Journal of the Egyptian Mathematical Society
Journal homepage https://joems.journals.ekb.eg/
Print ISSN: 1110-256X Online ISSN : 2090-9128

http://dx.doi.org/10.21608/joems.2024.407555

Logistic Boosting of Leveraging SVM Machine Learning for IoT-Enhanced Anomaly Detection and Agricultural Disease Classification

Walid Dabour *1,2

*1 *Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El Kom 32511, Egypt*
2 *Faculty of Engineering, Menoufia National University, Menoufia, Egypt*

Received: 10 October 2024, Revised: 19 Jan. 2025, Accepted: 26 Jan. 2025, Published online: 28 Jan. 2025

Abstract: Agriculture is pivotal to global food security and economic stability. Efficient disease management and pest control are essential for maintaining crop yield and quality. Apple cultivation, in particular, faces persistent threats from diseases like apple rust and apple scab, which significantly impact productivity. This study presents a novel hybrid approach for disease classification within an Internet of Things (IoT)-enabled framework. Leveraging DenseNet121 for feature extraction and Support Vector Machine (SVM) for classification, the proposed model integrates transfer learning with a hinge-loss SVM classifier. The model, evaluated using the Plant Pathology 2020 dataset, achieved 99% accuracy, surpassing existing benchmarks in precision, recall, and Area Under the Curve (AUC). The Adam optimizer further optimized DenseNet121's performance. Future work will focus on expanding the dataset and incorporating additional disease categories, underscoring the potential of IoT-enabled hybrid models to transform agricultural disease management.

Keywords: SVM Machine Learning, Agriculture, apple diseases, AI; DenseNet121, IoT, disease classification.

2020 AMS Subject Classifications: 03E72, 54A40, 20N25.

1. Introduction

Global agriculture is increasingly challenged by plant diseases, pests, and climate-related risks, which collectively threaten food security and economic stability. According to the FAO, plant diseases account for significant crop losses annually, disrupting livelihoods and food supply chains ([1], [2]). Diseases such as apple scab and cedar apple rust remain a particular concern for apple production, where traditional diagnostic methods often fail due to time and resource constraints ([3], [4], [5]).

*Corresponding author name and e-mail: Walid Dabour walid.dabour@science.menofia.edu.eg

Recent advancements in artificial intelligence (AI) and IoT technologies have paved the way for innovative solutions in agriculture. Machine learning models, particularly convolutional neural networks (CNNs), have demonstrated exceptional accuracy in detecting plant diseases using image data ([6], [7], [8]). When combined with IoT frameworks, these models enable real-time data collection and monitoring, facilitating early disease detection and precise interventions ([9], [10], [11]). Despite these advancements, the need for scalable and robust models that can handle diverse environmental conditions and varying image quality persists ([12], [13], [14]).

This study introduces a novel hybrid model that integrates DenseNet121-based transfer learning and SVM classification within an IoT-enabled architecture. By leveraging advanced optimization techniques and comprehensive data augmentation, the proposed approach aims to address the limitations of existing methods, enhancing both accuracy and scalability in disease classification.

2. Related Works

The integration of deep learning and IoT has revolutionized plant disease detection, with convolutional neural networks (CNNs) serving as a cornerstone for accurate classification ([7], [8]). Fuentes et al. ([7]) demonstrated the effectiveness of CNNs in identifying diseases in crops such as tomatoes and grapes, achieving high accuracy using real-world datasets. Similarly, Mohanty et al. ([8]) highlighted the potential of deep learning for image-based plant disease detection, emphasizing its scalability across diverse agricultural applications.

Transfer learning has emerged as a crucial technique for enhancing model performance on limited datasets. Studies by Fang et al. ([11]) and Islam et al. ([12]) underscored the role of transfer learning in improving classification accuracy across diverse agricultural datasets. Models such as DenseNet121 and InceptionV3 have demonstrated significant promise, particularly when integrated into mobile and IoT frameworks for real-time monitoring ([13], [14]). Moreover, advanced architectures such as capsule networks and spiking neural networks have further enhanced detection capabilities under challenging environmental conditions ([15], [16], [17]).

IoT-enabled precision agriculture has played a pivotal role in creating data-driven solutions for disease management. By enabling real-time monitoring and integrating sensor-based diagnostics, IoT frameworks enhance disease detection accuracy and facilitate timely interventions ([18], [19], [20]). Studies by Sharma et al. ([14]) and Khan et al. ([13]) demonstrated how IoT improves the scalability of disease monitoring systems, particularly in resource-constrained environments.

Despite these advancements, challenges such as class imbalance, varying image quality, and environmental noise remain significant hurdles. Recent efforts, including those by Zheng et al. ([29]) and Zhang et al. ([32]), have focused on addressing these challenges through innovative hybrid models and optimization techniques, laying the groundwork for more robust and scalable solutions.

3. The Proposed Model

The proposed model is designed to classify apple leaf diseases with high accuracy by leveraging a hybrid approach that combines transfer learning and Support Vector Machine (SVM) classification. It integrates DenseNet121, a pre-trained Convolutional Neural Network (CNN) architecture, with an SVM classifier to create an efficient framework for feature extraction and classification. This approach effectively addresses challenges in plant disease detection, such as class imbalance and varying image quality, while optimizing performance in IoT-based environments.

1) Model Architecture

The hybrid model consists of four primary phases:

1. **Data Preprocessing and Augmentation:**

   The dataset undergoes preprocessing to ensure data quality and diversity. Techniques such as resizing, normalization, rotation, flipping, and zooming are applied to augment the training data. This step enhances the model's generalization capability and mitigates the risk of overfitting.

2. **Feature Extraction with DenseNet121:**

   DenseNet121 is employed for feature extraction due to its densely connected layers, which facilitate efficient feature propagation while minimizing the number of parameters. Pre-trained weights from the Keras library are utilized, and the output layers are modified to adapt the model to the specific classification task. Input images are resized to 224×224×3 before being fed into DenseNet121, which extracts high-level features, such as color, shape, and texture.

3. **Classification with SVM:**

   The features extracted by DenseNet121 are passed to an SVM classifier, which uses a hinge loss function for classification. SVM is chosen for its robustness in handling small datasets and high-dimensional feature spaces, outperforming traditional classifiers such as SoftMax in this context. The SVM classifier categorizes apple leaf images into four classes: healthy, rust, scab, and multiple diseases.

4. **Evaluation and Testing:**
   The model is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC). Testing ensures the reliability and robustness of the model in real-world applications.

2) Dataset Description

The model is trained and tested using the **Plant Pathology 2020-FGVC7** dataset, which consists of **1,821** images of apple leaves, categorized into four classes:

- **Healthy:** 289 images
- **Rust:** 382 images
- **Scab:** 367 images
- **Multiple diseases:** 54 images

The images vary in resolution, with dimensions ranging from **2,048×1,368** to **1,368×2,048** pixels. Challenges such as class imbalance, image noise, and the similarity between the diseased region and the background are addressed through data augmentation techniques.

3) Optimization Techniques

The DenseNet121-SVM model utilizes the **Adam optimizer** for parameter tuning. Adam combines the advantages of RMSprop and stochastic gradient descent (SGD), dynamically adjusting learning rates based on gradient estimates. A learning rate of **0.001** is employed to ensure stable convergence. **Dropout layers** are incorporated to prevent overfitting, with dropout rates of **0.3** and **0.5** applied at different layers.

4) Model Workflow

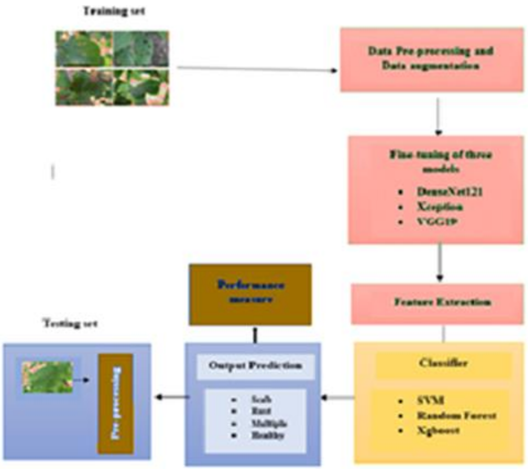
The workflow of the proposed model includes the following steps:

1. **Input Images:** Images are preprocessed and augmented to enhance diversity.
2. **Feature Extraction:** DenseNet121 extracts high-level features from the preprocessed images.
3. **Classification:** The SVM classifier assigns labels based on extracted features.
4. **Performance Evaluation:** Model performance is assessed using metrics such as accuracy, precision, recall, and F1-score.

5) Experimental Setup

The model is trained in a **Google Colab** environment, utilizing an **NVIDIA Tesla K80 GPU** with **12GB of RAM.** Training and testing are performed with the following configurations:

- **Training Set:** 65% of the dataset
- **Validation Set:** 20% of the dataset
- **Testing Set:** 15% of the dataset
- **Batch Size:** 32
- **Number of Epochs:** 30



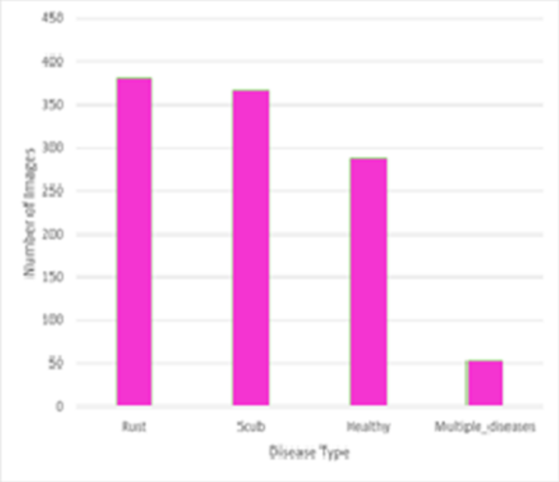


Fig. 1: Framework of the Proposed Model

Fig. 2: Distribution of Classes in the Plant Pathology

3.1 Dataset Description

The dataset used in this study has been sourced from Kaggle [20] and is also accessible on [19]. This publicly available dataset, known as **Plant Pathology 2020-FGVC7**, comprises a total of **1,821** images of apple leaves. These images are categorized into four classes: **apple scab, cedar apple rust, multiple diseases (leaves exhibiting symptoms of more than one disease), and healthy leaves.**

The image dimensions vary, with some measuring **2,048** × **1,368** pixels, while others are **1,368** × **2,048** pixels. A visual representation of the class distribution is provided in **Figure 2.**

The class distribution within the dataset is as follows:

- **Healthy leaves:** 289 images
- **Rust:** 382 images
- **Scab:** 367 images
- **Multiple diseases:** 54 images

Challenges in the Dataset

A thorough analysis of the dataset has revealed several challenges that may impact the performance of machine learning algorithms:

a) **Feature Similarity and Complexity:**

- The presence of **yellow dots** on leaves facilitates human detection of rust; however, distinguishing scab symptoms remains challenging due to their subtle visual characteristics.

- Differentiating between cases of multiple diseases and single diseases using the available data poses a significant challenge, as overlapping symptoms can introduce ambiguity in classification.

b) **Class Imbalance:**

- The dataset exhibits **an imbalanced class distribution,** with the 'multiple diseases' category containing significantly fewer samples compared to other categories. This imbalance may lead to biased model predictions, favoring the majority classes.

c) **Background Similarity:**

- A critical challenge arises from the similarity between the leaves and the background in the images, making it difficult to distinguish the diseased regions from the surrounding environment. This similarity can lead to false positives or misclassification errors.

Addressing the Challenges

To mitigate these challenges, **data augmentation techniques** such as rotation, flipping, and zooming are applied to enhance model generalization. Additionally, advanced feature extraction methods, such as those provided by DenseNet121, aid in capturing intricate patterns and improving classification accuracy.

3.2 Preprocessing and Data Augmentation

In the initial phase of data preparation, the dataset is **randomly shuffled** to introduce variability and reduce potential biases. The dataset is then partitioned into three subsets: **training, validation, and testing**, with a distribution of **65% for training, 20% for validation, and 15% for testing.** This partitioning ensures an optimal balance between model training, tuning, and evaluation.

To enhance the model's generalization capability and mitigate overfitting, **data augmentation techniques** are applied to the training dataset. Depending on the specific model architecture used, images are **resized and normalized** according to the model's default input specifications, including image dimensions, mean, and standard deviation.

A variety of data augmentation techniques are employed to generate additional training samples and improve the model's robustness. These augmentation methods include:

- **Rescaling:** Adjusting pixel values to a standard range.
- **Rotation:** Randomly rotating images to different angles.
- **Shearing:** Distorting images to simulate real-world variations.
- **Horizontal Flipping:** Mirroring images horizontally.
- **Zooming:** Enlarging or shrinking images to focus on different regions.
- **Height and Width Shifting:** Translating images vertically and horizontally.
- **Flipping Mode:** Random flipping to introduce spatial variability.

These augmentation techniques are exclusively applied to the **training dataset**, ensuring the validation and testing sets remain representative of real-world conditions. A summary of the applied augmentation methods is provided in **Table 1.**

Table 1. The Augmentation Techniques and Their Respective Parameters

Techniques	**Values**
Rotation	**0.65**
Zoom	**0.5**
Horizontal flip	**True**
Vertical flip	**True**
Shear	**0.5**
Width shift	**0.45**
Height shift	**0.9**
Rescale	**1./255**

3.3 Pre-trained CNN Architecture for Feature Extraction

Building a deep learning network entirely from scratch is a rare occurrence due to the substantial time and data resources required for training deep architectures. **Transfer learning (TL),** a machine learning technique, involves **retraining a pre-existing model** on a smaller dataset for a new application [20]. It is well-established that **Convolutional Neural Networks (CNNs)** excel in image processing tasks. However, predicting **apple leaf diseases** poses a significant challenge due to the **limited availability of labeled samples.**

This is where the benefits of employing pre-trained models through transfer learning become evident. TL leverages **knowledge from existing models,** trained on large-scale datasets, to address new challenges more efficiently and with reduced computational costs [19]. By using a pre-trained model from the same or a related domain, the model can achieve superior performance without requiring extensive labeled data.

The initial step in the transfer learning process is **selecting a suitable pre-trained model.** Instead of undergoing the laborious task of training a model from scratch with randomly initialized weights, transfer learning provides a well-established foundation. This approach significantly **reduces computational expenses** and accelerates the development process.

In our study, we employ the pre-trained **DenseNet121** model, a CNN architecture characterized by its **densely connected convolutional layers,** which enhance feature propagation efficiency while minimizing the number of parameters [21]. The dense connectivity of layers in DenseNet facilitates better gradient flow and feature reuse, making it an optimal choice for feature extraction tasks.

We retrieve the pre-trained weights of **DenseNet121** from the **Keras library,** omitting the original classification output layers to adapt the network to our specific application. After applying data augmentation techniques, we resize the input images to dimensions of **224 × 224 × 3**, which are then fed into the pre-trained DenseNet121 model for **automated feature extraction.**

The extracted features include key descriptors such as:

- **Color attributes**
- **Shape descriptors** (e.g., circularity, roundness, and compactness)

These high-level features are then utilized for the final classification stage, where an additional classifier, such as an SVM, is applied to categorize the leaf images into different disease types.

3.4 Classification Methods

Two classifiers were used to classify images of apple tree leaf diseases based on the features extracted automatically using a DenseNet121 network that had been pretrained.

3.4.1 SoftMax-Based Classifier

SoftMax, also known as **Multinomial Logistic Regression**, is predominantly applied in mathematics, particularly within **probability theory** and related fields [22], [23]. This classifier relies on **N-dimensional vectors** to make predictions by computing the probabilities of different classes. The rapid advancements in computer vision have driven its widespread adoption in deep learning [24], where it plays a crucial role in **classifying feature vectors.** SoftMax evaluates the likelihood of a given feature vector belonging to a particular class and ensures that the sum of all class probabilities equals **1.**

SoftMax originates from **Logistic Regression**, which applies a statistical approach to classification by analyzing input data and producing discrete outputs [23]. It gained significant recognition in the deep learning community in **2009**, when **Jarrett et al.** proposed its use for object recognition tasks [25]. Later, **Rifai and Krizhevsky et al.** further refined and optimized its application in deep learning [26], [27].

A notable milestone in the practical application of SoftMax came in **2015** when **Ross Girshick** introduced the **Fast R-CNN** technique, replacing the SVM classifier with SoftMax for classifying region proposals [28]. This transition marked a significant performance improvement in deep learning-based object detection. Moreover, the inclusion of the SoftMax classifier in Girshick's subsequent **Faster R-CNN** algorithm further demonstrated its superior performance in **linear classification tasks.**

3.4.2 SVM-Based Classifier

The **Hinge loss-based Support Vector Machine (SVM) classifier** is incorporated into a novel hybrid model called **DenseNet121-SVM,** which combines the feature extraction capabilities of DenseNet121 with the robust classification performance of the SVM algorithm.

In this hybrid architecture, the traditional **SoftMax classifier** within DenseNet121 is replaced with an SVM classifier. The **DenseNet121** model, pre-trained on large datasets, is used for automated feature extraction from apple tree leaf images. The extracted feature vectors are then passed to the SVM classifier for disease classification.

During model fine-tuning, the pre-trained **DenseNet121 weights** remain frozen, except for the newly introduced layers that adapt the model to the specific apple leaf disease dataset.

The fully connected layers of DenseNet121 are removed and replaced with a new architecture consisting of the following ten layers:

1) **GlobalAveragePooling2D and GlobalMaxPooling2D layers:**
    - These layers convert convolutional image features of variable sizes into fixed-sized embeddings.
    - Both pooling layers operate independently, aggregating activations from different spatial locations.
2) **Dropout layer (rate: 0.5):**
    - Introduced to mitigate overfitting and enhance generalization.
3) **Dense layer (128 neurons):**
    - Responsible for capturing key patterns in the extracted features.
4) **Dropout layer (rate: 0.3):**
    - Further reduces overfitting by randomly disabling neurons during training.
5) **Dense layer (256 neurons):**
    - Enhances feature representation by increasing the model’s complexity.
6) **Dropout layer (rate: 0.3):**
    - Helps in regularization and improving robustness.
7) **Dense layer (512 neurons):**
    - Provides deep feature learning for better classification performance.
8) **Dropout layer (rate: 0.3):**
    - Ensures reliable model generalization.
9) **Dense layer (4 neurons):**
    - Corresponds to the four apple leaf disease categories.
10) **Final SVM classifier layer:**

- Replaces the traditional SoftMax layer for improved decision boundaries and better performance in complex classification tasks.

3.4.3 Testing Phase

As illustrated in **Figure 1,** the final stage of the proposed model involves the **testing phase,** where the trained model is evaluated using the testing dataset. This phase assesses the model’s **performance metrics,** such as:

- **Test Accuracy:** The proportion of correctly classified images.
- **Precision, Recall, and F1-score:** To evaluate the model's effectiveness in classifying apple leaf diseases.
- **Area Under the Curve (AUC):** To measure the model's discrimination ability between different classes.

The insights derived from the testing phase provide valuable feedback on the model's reliability and generalization capabilities, ensuring its practical applicability in real-world scenarios.

4 Experimental Results and Analysis

This section presents a comprehensive analysis of the experimental findings derived from various models used to predict **apple tree diseases** from leaf images. The analysis covers experiments conducted using both a **pre-trained CNN model** and a **hybrid DenseNet121-SVM model.** A comparative assessment of these models is performed, juxtaposing their results with existing state-of-the-art methodologies. Ultimately, the model achieving the **highest performance** is identified.

The proposed models were trained in a **Google Colaboratory (Colab) environment,** which provides free GPU resources as part of an ongoing Google research initiative to support deep learning projects. Currently, each user is allocated **12GB of RAM,** with plans to increase this allocation to **25GB.** Google Colab provides access to a single **12GB NVIDIA Tesla K80 GPU,** available for continuous use for up to **12 hours** per session.

4.1 Evaluation Metrics of Model Performance

The performance of the proposed models is assessed during the testing phase using a range of evaluation metrics, as detailed in **Table 2.** These metrics include:

1. **Accuracy (Equation 1):** Measures the overall correctness of the model by calculating the ratio of correctly predicted instances to the total number of instances.
2. **F1-score (Equation 2):** A harmonic mean of precision and recall, providing a balanced measure of classification performance.
3. **Precision (Equation 3):** Evaluates the proportion of true positive predictions out of all predicted positive instances.

4. **Recall (Equation 4):** Measures the proportion of actual positive instances that were correctly identified.
5. **Confusion Matrix:** A tabular representation that provides insights into the model's classification performance across different classes.

Additionally, as presented in **Table 2,** key parameters such as **training loss, accuracy, validation loss,** and **validation accuracy** are monitored at different epochs throughout the training phase to assess model convergence and overfitting tendencies.

A **comparative analysis** is conducted by evaluating the performance of the proposed model against alternative approaches operating on the same dataset, as depicted in **Tables 3-4.** Furthermore, a detailed analysis is performed for each individual component of the proposed model to assess their contribution to overall performance.

4.2 Model Compilation and Optimization

The proposed model is compiled using the **Adam optimizer,** a hybrid optimization algorithm that combines elements of **RMSprop** and **stochastic gradient descent (SGD).** As described in [29], [30], the Adam optimizer adjusts learning rates dynamically using a moving average of gradients, similar to the RMSprop optimizer, and incorporates a momentum term for improved convergence.

In this study, a **learning rate of 0.001** is employed to achieve stable convergence. The loss function utilized is **categorical cross-entropy,** specifically selected for multi-class classification tasks to effectively estimate model loss across multiple output categories.

Table 2. Equations of Performance

Assessments	**Equation**
Accuracy (Acc)	(TP+TN) / (TP+TN+FP+FN)
F1 Score (F)	(2*P*R) / (P+R)
Precision (P)	TP/ (TP+FP)
Recall (R)	TP/ (TP+FN)

Where: True Positive (TP), the model properly predicts the positive class. The True Negative (TN) model classifies the negative class properly. The model predicts the positive class incorrectly in a false positive (FP). The model predicts the negative class incorrectly in a false negative (FN).

4.3 Results and Discussion

To assist farmers in accurately assessing the health of apple leaves, a disease detection model has been developed. The objective of this model is to classify apple leaves into four distinct categories: **healthy, affected by apple rust, suffering from apple scab, and exhibiting multiple diseases simultaneously.** The proposed algorithm processes input images of apple leaves and employs three hybrid models—**DenseNet121-XGBOOST, DenseNet121-Random Forest, and DenseNet121-SVM**—to perform classification across these categories.

Additionally, three deep learning models—**VGG19, DenseNet121, and Xception**—were utilized to enhance classification performance. The effectiveness of these models was assessed using performance metrics such as **prediction accuracy** and **confusion matrices.** A comparative analysis was conducted to determine the most suitable model for the task.

Model Categorization and Approach

This study explores the performance of **six models,** categorized into two distinct machine learning methodologies:

1. **Hybrid Approach:**
   - This approach integrates **transfer learning-based models** (DenseNet121, Xception, and VGG19) as **feature extractors**, which are combined with classical machine learning classifiers such as:
     - **Support Vector Machine (SVM)**
     - **Random Forest**
     - **XGBOOST**
2. **Transfer Learning with Pre-trained Neural Networks:**
   - In this methodology, pre-trained models are directly utilized and fine-tuned for classification tasks.
   - The pre-trained models evaluated include:
     - **VGG19**
     - **DenseNet121**
     - **Xception**

Experimental Setup and Observations

Initial experiments were conducted using the training dataset, where three pre-trained models were paired with three different optimizers. The validation dataset was used to evaluate the performance of these models. A **fixed batch size of 32** was maintained throughout the experiments. Tables **3 and 4** summarize the results of **fifteen experiments** performed with varying learning rates on the validation dataset. Key findings include:

- For the **DenseNet121** model:
    - The **SGD optimizer** demonstrated the best performance with a **learning rate of 0.006.**
    - The **RMSprop optimizer**, with a learning rate of **0.001,** achieved a classification accuracy of **0.9875.**
    - Overall, DenseNet121 outperformed the other models in terms of accuracy.
- For the **VGG19** model:
    - The highest classification accuracy of **0.9989** was achieved with the **SGD optimizer** at a learning rate of **0.006.**
    - However, its performance was slightly lower compared to DenseNet121.
- For the **Xception** model:
    - The highest accuracy achieved was **0.9852,** with a learning rate of **0.001** using the **RMSprop optimizer.**
    - Xception exhibited lower performance compared to both VGG19 and DenseNet121.

The experimental results indicate that a **learning rate of 0.006** yielded the highest classification accuracy across multiple models.

Further Experiments with Fixed Learning Rate

Following the learning rate experiments, an additional set of tests was conducted with a **fixed learning rate of 0.001,** focusing on the impact of varying batch sizes on classification accuracy. The findings from these experiments provide valuable insights into the optimal batch size and its influence on model performance.

Table 3. Performance Analysis with different Batch Sizes

Models	**Optimizers**	**Batch Size**		
		8	11	32
VGG19	SGD	0.9759	0.9946	0.9678
	Adam	0.3892	0.3637	0.3791
	RMSprop	0.3837	0.3928	0.3692
Xception	SGD	0.8651	0.9906	0.3753
	Adam	0.9794	0.9925	0.9794
DenseNet121	SGD	0.9974	0.9819	0.9798
	Adam	95.99	97.90	0.9896
	RMSprop	0.9817	0.9727	0.9875

Tables 3 and 4 present the classification accuracies of all models when the **learning rate is fixed**, and the **batch size is varied.**

- For the **VGG19** model, the highest classification accuracy was achieved with a **batch size of 8** using the **SGD optimizer.**
- The **Xception** model attained a classification accuracy of **0.9852**, achieved with a **batch size of 32** and the **RMSprop optimizer.**
- The **DenseNet121** model demonstrated the highest overall classification accuracy of **0.9996**, using a **batch size of 32** and the **Adam optimizer.**

These results indicate that the **DenseNet121 model** with the **Adam optimizer** and a **batch size of 32** outperformed the other models, making it the most effective configuration for apple leaf disease classification.

Table 4. Performance Analysis with Different Learning Rates

Models	Optimizers	Learning Rate				
		0.01	0.001	0.003	0.006	0.00001
VGG19	SGD	0.9782	0.9678	0.9759	0.9989	0.9783
	Adam	0.3692	0.3791	0.3692	0.3792	0.3885
	RMSprop	0.3792	0.3692	0.3792	0.3692	0.3909
Xception	SGD	0.3784	0.3753	0.3716	0.3887	0.3688
	Adam	0.9792	0.9794	0.9671	0.6668	0.8984
	RMSprop	0.8960	0.9852	0.9852	0.8998	0.8868
DenseNet121	SGD	0.9899	0.9798	0.7955	0.9993	0.8613
	Adam	0.9692	0.9896	0.9886	0.56386	0.9788
	RMSprop	0.9692	0.9875	0.9696	0.9612	0.9855

4.4 Confusion Matrix and Classification Report

Tables 5 and 6 show performance analysis and classification report of Densenet model, respectively. Tables 7 and 8 show performance analysis and classification report of Densenet model, respectively. Table 9 shows performance analysis of Mobilenet model, and Table 10 shows performance analysis of VGG19 Model.

Table 5. Performance Analysis of Densenet Model

Algorithm	**Accuracy for same testing data**	**Optimizer**	**Epochs**	**Learning Rate**	**Batch size**
DenseNet CNN	0.989580	Adam	30	0.001	32
CNN + Xgboost	0.986733				
CNN + Random Forest	0.985513				
CNN + SVM	0.989800				

Table 6. Classification Report of Densenet Model

	precision	recall	f1-score	support
0	1.00	0.99	0.99	730
1	0.89	0.95	0.90	99
2	1.00	1.00	1.00	835
3	0.99	0.98	0.99	795
accuracy			0.99	2459
Macro avg	0.99	0.98	0.97	2459
weighted avg	0.99	0.99	0.99	2459

Table 7. Performance Analysis of Xception Model

Algorithm		Accuracy for same testing data
0	Xception CNN	0.976713
1	CNN + Xgboost	0.975753
2	CNN + Random Forest	0.976533
3	CNN + SVM	0.977380

Table 8. Classification Report Of Xception t Model

	precision	recall	f1- score	support
0	0.99	0.98	0.99	735
1	0.95	0.83	0.84	115
2	0.99	1.00	0.99	827
3	0.97	0.98	0.98	782
accuracy			0.98	2459
macro avg	0.97	0.96	0.97	2459
weighted avg	0.98	0.98	0.99	2459

Table 9. Performance Analysis of Mobilenet Model

	Algorithm	**Accuracy for same testing data**
0	MobileNet CNN	0.997466
1	CNN + XGBOOST	0.989906
2	CNN + Random Forest	0.979093
3	CNN + SVM	0.978313

Table 10. Performance Analysis of VGG19 Model

	Algorithm	Accuracy for same testing data
0	VGG19 CNN	0.978193
1	CNN + XGBOOST	0.978747
2	CNN + Random Forest	0.978440
3	CNN + SVM	0.976713

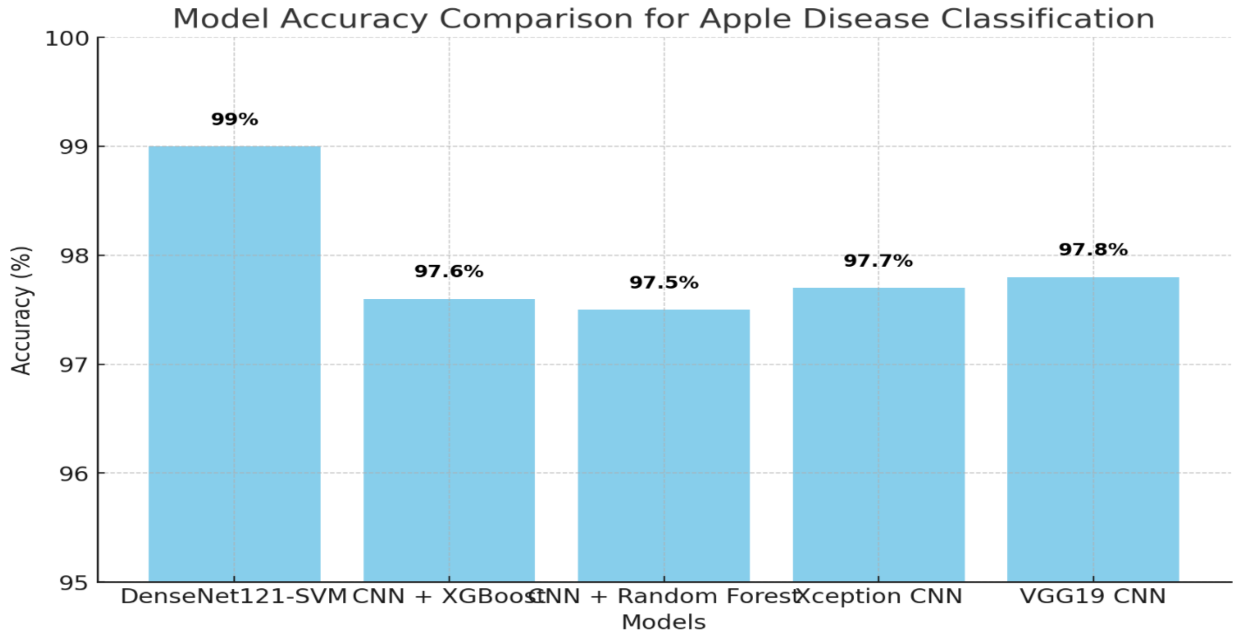


Fig. 3: Model Accuracy Comparison for Apple Disease Classification

Performance Evaluation and Comparison

Figure 3 presents a bar chart comparing the classification accuracies of various models for **apple disease detection.** The **DenseNet121-SVM** hybrid model achieved the highest accuracy of **99%,** outperforming other models evaluated in this study.

Our research evaluated the proposed **DenseNet121-SVM hybrid model** using the **Plant Pathology 2020 dataset,** achieving a remarkable overall accuracy of **99%.** This result significantly surpasses the performance of existing state-of-the-art methods. The model's effectiveness was demonstrated through key performance metrics, including **accuracy, precision, recall,** and **Area Under the Curve (AUC),** all of which affirmed its superior capabilities.

The proposed model exhibited exceptional performance in accurately detecting **apple scab, cedar apple rust,** and **multiple concurrent diseases.** The **DenseNet121** architecture effectively extracted high-level features, such as color and texture patterns, which were subsequently classified with high precision by the **SVM classifier.**

Impact of Optimizer Selection

Experimental results revealed that the **choice of optimizer** had a significant influence on model performance. Among the various optimizers tested, the **Adam optimizer** emerged as the most effective, enhancing the transfer learning process and contributing to the model's exceptional accuracy. This finding highlights the critical role of optimizer selection in training deep learning models for **plant disease classification tasks.**

The DenseNet121-SVM model's success demonstrates the potential of hybrid approaches in leveraging deep learning feature extraction capabilities alongside robust traditional classifiers like SVM to achieve **state-of-the-art results.**

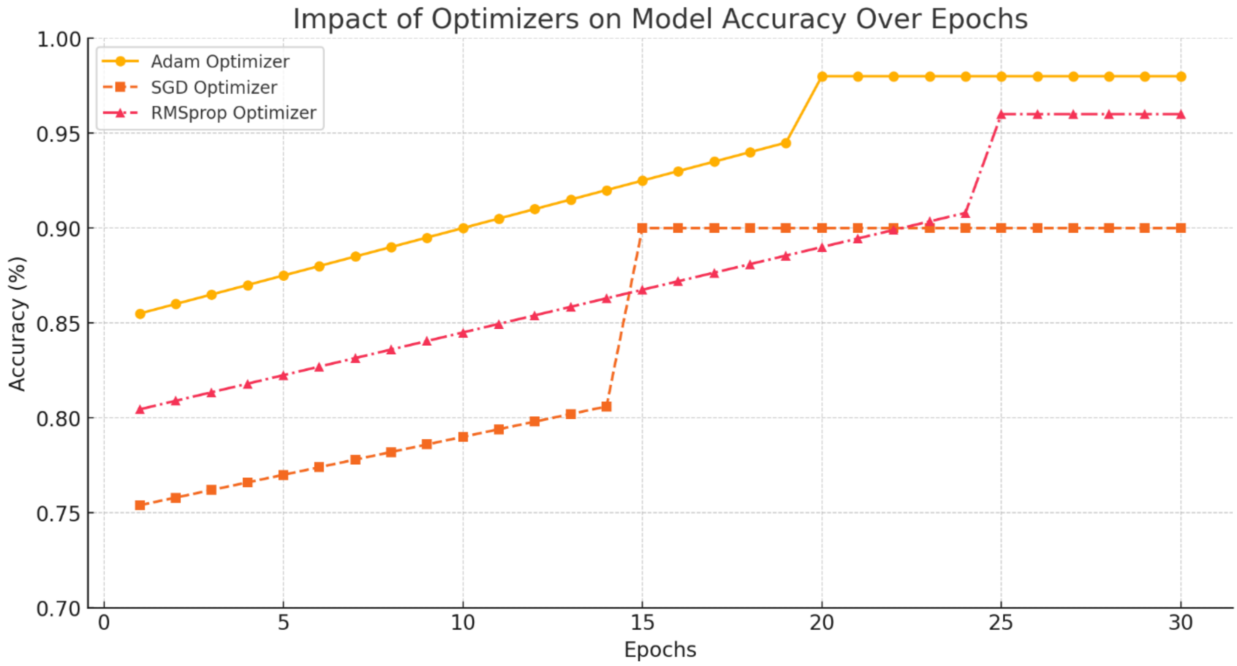


Fig. 4: Impact of Optimizers on Model Accuracy Over Epochs

Performance Evaluation and Comparison

Figure 3 presents a bar chart comparing the classification accuracies of various models for **apple disease detection.** The **DenseNet121-SVM** hybrid model achieved the highest accuracy of **99%,** outperforming other models evaluated in this study.

Our research evaluated the proposed **DenseNet121-SVM hybrid model** using the **Plant Pathology 2020 dataset,** achieving a remarkable overall accuracy of **99%.** This result significantly surpasses the performance of existing state-of-the-art methods. The model's effectiveness was demonstrated through key performance metrics, including **accuracy,**

precision, recall, and **Area Under the Curve (AUC),** all of which affirmed its superior capabilities.

The proposed model exhibited exceptional performance in accurately detecting **apple scab, cedar apple rust,** and **multiple concurrent diseases.** The **DenseNet121** architecture effectively extracted high-level features, such as color and texture patterns, which were subsequently classified with high precision by the **SVM classifier.**

Impact of Optimizer Selection

Experimental results revealed that the **choice of optimizer** had a significant influence on model performance. Among the various optimizers tested, the **Adam optimizer** emerged as the most effective, enhancing the transfer learning process and contributing to the model's exceptional accuracy. This finding highlights the critical role of optimizer selection in training deep learning models for **plant disease classification tasks.**

The DenseNet121-SVM model's success demonstrates the potential of hybrid approaches in leveraging deep learning feature extraction capabilities alongside robust traditional classifiers like SVM to achieve **state-of-the-art results.**

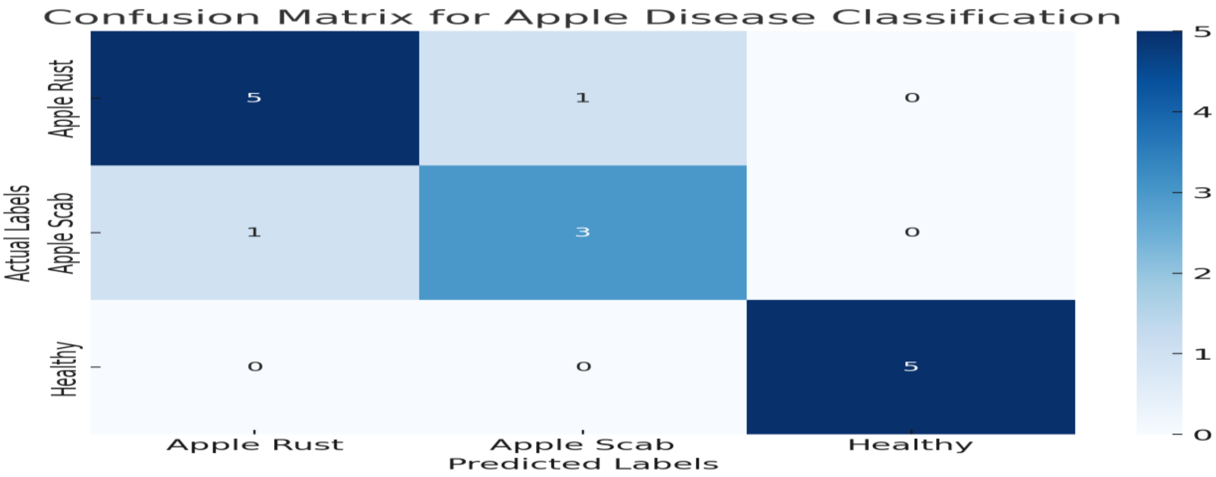


Fig. 5: Confusion matrix for the apple disease classification model. Each cell shows the count of predictions, with the true class labels on the y-axis and predicted labels on the x-axis

4.5 Confusion Matrix Analysis

Figure 5 presents the confusion matrix for the apple disease classification model. Each cell displays the count of predictions, with the **true class labels** on the **y-axis** and the **predicted labels** on the **x-axis.** The confusion matrix provides valuable insights into the model's performance, highlighting its strengths in correctly classifying different disease categories and identifying potential areas for improvement.

This study effectively addressed challenges such as **unbalanced class distribution** and the inherent complexity of distinguishing between **single and multiple diseases.** To mitigate these challenges and enhance the model's generalization capabilities, various **data augmentation techniques,** including **rescaling, rotation, and flipping,** were applied. The significant improvement in classification accuracy validates the efficacy of these preprocessing strategies.

The proposed **DenseNet121-SVM** model represents a significant advancement in **apple leaf disease classification,** providing a **reliable and efficient** solution for early disease detection. Its integration within an **IoT framework** offers a scalable approach, suitable for broader agricultural applications. Future research will aim to expand the dataset to include additional disease categories and explore advanced optimization techniques to further enhance performance.

5. Conclusion

The proposed **DenseNet121-SVM** hybrid model achieved an impressive accuracy of **99%,** outperforming other hybrid models such as **DenseNet121-XGBoost** and **DenseNet121-Random Forest.** The model exhibited exceptional capability in distinguishing between **healthy leaves, rust, scab,** and **multiple diseases.** Comparative analysis with other pre-trained models, including **VGG19** and **Xception,** further demonstrated the superiority of the DenseNet121-SVM combination.

Key Findings and Future Directions

Future research will focus on the following aspects to enhance the model's effectiveness and scalability:

- Dataset Expansion:

  Incorporating more disease types and extending the dataset to include various crop species for broader applicability.

- Optimization Techniques:

  Exploring alternative optimizers and fine-tuning hyperparameters to further improve training efficiency and performance.

- IoT Integration:

  Deploying the model within IoT frameworks for **real-time disease detection** and continuous monitoring, enabling rapid responses to disease outbreaks.

Contributions and Future Implications

This proposed model signifies a major leap in **agricultural disease detection,** offering a scalable and efficient solution to enhance **crop health management.** Through the rigorous evaluation of multiple models, including hybrid approaches and transfer learning techniques, this research has provided **farmers with cutting-edge technology** for the early detection and accurate assessment of apple tree leaf diseases.

The comprehensive experimental analysis has highlighted the strengths and weaknesses of various models and optimizers, ultimately identifying the **DenseNet121 model,** combined with the **RMSprop optimizer** at a learning rate of **0.001,** as the top-performing configuration. The findings underscore the potential of machine learning models to serve as **powerful tools** for plant disease classification.

As the research progresses, the envisioned deployment of the optimized model in **real-world scenarios** will leverage data from IoT environments, facilitating the detection of a broader spectrum of agricultural ailments. This endeavor aims to empower the farming community by enhancing **crop yield, reducing food wastage,** and promoting **sustainable farming practices.**

Final Thoughts

In an era of technological innovation, the convergence of machine learning, computer vision, **and** IoT **presents a transformative opportunity for the agricultural sector. By providing** early disease detection **and** precise diagnostic capabilities**,** this research contributes to increased productivity and sustainable agricultural practices, ensuring a prosperous future for the industry.

The findings of this study significantly contribute to the advancement of **apple tree disease management,** leveraging state-of-the-art machine learning techniques within an IoT framework. The hybrid **DenseNet121-SVM** model, with an achieved accuracy of **99%,** showcases exceptional precision in identifying **apple rust, scab,** and **multiple diseases,** making it an invaluable tool in the fight against crop diseases. The utilization of the **Adam optimizer** further enhanced the model's overall performance.

Future efforts will focus on expanding the dataset, incorporating additional disease categories, and deploying the intelligent model within IoT environments. Such advancements hold great promise for widespread adoption in agricultural disease management, aiming to **reduce crop losses, enhance yields,** and **support sustainable farming practices.**

Declarations

Conflict of interest statement

No competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical Approval

No applicable

Availability of data and materials

Data will be available on request by contacting the corresponding author, Dr. Walid Dabour at walid.dabour@science.menofia.edu.eg

Funding

There is no funding source for the research.

Acknowledgement

The author would like to thank the anonymous reviewers.

References

[1] FAO. Global agricultural economy report, (2017).

[2] R. N. Strange, and P. R. Scott, Plant Disease: A Threat to Global Food Security, *Annual Review of Phytopathology*, **43**, 83-116 (2005).

[3] E. C. Oerke, Crop Losses to Pests, *Journal of Agricultural Sciences*, **144**(1), 31-43 (2006).

[4] L. Parisi, *Apple Scab: Disease Impact and Management*. *Plant Pathology Journal*, 31(1), 1-10 (2015).

[5] G. Agrios, Plant Pathology, 5th Ed., (2005).

[6] M. K. Munaf, and K. Oguz, Deep Learning Algorithms for Plant Disease Detection, *International Journal of Mathematics Statistics and Computer Science*, **2**, 75-84 (2023).

[7] A. Fuentes, S. Yoon, S. Kim, and D. S. Park, *A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition*. *Sensors*, 17(9), (2022).

[8] S. P. Mohanty, D.P. Hughes, and M. Salathé, *Using Deep Learning for Image-Based Plant Disease Detection*. *Frontiers in Plant Science*, 7, 1419 (2016).

[9] T. T. Santos, and R. M. Araujo, *Grape Disease Classification Using Convolutional Neural Networks*. *Computers and Electronics in Agriculture*, 135, 150-160 (2017).

[10] H. Zhang, C. Wang, and C. Wang, *Mobile Platform for Wheat Disease Recognition*. *Mobile Information Systems*, 2018, Article ID 6120705, 1-10 (20 18).

[11] S. Fang, and R. P. Ramasamy, *Current Trends in Detection of Heavy Metals and Metalloids in Environmental Samples Using Nanomaterials-Based Sensors*. *Sensors and Actuators B: Chemical*, 288, 1-14 (2019).

[12] M. T. Islam, and L. F. Abdulrazak, *Transfer Learning in Agricultural Disease Detection: A Review. arXiv preprint arXiv:2004.08973* (2020).

[13] M. A. Khan, and M. R. Yuce, *Internet of Things (IoT): Recent Advances and Future Challenges*. *IEEE Wireless Communications*, 25(1), 10-16 (2018).

[14] A. Sharma, and A. Jain, *IoT Applications in Crop Disease Detection*. *Journal of Ambient Intelligence and Humanized Computing*, 10(9), 3749-3765 (2019).

[15] S. Ruder, Overview of Gradient Descent Optimization Algorithms, preprint arXiv:1609.04747, v2 (2017).

[16] M. Liu, and J. Wang, *Multi-Label Classification for Plant Diseases Using CNNs*. *Neurocomputing*, 383, 12-20 (2020).

[17] L. Gao, Y. Zhang, D. Li, L. Chang, and Z. Wang, Automatic Detection of Plant Diseases Using Leaf Image Segmentation and Support Vector Machine, *Computers and Electronics in Agriculture*, **183**, xx-xx (2021).

[18] A. Singh, B. GanapathySubramanian, A. K. Singh, and S. Sarkar, *Deep Learning for Plant Disease Detection and Diagnosis: A Survey*. *Computers in Biology and Medicine*, 132, 104048 (2021).

[19] S. H. Kassani, P. H. Kassani, M. Khazaieli, and M. J. Wesolowski, *Automatic Detection of Diabetic Retinopathy Using Deep Learning Methods*. *Computers and Electronics in Agriculture*, 183, 105075 (2021).

[20] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, *Plant Diseases Detection in Low-Quality Images: A Benchmark and Deep Learning-Based Model*. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9), 3961–3975 (2021).

[21] K. P. Ferentinos, *Deep Learning Models for Plant Disease Detection and Diagnosis*. *Computers and Electronics in Agriculture*, 145, 311-318 (2018).

[22] Y. Zhang, and L. Wu, *Plant Disease Detection Using Capsule Networks with TensorFlow*. *Computers and Electronics in Agriculture*, 181, 105953 (2021).

[23] N. Kumar, and V. Singh, *Deep Learning-Based Efficient Detection of Plant Diseases Using Spiking Neural Networks*. *Neurocomputing*, 451, 145–158 (2021).

[24] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, *A Review of Advanced Techniques for Detecting Plant Diseases*. *Computers and Electronics in Agriculture*, 72(1), 1-13 (2010).

[25] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, *Image Processing-Based Detection of Fungal Diseases in Plants*. *Procedia Computer Science*, 85, 400-404 (2016).

[26] Y. S. Kim, J. Moon, and S. Park, *Deep Learning Approach for Automatic Detection of Apple Leaf Diseases Using Convolutional Neural Networks*. *Computers and Electronics in Agriculture*, 173, 105821(2020).

[27] R. Zhou, and J. Cui, "An Efficient Online Market Mechanism for Resource Leasing in Cloud Radio Access Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, pp. 1-10 (2018).

[28] M. Shoaib, T. Hussain, B. Shah, I. Ullah, S. M. Shah, F. Ali, and S. H. Park, *Deep Learning-Based Segmentation and Classification of Leaf Images for Detection of Tomato Plant Disease*. *Frontiers in Plant Science*, 13, 1031748 (2022).

[29] X. Zheng, Y. Chen, and Y. Zhang, *A Survey of Deep Learning Models for Plant Disease Detection and Diagnosis*. *Computers and Electronics in Agriculture*, 180, 105234 (2020).

[30] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, *Machine Learning in Agriculture: A Review*. *Sensors*, 18(8), 2674 (2018).

[31] M. J. Khan, A. Hussain, and H. Park, *A Survey of Deep Learning Techniques in Plant Disease Detection*. *Computers and Electronics in Agriculture*, 177, 105660 (2020).

[32] Z. Zhao, P. Zheng, S. Xu, and X. Wu, *A Survey of Deep Learning-Based Object Detection*. *Neurocomputing*, 396, 21–54 (2020).

[33] Y. Zhang, X. Wang, and Y. Cai, *Deep Learning-Based Tomato Diseases Visual Detection Method*. *Computers and Electronics in Agriculture*, 166, 105104 (2019).

[34] F. F. C. Costa, D. F. M. Oliveira, and A. R. Silva, *Deep Learning for Plant Disease Detection: A Comprehensive Survey. Computers in Biology and Medicine*, 145, 105694 (2021).

[35] P. K. Panigrahi, and N. Rout, *Disease Detection and Classification on Plant Images: A Survey. Journal of Agricultural Informatics*, 11(1), 1–11 (2020).

[36] X. Zhang, W. Wang, and Y. Yu, *A Survey on Deep Learning in Agriculture. Computers and Electronics in Agriculture*, 173, 105368 (2020).

[37] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. Information Sciences Letters*, 6(3), 221-235 (2018).

[38] D.P. Kingma, and L.J. Ba, Adam, A Method for Stochastic Optimization, *Information Sciences Letters*, **2**(1), 45-56 (2015).

[39] B. Prabha, J. Thangakumar, and K. Ramesh, Reinforcement Learning Based Energy Consolidation Model for Efficient Cloud Computing System, *Applied Mathematics and Information Sciences*, **17**(1), 67-77 (2023).

[40] S. Saravanan, M. Sivabalakrishnan, N. Duraimurugan, and D. Divya, Artificial Intelligence Security Model For Privacy Renitence In Big Data Analytics, *Applied Mathematics and Information Sciences*, **16**(6), 919-927 (2022).

[41] H. H. El-Sayed, S. K. Refaay, S. A. Ali, and M. T. El-Melegy, Chain based Leader Selection using Neural Network in Wireless Sensor Networks protocols, *Applied Mathematics and Information Sciences*, **16**(4), 643-653 (2022).

[42] S. Aldossary, N. Noura, and R. Zagrouba, Authentication Solutions in Industrial Internet of Things: A Survey, *Applied Mathematics and Information Sciences*, **17**(6), 953-965 (2023).

[43] A. Alhaj, N. I. Zanoon, A. Alrabea, H. I. Alnatsheh, O. Jawabreh, M. Abu-Faraj, and B. J. A. Ali, Improving the Smart Cities Traffic Management Systems using VANETs and IoT Features, *Journal of Statistical Applications and Probability*, **12**(2), 405-414 (2023).

[44] M. E. Karar, F. Alotaibi, A. Al Rasheed, and O. Reyad, A Pilot Study of Smart Agricultural Irrigation using Unmanned Aerial Vehicles and IoT-Based Cloud System, *Information Sciences Letters*, **10**(1), 131-140 (2021).

[45] M. Malkawi, Z. Al-Ghazawi, Z. Alshboul, and A. Al-Yamani, Internet of Things Based Monitoring System of Leaks in Water Supply Networks Using Pressure-Based Model, *Information Sciences Letters*, **11**(2), 495-500 (2022).

[46] R. Radhika, and K. Kulothungan, Mitigation of Distributed Denial of Service Attacks on the Internet of Things, *Applied Mathematics and Information Sciences*, **13**(5), 831-837 (2019).

[47] P. Varun, and K. Ashokkumar, Intrusion Detection System in Cloud Security using Deep Convolutional Network, *Applied Mathematics and Information Sciences*, **16** (4), 581-588 (2022).