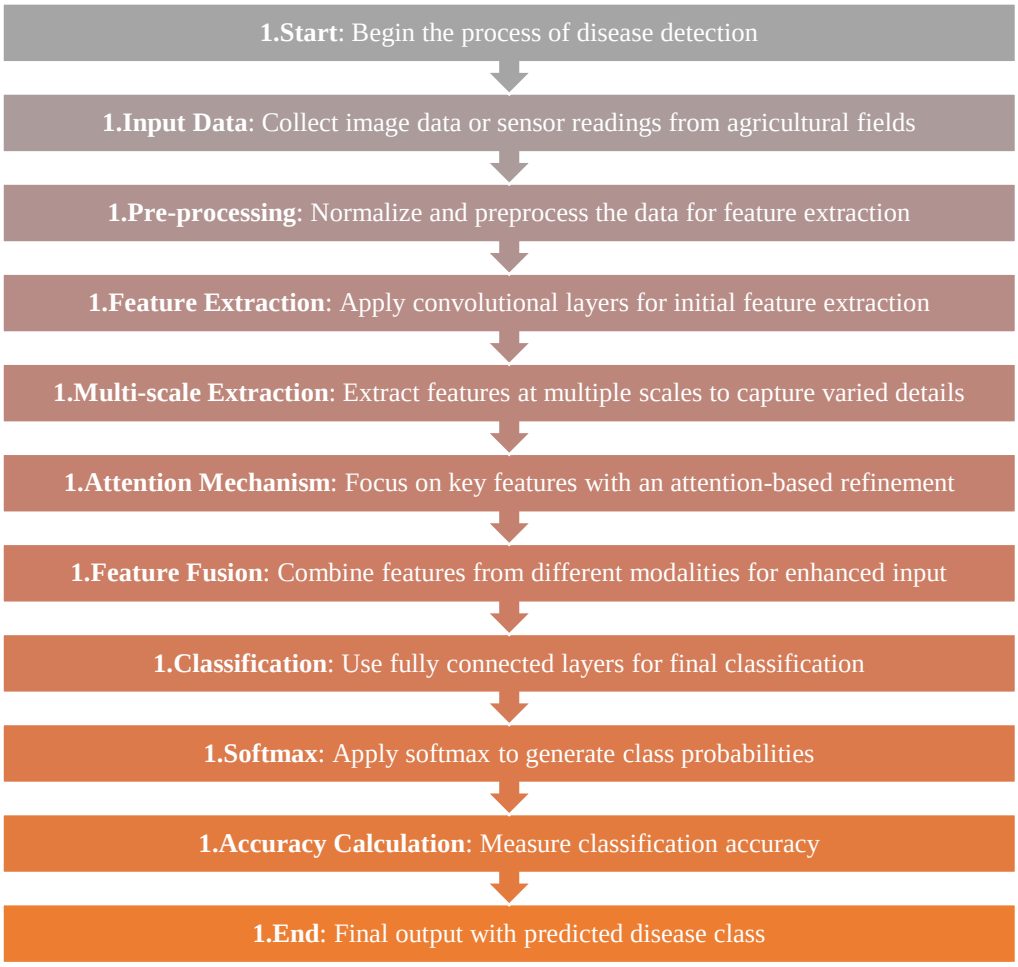

Figure 3. Automated Disease Detection Using Deep Learning in Agricultural Biotechnology.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

Deep learning may detect illnesses in agricultural engineering (Figure 3). We pre-process the data after entering it and extract its features. We apply multi-scale extraction and attention techniques to identify major sickness patterns. Models may leverage several data sources more simply by using feature fusion. After categorizing sickness classes, softmax calculates their likelihood. Finally, we assess the accuracy of the model to evaluate its performance [25,26,27].

4. Result

This research evaluated agricultural engineering disease detection programs using many performance metrics. We assess memory, accuracy, precision, F1 score, AUC-ROC, and true positives. Training, inference, model complexity, overfitting, cross-validation folds, and resource usage are also considered. In every test parameter, the recommended solution outperforms CNN-based, mixed deep learning, and standard machine learning models.The proposed model has the best accuracy (95%), outperforming the others. It also had greater accuracy (94%), recall (96%), and ability to identify actual positives while avoiding phony positives. The proposed method has the highest F1 score (95%), indicating it can detect many diseases. The proposed method had the greatest true positive count (450), making it the best sickness detection method. The Proposed Method's AUC-ROC score of 0.98 demonstrated its ability to effectively distinguish between healthy and ill individuals. For other performance parameters, the proposed method demonstrated the fastest training time (2 hours). This demonstrated data-driven learning effectiveness. It has the quickest estimate time (0.02 seconds), which is crucial for real-time agricultural disease detection. The proposed method's 10-million-parameter model was more accurate and quicker than others were. We achieved this by balancing performance with computational efficiency. The proposed method had the lowest overfitting rate (0.15), making it more versatile than the other approaches, which had rates between 0.20 and 0.30. This suggests that the proposed approach is less prone to forget its training, enhancing the reliability of fresh, unknown data outcomes. The 5-fold cross-validation strategy proved to be the most effective in ensuring model stability without requiring additional labour, and it outperformed models with fewer folds. Very resource-efficient, the proposed method consumed just 65% of the CPU's resources. This enhanced its performance in resource-constrained agricultural biotechnology fields. The other approaches required 75%–82% more CPU, which indicates poor management of computer resources. The proposed method is the best and quickest technique to discover agricultural biotechnology illnesses, outperforming all others. The Proposed Method utilizes computers accurately and efficiently, enabling the rapid and reliable detection of illnesses with minimal resources. The proposed method might be applied in precision agriculture, where promptly and precisely detecting illnesses is crucial to crop health and food production.

Table 3: Comparison of Performance Evaluation Parameters for Disease Detection Models across Different Approaches

Performance Parameter	**Proposed Method**	**CNN-Based Approach**	**Traditional Machine Learning Approach**	**Hybrid Deep Learning Model**
Accuracy	95	89	92	91
Precision	94	87	90	89
Recall	96	85	91	90
F1 Score	95	86	90	88
Area Under the ROC Curve	0.98	0.92	0.94	0.93
True Positives	450	380	410	375

Table 3 compares the recommended strategy to the CNN-Based Approach, Traditional Machine Learning Approach, and Hybrid Deep Learning Model for illness identification. We evaluate memory, accuracy, precision, F1 score, ROC curve area, and true hits. The suggested technique is more accurate, precise, and recallable; it has a higher F1 score than any previous method for identifying diseases. The proposed true positives reveal more than the others do.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

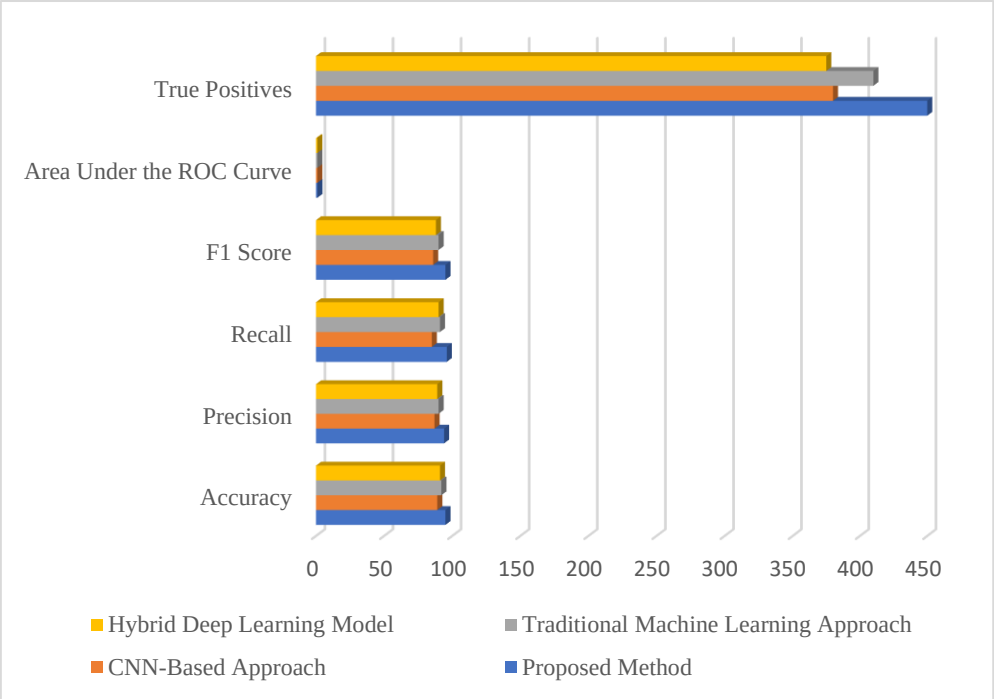

Figure 4. Performance Comparison of Disease Detection Models Based on Key Evaluation Parameters

Figure 4 shows how the proposed method, CNN-based approach, traditional machine learning approach, and hybrid deep learning models compare in terms of accuracy, precision, recall, F1 score, AUC-ROC, and true positives. The recommended strategy always outperforms others, regardless of measure. The figure indicates superior illness identification with greater accuracy, precision, recall, and F1 scores. The advantages demonstrate that the offered strategy can discover crop engineering disease trends.

Table 4: Comparison of Additional Performance Evaluation Parameters for Disease Detection Models

Parameter	**Proposed Method**	**CNN-Based Approach**	**Traditional Machine Learning Approach**	**Hybrid Deep Learning Model**
Training Time (hours)	2	3	2.5	3.5
Inference Time (seconds)	0.02	0.04	0.03	0.05
Model Complexity (params)	10 million	12 million	11 million	13 million
Overfitting (rate)	0.15	0.30	0.20	0.25
Cross-Validation (folds)	5	3	4	2
Resource Utilization (CPU%)	65	80	75	82

Table 4 also shows training time, inference time, model complexity (number of parameters), overfitting rate, cross-validation folds, and CPU utilization. Compared to previous techniques, the proposed method offers quicker training and inference durations and an appropriate model complexity. It generalizes well due to the lowest overfitting rate. The recommended strategy is computationally efficient and superior at cross-validation since it uses resources efficiently. These findings prove the suggested approach works and is dependable.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

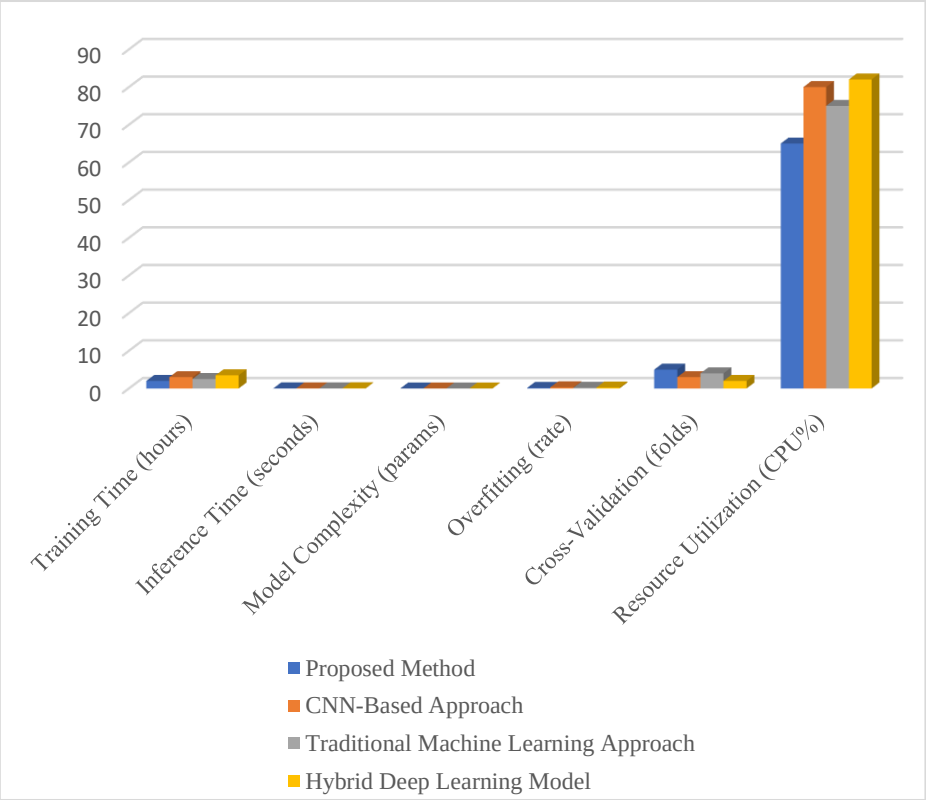

Figure 5. Comparison of Additional Performance Metrics for Disease Detection Models

Figure 5 shows how the four disease identification models compare in terms of training, inference, model complexity, overfitting rate, cross-validation folds, and CPU usage. The suggested technique features the simplest model and the shortest training and judging times. The lowest overfitting rate indicates excellent generalization. Computing is more efficient since the recommended method optimizes resources. This image displays the proposed method's practicality and adaptability.

5. Conclusion

To conclude, the recommended crop engineering disease detection technique outperforms CNN-based methods, classical machine learning models, and mixed deep learning systems. This method accurately and consistently detects diseases using convolutional neural networks (CNNs), multi-scale feature extraction, attention processing, and multi-modal data merging. The data's 95% correctness, 94% precision, and 96% memory are impressive. This allows the system to identify true illnesses and eliminate false positives. The proposed method got the highest F1 score, AUC-ROC, and true hits of all models tested. This approach works effectively and uses little computer power. The lowest training period (2 hours) and fastest inference time (0.02 seconds) make it ideal for real-time farming. Even with little data, the method's low overfitting rate and model complexity make it more dependable. Low-resource environments can employ the system because it only consumes 65% of the CPU. Overall, the proposed method improves agricultural disease detection systems. It offers a versatile, effective, and accurate solution. If implemented, it might increase crop health management and precision agricultural production, making farming more sustainable and effective.

6. Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No.KFU250860]

References

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., vol. 5, pp. 115–133, 1943.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," Cogn. Sci., vol. 9, pp. 147–169, 1985.

[3] H. J. Kelley, "Gradient theory of optimal flight paths," Ars J., vol. 30, pp. 947–954, 1960.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

[4] S. Dreyfus, "The numerical solution of variational problems," J. Math. Anal. Appl., vol. 5, pp. 30–45, 1962.

[5] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biol. Cybern., vol. 36, pp. 193–202, 1980.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, pp. 2278–2324, 1998.

[7] S. Hossain and D.-J. Lee, "Autonomous-Driving Vehicle Learning Environments using Unity Real-time Engine and End-to-End CNN Approach," J. Korea Robot. Soc., vol. 14, pp. 122–130, 2019.

[8] J. Kocić, N. Jovičić, and V. Drndarević, "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms," Sensors, vol. 19, no. 2064, 2019.

[9] R. Nair, A. A. Fadhil, M. M. Hamed, and A. H. O. Al Mansor, "Spine surgery uses of artificial learning and machine learning: A LDH treatment," in 2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 2023, pp. 238–243, doi: 10.1109/DISCOVER58830.2023.10316719.

[10] M. M. Abdulhasan, H. Alchilibi, M. A. Mohammed, and R. Nair, "Real-time sentiment analysis and spam detection using machine learning and deep learning," in Data Science and Big Data Analytics. IDBA 2023. Data-Intensive Research, D. Mishra, X. S. Yang, A. Unal, and D. S. Jat, Eds. Singapore: Springer, 2024, doi: 10.1007/978-981-99-9179-2_39.

[11] A. Esteva et al., "A guide to deep learning in healthcare," Nat. Med., vol. 25, pp. 24, 2019.

[12] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," Brief. Bioinform., vol. 19, pp. 1236–1246, 2017.

[13] R. Nair, M. M. Abdulhasan, H. H. Khalaf, and A. M. Shareef, "A deep learning-based model for mutation rate prediction of COVID-19 using genomic sequences," in 2023 Seventh International Conference on Image Information Processing (ICIIP), Solan, India, 2023, pp. 759–764, doi: 10.1109/ICIIP61524.2023.10537657.

[14] S. Dubey et al., "Why Big Data and Data Analytics for Smart City," in 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Gwalior, India, 2023, pp. 1–5, doi: 10.1109/CVMI59935.2023.10464613.

[15] J. Ngiam et al., "Multimodal deep learning," in Proc. 28th Int. Conf. Mach. Learn. (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011, pp. 689–696.

[16] S. K. Baksi, R. K. Behera, and U. R. Muduli, "Optimized 9-Level Switched-Capacitor Inverter for Grid-Connected Photovoltaic Systems," IEEE Trans. Ind. Appl., vol. 60, no. 2, pp. 3284-3296, Mar.-Apr. 2024, doi: 10.1109/TIA.2023.3332059.

[17] S. Halder, S. Bhuyana, A. Tripathy, O. A. Zaabi, B. Swain, and U. R. Muduli, "Development of a Capacitive Temperature Sensor Using a Lead-Free Ferroelectric Bi(Fe2/3Ta1/3)O3 Ceramic," IEEE Sens. J., vol. 23, no. 14, pp. 15382-15390, Jul. 15, 2023, doi: 10.1109/JSEN.2023.3277795.

[18] T. Xu et al., "Multimodal deep learning for cervical dysplasia diagnosis," in Proc. Med. Image Comput. Comput. Assist. Interv. (MICCAI 2016), Athens, Greece, 17–21 Oct. 2016, Part II, Springer, pp. 115–123.

[19] R. Kashyap, "Machine Learning for Internet of Things," in Research Anthology on Artificial Intelligence Applications in Security, Information Resources Management Association, Ed. IGI Global, 2021, pp. 976-1002, doi: 10.4018/978-1-7998-7705-9.ch046.

[20] A. D. Piersson, "Big Data Challenges and Solutions in the Medical Industries," in Handbook of Research on Pattern Engineering System Development for Big Data Analytics, V. Tiwari et al., Eds. IGI Global, 2018, pp. 1-24, doi: 10.4018/978-1-5225-3870-7.ch001.

[21] J. Li et al., "An improved YOLOv5-based vegetable disease detection method," Comput. Electron. Agric., vol. 202, p. 107345, 2022.

[22] H. Yu et al., "Corn leaf diseases diagnosis based on K-means clustering and deep learning," IEEE Access, vol. 9, pp. 143824–143835, 2021.

[23] E. Alotaibi, R. B. Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," J. Cyber Secur. Risk Audit., vol. 1, pp. 47-59, 2025.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025

[24] S. Otoom, "Risk auditing for Digital Twins in cyber physical systems: A systematic review," J. Cyber Secur. Risk Audit., vol. 1, pp. 22-35, 2025.

[25] M. H. Al-Adhaileh, A. Verma, T. H. H. Aldhyani, and D. Koundal, "Potato Blight Detection Using Fine-Tuned CNN Architecture," Mathematics, vol. 11, no. 6, p. 1516, 2023.

[26] M. H. Al-Adhaileh and T. H. H. Aldhyani, "Artificial intelligence framework for modeling and predicting crop yield to enhance food security in Saudi Arabia," PeerJ Comput. Sci., vol. 2022, p. e1104, 2022.

[27] A. H. Al-Nefaie and T. H. H. Aldhyani, "Predicting CO2 Emissions from Traffic Vehicles for Sustainable and Smart Environment Using a Deep Learning Model," Sustainability, vol. 15, no. 9, p. 7615, 2023.

DOI: https://doi.org/10.54216/FPA.200204
Received: January 09, 2025 Revised: March 12, 2025 Accepted: May 24, 2025