International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I9P111

Volume 72 Issue 9, 127-138, September 2024

Original Article

Enhancing Tomato Fruit Disease Detection using Dung Beetle Optimization with Deep Transfer Learning based Feature Fusion Model

K. Sundaramoorthi[1], Mari Kamarasan[2]

[1,2]*Department of Computer and Information Science, Annamalai University, Annamalainagar, Tamil Nadu, India.*

[1]*Corresponding Author : csksundar7@gmail.com*

Received: 01 April 2024 Revised: 02 August 2024 Accepted: 03 September 2024 Published: 28 September 2024

Abstract - *Tomato fruit, scientifically known as Solanum lycopersicum, is economically important and the most widely consumed fruit worldwide. Tomato fruit disease encompasses a range of ailments that may damage the yield, marketability, and quality of tomatoes. Several pathogens, such as viruses, fungi, bacteria, and environmental conditions cause this. Plant disease is most major aspect restricting the sustainable process of agriculture and has often been a challenging conundrum in agricultural production. Current developments in technology, including the usage of Deep Learning (DL) methods for disease detection, provide promising avenues for timely intervention and target management of Tomato Fruit Disease. These systems can precisely diagnose diseases through analyzing images of diseased plants, enabling growers to perform early detection and diminish crop loss. In this study, a new Tomato Fruit Disease Detection using Dung Beetle Optimization with Deep Feature Fusion (TFDD-DBODFF) model is introduced. The TFDD-DBODFF model aims to improve tomato fruit disease detection. Primarily, the CLAHE-based preprocessing is performed. Besides, the TFDD-DBODFF technique follows a deep feature fusion process containing 3 DL approaches such as Residual Network (ResNet), Capsule Network (CapsNet), and SqueezeNet. In addition, the hyperparameter tuning of these DL models can be performed using the DBO technique. Finally, the classification and detection of tomato fruit diseases take place using a Stacked Autoencoder (SAE). The investigational outcome of the TFDD-DBODFF approach can be examined using a benchmark dataset. The experimental validation indicates the optimum performance of the TFDD-DBODFF approach with existing methods under diverse measures.*

Keywords - *Tomato Fruit Disease Detection, Feature fusion, Dung Beetle Optimization, CLAHE, Deep learning.*

1. Introduction

Plant diseases provide a great domain for analysis in the scientific field and are considered biological features of diseases. Nowadays, recognition of plant diseases is demonstrated to be challenging and needs special attention [1]. Earlier identification of these diseases supports sustaining the better quality of tomatoes, as estimated by consumers. A conventional technique for disease analysis, disease recognition, and controlling diseases in tomatoes encompasses manually performing these processes [2]. The expansion of an automatic system to perform all these tasks for the user can eliminate difficulties and minimize the time taken by executing them manually [3]. In the agricultural industry, plant diseases are the primary cause of this. Cultivators or farmers could face numerous tomato diseases. It could be identified or positioned on fruits, roots, stems, and leaves of the plant [4]. General indications of plant damage are nematodes, fungal, bacterial, and virus diseases, which can be the source of spots in the yellowing of minor leaves, black or brown lesions, stem or leaves, final death of low leaves, and black spots [5]. The humidity and alternative ecological modifications in the existing day generate these. Each disease has various elected controls or avoided to these diseases. Extensive methods are employed in chemical usages, cultural practice, and disease-resilient variabilities [6]. The conventional machine vision techniques for the identification of crop leaf ailments have three stages, namely (1) image pre-processing, (2) Research workers methodically developing intricate disease features for feature extraction, and (3) Machine Learning (ML) methods for categorization [7]. The authors presented an automated detection methodology for cucumber disease. Depending on the various spectral reflection features and the effect of optical filtering under disease identification [8]. It can be utilized as a Genetic Algorithm (GA) to develop identification parameters from two viewpoints of spectral reflection and shape features to recognize the diseases. Conventional machine vision techniques need multi-faceted pre-processing and development of image features that are laborious and time-consuming. The efficiency of this technique is especially

reliant on the precision of the artificial form features and learning method [9]. With the fast expansion of DL, the precision of object detection and image classification becomes significantly enhanced, and it will correctly classify massive datasets, even higher than humans in several features [10]. The DL method has the benefit of directly extracting classification features. This study introduces a new Tomato Fruit Disease Detection using Dung Beetle Optimization with Deep Feature Fusion (TFDD-DBODFF) methodology. The purpose of the TFDD-DBODFF model is to improve tomato fruit disease detection. At the initial stage, the CLAHE-based preprocessing is performed. Besides, the TFDD-DBODFF technique follows a deep feature fusion model containing 3 DL approaches: Residual Network (ResNet), Capsule Network (CapsNet), and SqueezeNet. In addition, the DBO technique can perform the hyperparameter tuning of these DL approaches. Finally, the classification and detection of tomato fruit diseases take place using a Stacked Autoencoder (SAE). The investigational outcome of the TFDD-DBODFF technique can be examined using a benchmark dataset.

2. Literature Works

Anu et al. [11] developed DL method. This method utilized image processing techniques for the aim of segmentation and pre-processing in conjunction with a multiple-class CNN for classification. The method of separating the unclean region at the modest image regions is achieved using Gaussian blur, thresholding, and canny edge recognition methods. Afterwards, the features could be removed by utilizing a CNN. In [12], architecture was developed by implementing the DL multivariate normal DL-NN (MNDLNN) method. Primarily, the input image colors were transferred into HSI formats. Next, by employing a Random Motion Squirrel Search Optimizer (RMSSO), the vital features have been removed. In conclusion, MNDLNN successfully identifies and categorizes the disease categories. Shoaib et al. [13] introduced a solution employing a DL-based method.

This model employed a model for DL dependent upon a newly designed CNN employing a supervised learning method and Inception Net architecture. Besides, 2 recent semantic segmentation methods, such as modified UNet and UNet, were exploited. Abouelmagd et al. [14] presented a CV technique that employs an improved CapsNet for identifying and categorizing 10 tomato leaf diseases employing conventional database images. Preprocessing and data augmentation models have been implemented at the training stage to mitigate overfitting. CapsNet has been elected through CNNs because of its greater capacity for capturing spatial positioning along with the image. In [15], a custom CNN method (CCNN) was developed. Alternatively, in recent models like AlexNet and VGG16, the developed CCNN model has three convolution and Fully Connected (FC) layers that decrease the executing time and commutating ability although accomplishing a superior accuracy in the categorization of diverse diseases. Besides, the amount of parameters of the developed method was considerably diminished when compared to the present methods. Arafat et al. [16] introduced a robotized model. The examination zeroed in on automating the earlier position of tomato leaf diseases by employing an IoT platform and a modified ResNet-50 DL model. Initially, IoT devices, comprising sensors and cameras, were transmitted in tomato domains to collect plant-relevant data and images. The system also considered evaluating the hyper limitations of preprepared methods, comprising ResNet-50, GoogLeNet, and SquezeNet. Paul et al. [17] developed a lightweight modified CNN method and employed Transfer Learning (TL)-based approaches VGG16 and VGG19 for classification. Furthermore, an ablation analysis could be executed to determine the optimum parameters for the developed system. Thangaraj et al. [18] propose the Modified-Xception-assisted Multiple Level Feature Fusion (MX-MLF2) technique. The methodology employs multi-level extraction, integrated with TL and fine-tuning for prediction. Ye et al. [19] introduce the Tswin-F network, a Transformer-based approach. It incorporates modules, namely bilateral attention and self-supervised learning, to capture and improve positional data. The incorporation of these modules enhances spatial connections and model accuracy. Furthermore, the Feature Fuse Local Attention (FFLCA) method addresses the challenge of increasing attention distances in deeper transformer layers. Patel and Patil [20] propose an advanced CNN with Spatial Pyramid Pooling (SPP) and adaptive momentum backpropagation, incorporated with an FIR filter for preprocessing. Additionally, a stacked sparse denoising AE-SVM (SSDAE-SVM) model, with dropout mechanisms, enhances fruit grading effectiveness. Sun, Nicholaus, Fu, and Kang [21] present a lightweight YOLOv8n-based object-detection approach that maintains high accuracy. The method features the FMDI neck for multi-scale fusion, the MFN for effective feature extraction, and the UIB block for a simplified model structure. Indumathi and Kumuthaveni [22] introduce an Extended Stochastic Coati Optimized Transfer Learning with Vision Transformer (ESCOTLViT) model that uses the ESCO to fine-tune pre-trained CNN models for effectual plant disease detection. ESCO improves model performance by optimizing parameters such as neurons and learning rates through a two-group division and three-phase process, enhancing the Coati Optimization Algorithm (COA) model. Pandurangan et al., [23] present GD-Attention, a self-attention mechanism that utilizes global pixel value distribution. By integrating both image and pixel distribution information, GD attention assists the model in better prioritising and extracting features related to disease detection. Chilakalapudi and Jayachandran [24] propose a Chronological Flamingo Search Algorithm (CFSA) with TL technique. It comprises noise reduction with Adaptive Anisotropic Diffusion, segmentation with U-Net++ and MGRA, image augmentation for dimensionality reduction, feature extraction, and two-level classification—plant type

and disease—using CNN-based TL with LeNet, trained with CFSA. The existing techniques utilizing Gaussian blur and edge detection for segmentation may face difficulty with noisy or complex images, potentially limiting CNN performance. Models comprising RMSSO for feature extraction can be computationally intensive, and transitioning to HSI color formats adds complexity. Semantic segmentation and supervised learning methods may encounter threats with diverse leaf images and new diseases. Enhanced CapsNet techniques could have scalability issues with conventional images, while custom CNNs optimized for speed might sacrifice accuracy. Lightweight CNN approaches utilizing TL may not handle all diseases efficiently without fine-tuning. Multi-level feature extraction methodologies can enhance computational complexity, and transformer-based models may find difficulty due to high costs and long training times. Techniques depending on global pixel distribution may face difficulty if the data is not representative, and models utilizing chronological Flamingo search may face efficiency issues with massive datasets. Present techniques for plant disease detection face threats, namely handling noisy or complex images, high computational costs, and limited adaptability to various or new disease types. There is a requirement for more effective, scalable methods that enhance accuracy while managing resource constraints and can generalize better to diverse plant conditions and diseases.

3. The Proposed Model

This study introduces a new TFDD-DBODFF methodology. The TFDD-DBODFF model aims to enhance the recognition of the results of tomato fruit diseases. It contains different kinds of procedures involved as preprocessing, feature fusion, DBO-based tuning, and SAE-based classification procedure. Figure 1 specifies the process of the TFDD-DBODFF methodology.

3.1. CLAHE-based Preprocessing

Primarily, the CLAHE-based pre-processing is achieved. During the pre-processing stage, resizing and contrast enhancement methods have been executed [25]. The primary stage is used to resize the input and to use the CLAHE technique to increase contrast. The CLAHE method has been proposed to boost image clarity with lower contrast. This can be demonstrated as an efficient tool for establishing and enhancing digital images. Apart from the standard Histogram Equalization (HE) methods, CLAHE has two vital advantages. Firstly, the CLAHE method will be estimated to increase the brightness of distinct pixels more consistently. By implementing conventional HE methods, the histogram level will be extended, which leads to an additional uniform dispersion of grey values through the images. CLAHE improves the contrast at every region; it may result in an increased contrast in the whole image.

$$D_B = \frac{255}{8 \times 8} \sum_{i=0}^{D_A} H\,(i) \qquad (1)$$

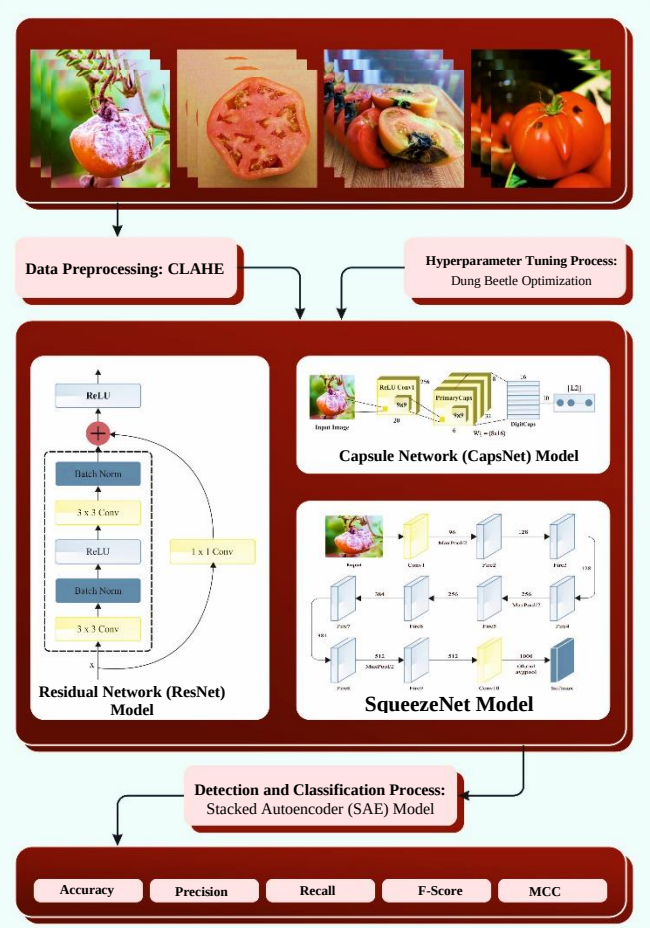

Fig. 1 Workflow of TFDD-DBODFF methodology

Secondarily, by limiting contrast improvement, CLAHE may decrease the complexity of noise improvement. The histogram's height should be enhanced to L to improve the overall visual illustration while retaining the similar histogram field as previously. The last reviewed histogram is shown in Equation (2).

$$H(i) == \begin{cases} H(i) + L, H(i) < H_{max} \\ HH(i)_{maxmax}\{ \end{cases} \qquad (2)$$

CLAHE is an image processing method that can be devised to higher the contrast of an image with the help of increasing the visibility of features. CLAHE performs on tiny, overlapping fields of an image, computing the histograms and reallocating pixel intensities locally. Such an adaptable technique supports enriching the contrast in a particular region of an image with no effect on the whole image consistently. To increase the visibility of features with the images, they may mainly be hidden because of irregular illumination or poorer lighting conditions.

3.2. Feature Fusion Process

The TFDD-DBODFF approach follows a deep feature fusion model containing DL methodologies such as ResNet, CapsNet, and SqueezeNet.

3.2.1. ResNet Model

The authors introduced the ResNet architecture that considerably enhances the efficiency of deep networks with distinctive residual connectivity when related to conventional CNNs [26]. This technique allows the network to simply learn the continual mappings so that it successfully resolves the network degradation issues. The residual block encompasses the residual path and main path. The main path is learning how to map the input information. The residual path has been learned from the residuals of the input about that mapping (X_L, W_L). The residuals must be included in the input to gain the last output (X). It permits the network to directly learn the residual data, hence efficiently resolving the gradient vanishing complexity. The mathematical model will be demonstrated by Equation (3).

$$H(X) = X_L + F(X_L, W_L) \tag{3}$$

Here, W_L denotes the L-layer convolution factor and X_L refers to the L-layer feature mapping.

3.2.2. CapsNet Model

Firstly, Hinton et al. developed the CapsNet, which can address the complexity of losing location data by the pooling layer in conventional CNNs by presenting a capsule module [27]. Conventional CNNs frequently accomplish the preferred outcomes while identifying images in the same way as the training database. Nevertheless, the identification effectiveness of standard CNNs tends to endure any degree of rotation, distortion, or modification in the relevant locations of components in the image. Standard CNNs concentrated on local feature extraction and shortage of robustness for whole image modification. Similarly, pooling operations can blur the positional data more and diminish the performance of images with important spatial differences. By comparing CNNs, CapsNet will proficiently capture the position and spatial correlation of plant leaf spots, which will increase the accuracy of disease detection. The hierarchical architecture of CapsNets supports learning additional hierarchical feature representations and increases the capability for differentiation among diverse disease categories. CapsNets have a few drawbacks in several features, comprising the comparative simplicity of the feature extraction model and restrictions for larger-size images. Primarily, the feature extraction model of CapsNets is comparatively easy. Convolutional layers are frequently employed as feature extractors in conventional CapsNets, and this model executed well while dealing with easier image tasks. However, once regarding additional intricate image tasks, the feature extraction networking model can be moderately shallow and challenging to take deeper image features. Secondarily, standard CapsNets have been commonly developed to be trained and tested for fixed-size images. For instance, the popular MNIST database employs images of 28x28 pixels. This restriction describes how CapsNets might have difficulty with huge images. Images with huge input sizes are required to be modified and processed to fit the architecture of the CapsNet.

The CapsNet comprises a convolutional, main capsule, and digital capsule layers. The convolutional layer is accountable for removing the local features. Alternatively, the main capsule layer presents capsule modules for capturing the image's spatial model and position data at a hierarchical correlation. The output of capsule components is the existence and feature data; thereby, the network will retain the spatial data of the objects more precisely. The digital capsule layer integrates the outputs of various capsule modules to create a wide-ranging representation of the overall objective. This model permits CapsNet to demonstrate superior robustness after dealing with images with important spatial differences, particularly in the existence of distortions, rotations, and modified ions in the relevant positions of the elements that will increase the efficiency of identification. The dynamic routing mechanism is a significant feature of CapsNet that iteratively upgrades the weights to optionally allocate the capsule's outputs in the preceding layer to the capsules at the subsequent layer. This method has been accomplished automatically in training, progressively permitting the network to learn the accurate routing technique. Particularly for the dynamic routing weight C_{ij}, a sofnax function was employed to create the sum of the weights 1. Afterwards, applying the weights attained in the prior stage, the outputs of the leading capsule will be weighted and added to gain the weighted sum S_j. In conclusion, a nonlinear activation function was employed for getting the output of the j^{th} statistical capsule and then the dynamic routing, v_j. The above-mentioned three stages have been looped and iterated until convergence. The mathematical forms is illustrated in Equation (4).

$$C_{ij} = \frac{exp(b_{ij})}{\Sigma_k exp(b_{ij})} \tag{4}$$

$$S_j = \sum_i C_{ij} \cdot u_{ij} \tag{5}$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \tag{6}$$

Here C_{ij} means the dynamic routing weight, v_j defines the resultant of high-level capsule j, u_{ij} represents the output vector of the low-level capsule i, b_{ij} refers to the initialized marginal probability and S_j denotes the input of a higher-level capsule j.

3.2.3. SqueezeNet Model

SqueezeNet is a network structured with 1.24 M parameters and 18-layer depth [28]. The strategy aims to attain greater accuracy by optimizing network parameters. Ultimately, the precision of AlexNet architecture can be accomplished under ImageNet through SqueezeNet with $50\times$ some parameters. SqueezeNet has been developed for processing the images at a spatial resolution (SR) of $227 \times 227 \times 3$. The model will utilize the input SR to $224 \times 224 \times 3$ for similarity with alternative networks.

Additionally, each layer next to the depth concatenation layer, like fire9-concat, has been extracted. Consequently, the SR of the activation maps produced through the layer of fire9-concat is represented as $13 \times 13 \times 512$. A max-pooling layer is connected to the specified step and pooling dimensions. In conclusion, a feature mapping generator model could be attained to provide 512 activation maps at 7×7 SRs.

3.3. DBO-based Hyperparameter Tuning

During this stage, the DBO model performs the hyperparameter tuning of these DL approaches. The DBO is a recent SI optimization technique based on the behaviors of the dung beetle, namely ball-rolling, foraging, breeding, dancing, and stealing [29]. This model is designed to handle the problems of constrained and unconstrained optimization. The four sub-populations of the dung beeltes are ball-rolling, breeding, small, and stealing. The mathematical modeling is discussed in the following.

3.3.1. Ball-Rolling Behavior

It is noted that there are any obstacles during the process of ball-rolling will make dung beetles behave differently. Based on the following equation, the ball-rolling dung beetles update the positional information once they proceed to search according to the direction of the sun without obstacles:

$$x_i^{g+1} = x_i^g + a \times k \times x_i^{g-1} + b \times \left| x_i^g - x_{worst}^g \right| \quad (7)$$

The number of existing iterations is represented as g, the positional information of i^{th} dung beetles in the population at the g^{th} iteration is indicated by x_i^g, an invariant quantity indicating the flexure coefficient is represented as $k \in (0,0.2]$, a fixed parameter belonging to (0,1) is represented as b, the natural coefficient allocated 1 or- 1 representing deviation or no deviation from the original direction is formulated as a correspondingly, the global worst position at the g^{th} iteration is denoted by x_{worst}^g, $|x_i^g - x_{worst}^g|$ simulates the changes in the intensity of light.

Based on a tangent function belonging to $[0, \pi]$, dung beetles adjust their rolling direction by dancing when there are any obstacles preventing dung beetles from processing. Using Equation (8), the positional information of the ball-rolling dung beetles is updated.

$$x_i^{g+1} = x_i^g + \tan(\theta) \left| x_i^g - x_i^{g-1} \right| \quad (8)$$

The distance between i^{th} dung beetles at g^{th} iteration and at $(g-1)^{th}$ iteration is denoted as $|x_i^g - x_i^{g-1}|$. The location of the dung beetle will not be updated if the value of θ takes 0, $\frac{\pi}{2}$, or π.

3.3.2. Breeding Behavior

A frontier option technique simulates the area where the egg is produced based on the female dung beetles' behavior to choose an appropriate location to lay the eggs to provide a safer environment for the offspring:

$$\begin{cases} X^{Lb^*} = \max\{X^* \times (1-R), X^{Lb}\} \\ X^{Ub^*} = \min\{X^* \times (1+R), X^{Ub}\} \end{cases} \quad (9)$$

In Equation (9), the present local optimum location is represented as X^*, X^{Lb^*} and X^{Ub^*} are the Lower Boundary (LB) And Upper Boundary (UB) of the search space, X^{Lb} and X^{Ub} are the LB and UB of the problem space, $R = 1 - g/G$, and G are the maximum iteration count. Note that the female dung beetle produces only one egg at every iteration. The positional information of the female dung beetles laying eggs is dynamic for the boundary range during the iteration.

$$x_i^{g+1} = X^* + b_1 \times \left(x_i^g - X^{Lb^*} \right) + b_2 \times \left(x_i^g - X^{Ub^*} \right) \quad (10)$$

In Equation (10), the positional information of the $i^{\tau h}$ brood balls at the $g^{\tau h}$ iteration is represented as $\chi_i^{(q}$, b_1, and b_2 are two random and independent vectors with the $1 \times D$ size, and the number of dimensions in the problem is represented as D.

3.3.3. Foraging Behavior

Mature dung beetles, known as small dung beetles, emerge from the ground to search for food, with their search area dynamically adjusted as iterations progress.

$$\begin{cases} X^{Lb^b} = \max\{X^b \times (1-R), X^{Lb}\} \\ X^{Ub^b} = \min\{x^b \times (1+R), X^{Ub}\} \end{cases} \quad (11)$$

In Equation (11), the global optimum location is represented as X^b, and the LB and UB of the search area are X^{Lb^b} and X^{Ub^b}. The equation to update the positional information of small dung beetles is given by:

$$x_i^{g+1} = x_i^g + C_1 \times \left(x_i^g - X^{Lb^b} \right) + C_2 \times \left(x_i^g - X^{Ub^b} \right) \quad (12)$$

In Equation (12), the location of the i^{th} small dung beetles at the g^{th} iteration is denoted as x_i^g, uniform distribution random number within the range $(0,1)$ with the size of $1 \times D$ is indicated as $C_1$and C_2.

3.3.4. Stealing Behavior

The equation for updating the thief's position, which steals dung balls from others, can be formulated by:

$$x_i^{g+1} = X^b + S \times t \times \left(\left| x_i^g - X^* \right| + \left| x_i^g - X^b \right| \right) \quad (13)$$

In Equation (13), the positional information of the i^{th} thief at the g^{th} iteration is denoted as x_i^g, a normal distribution random vector with the $1 \times D$ size is represented as r, and S is a fixed parameter. The fitness choice is the main aspect of managing the solution of the DBO method. The parameter selection process comprises analyzing candidate solutions based on their encoded performance metrics. The DBO technique employs accuracy as the primary criterion to define the fitness function (FF), which is expressed as:

$$Fitness = \max(P) \quad (14)$$

$$P = \frac{TP}{TP + FP} \quad (15)$$

Where FP and TP illustrate the false and true positive rates.

3.4. Fruit Disease Classification using SAE

Finally, the classification and recognition of tomato fruit diseases take place using SAE. SAE is a DL neural network model encompassing encoders and decoders [30]. The goal is to learn the compressed form of the input dataset for the representation learning or dimensionality reduction tasks. The encoder transforms the input dataset into a low-dimensional latent representation or bottleneck layer.

Then, the decoder is used to map the low-dimensional into the original input dataset. The trained model minimizes the gap between the original input datasets and the decoder output, ensuring the output closely matches the input. The loss function reduces the difference. Figure 2 depicts the infrastructure of SAE.

The AE with multiple layers (N) and the input dataset with (D) dimension, then loss function, encoder, and decoder are evaluated by the following expression.

$The\ encoder$ assumes each layer, and the encoder function is described by:

$$en_i = act_i\left(wt_i * op_j - 1 + bias_i\right) \quad (16)$$

In Equation (16), output en_i is the encoder function, act_i refers to the activation function, wt_i indicates the weight matrix, $op_j - 1$ refers to an output of the prior layer and the bias vector of $the\ i^{th}$ layer is $bias_i$.

$Decoding$ The decoder function of the i^{th} layer is described by the following expression:

$$de_i = dact_i(dwt_i * den_i + dbias_i) \quad (17)$$

In Equation (17), de_i denotes the decoder function output, $dact_i$ is the activation function, den_i is the input from the prior encoder layer, dwt_i and $dbias_i$ indicate the weight matrix and the bias vector of i^{th} decoder layer.

$Loss\ function$ The MSE among the input as well as output data of the last decoder layer acts as a loss function and is represented as follows:

$$L = \frac{1}{(2N)} * \|I - dOUT\|^2 \quad (18)$$

Where I show the input, $dOUT$ refers to the last layer output, and $* \|. \|^2$ refers to the squared Euclidean norm.

The training is performed after the allocation of both functions, and a trained network is used for extraction later.

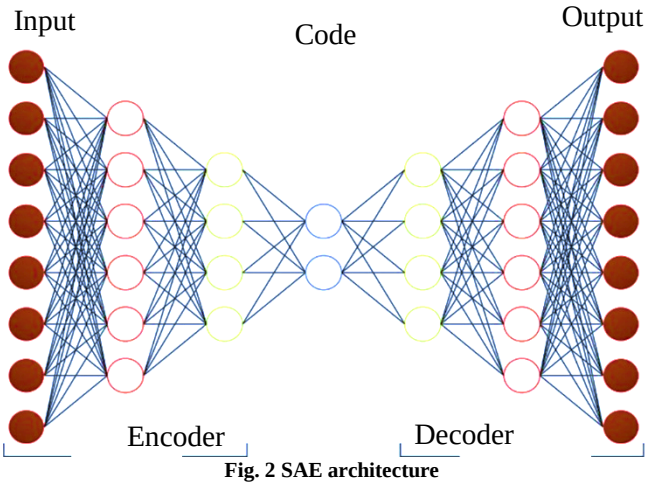

Fig. 2 SAE architecture

4. Performance Validation

The TFDD-DBODFF methodology is examined using the tomato fruit disease dataset collected from various sources. The dataset involves data augmentation in different ways, such as 150-450 (Rotation and Scaling). Table 1 represents a detailed description of the dataset. Figure 3 demonstrates the sample images.

Fig. 3 Sample images

Table 1. Dataset specification

Diseases	No. of Images
Healthy	450
Malformed fruit	450
Blotchy ripening	450
Puffy fruit	450
Blossom-end rot	450
Gray mold	450
Total Images	900

Figure 4 establishes the confusion matrices created by the TFDD-DBODFF approach at 80:20 and 70:30 of TRAS/TESS. The simulation outputs implied that the TFDD-DBODFF approach efficiently detected and classified six classes. Table 2 and Figure 5 depict the evaluation of the TFDD-DBODFF method on 80:20 TRAS/TESS.

The outputs infer that the TFDD-DBODFF method gains effective identification of healthy and diseased fruits. On 80% TRAS, the TFDD-DBODFF technique offers an average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 95.94%, 87.81%, 87.82%, 87.81%, and 85.38%, correspondingly. Also, on 20% TESS, the TFDD-DBODFF approach offers average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 96.79%, 90.47%, 90.38%, 90.39%, and 88.49%, correspondingly.

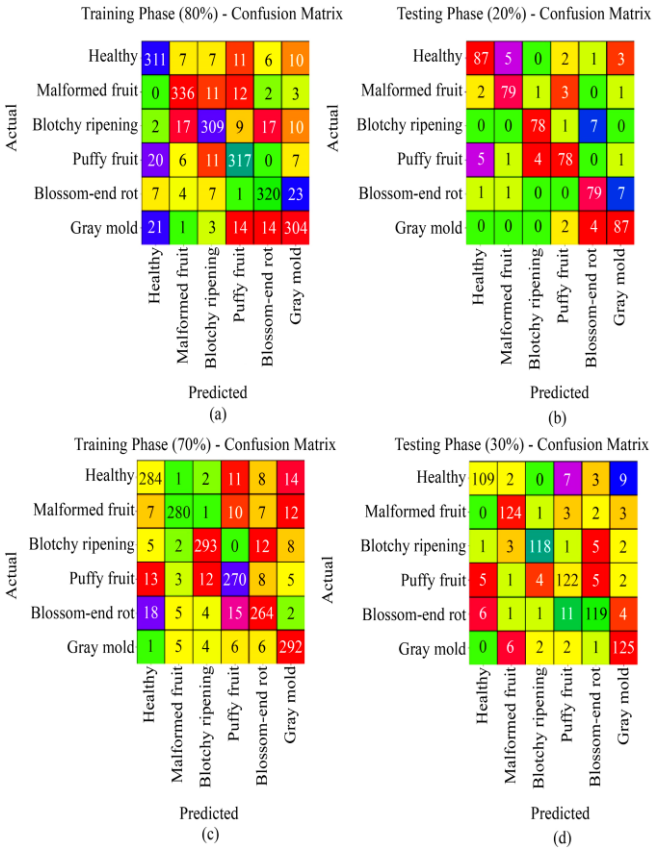

Fig. 4 Confusion matrices of TFDD-DBODFF technique (a-b) 80:20 and (c-d) 70:30

The performance of the TFDD-DBODFF method is graphically presented in Figure 6 under training accuracy (TRAA) and validation accuracy (VALA) curves on 80:20 TRAS/TESS. The outcome of the TFDD-DBODFF method over distinct epochs depicts its learning and generalization capabilities. Notably, a continuous enhancement with an epoch surge is also portrayed. It underscores the adaptive behaviour of the TFDD-DBODFF model in pattern detection on overall data. The growth in VALA emphasizes the ability of the TFDD-DBODFF model to adapt to TRA and outperform in giving an accurate classifier on unseen data, noting the robust generalized capabilities.

Table 2. Classifier outcome of TFDD-DBODFF method on 80:20 TRAS/TESS

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAS (80%)					
Healthy	95.79	86.15	88.35	87.24	84.72
Malformed fruit	97.08	90.57	92.31	91.43	89.68
Blotchy ripening	95.65	88.79	84.89	86.80	84.22
Puffy fruit	95.79	87.09	87.81	87.45	84.92
Blossom-end rot	96.25	89.14	88.40	88.77	86.52
Gray mold	95.09	85.15	85.15	85.15	82.21
Average	**95.94**	**87.81**	**87.82**	**87.81**	**85.38**
TESS (20%)					
Healthy	96.48	91.58	88.78	90.16	88.03
Malformed fruit	97.41	91.86	91.86	91.86	90.32
Blotchy ripening	97.59	93.98	90.70	92.31	90.90
Puffy fruit	96.48	90.70	87.64	89.14	87.06
Blossom-end rot	96.11	86.81	89.77	88.27	85.96
Gray mold	96.67	87.88	93.55	90.62	88.66
Average	**96.79**	**90.47**	**90.38**	**90.39**	**88.49**

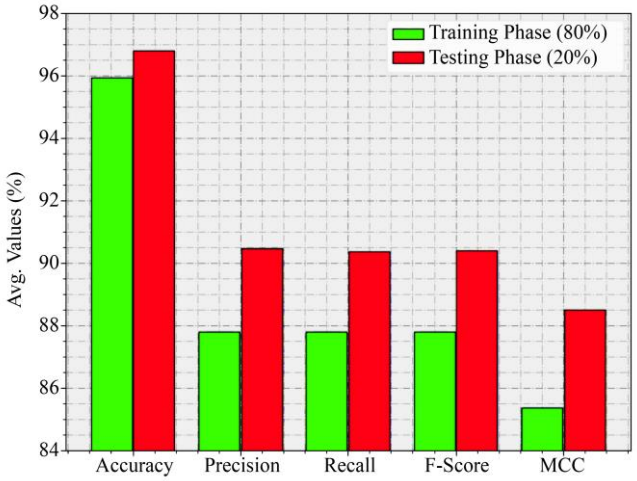

Fig. 5 Average of TFDD-DBODFF method on 80:20 TRAS/TESS

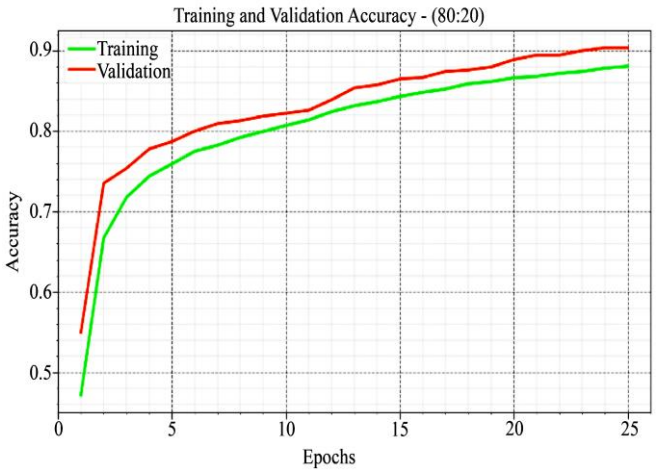

Fig. 6 $Accu_y$ curve of TFDD-DBODFF model on 80:20 TRAS/TESS

Figure 7 illustrates an overall depiction of the training loss (TRLA) and validation loss (VALL) solutions of the TFDD-DBODFF methodology over various epochs on 80:20 TRAS/TESS. The reduction in TRLA shows how the TFDD-DBODFF methodology optimizes weights and decreases classification errors on TRAS/TESS data, effectively capturing patterns and improving alignment with actual TRA classes. Examining the PR, as demonstrated in Figure 8, the outputs confirmed that the TFDD-DBODFF approach exhibits higher PR values under each class on 80:20 TRAS/TESS. It shows the improved ability of the TFDD-DBODFF approach in the identification of several classes, displaying proficient class recognition.

Additionally, in Figure 9, ROC curves produced by the TFDD-DBODFF approach are exhibited in the classification of distinct labels on 80:20 TRAS/TESS. It provides details into the tradeoff between TPR/FRP across diverse thresholds and epochs. The outputs emphasize that the TFDD-DBODFF technique outperforms in overall classes, accentuating its efficiency in handling diverse threats.

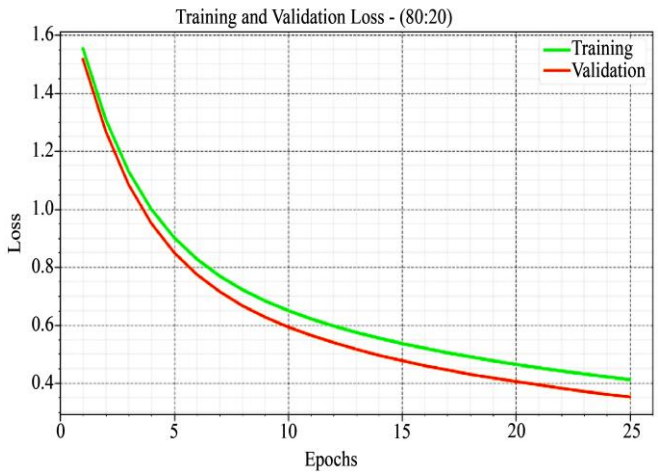

Fig. 7 Loss curve of TFDD-DBODFF model on 80:20 TRAS/TESS

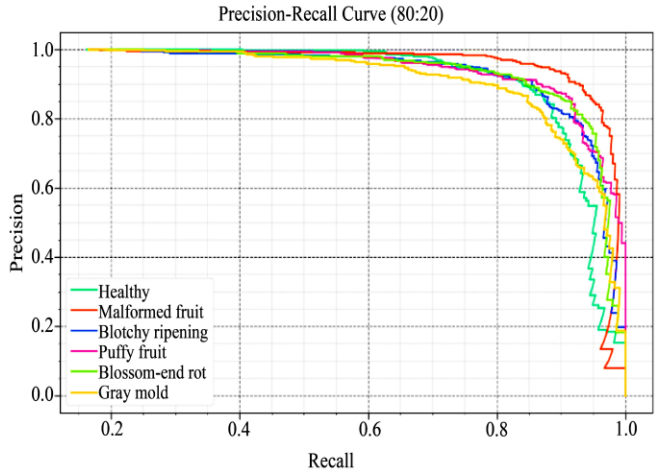

Fig. 8 PR curve of TFDD-DBODFF model on 80:20 TRAS/TESS

Table 3 and Figure 10 report a detailed outcome analysis of the TFDD-DBODFF approach on 70:30 TRAS/TESS. The outputs illustrate that the TFDD-DBODFF approach attains effectual detection of healthy and diseased fruits. On 70% TRAS, the TFDD-DBODFF methodology offers an average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 96.35%, 89.11%, 89.03%, 89.04%, and 86.87%, correspondingly. Moreover, on 30% TESS, the TFDD-DBODFF methodology presents an average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 96.17%, 88.69%, 88.56%, 88.57%, and 86.31%, correspondingly.

The performance of the TFDD-DBODFF technique is graphically projected in Figure 11 under TRAA/VALA curves on 70:30 TRAS/TESS. The figure specifies the learning and generalization of the TFDD-DBODFF technique across several epochs, depicting consistent enhancement in TRAA/VALA as epochs progress. This portrays the adaptability of the TFDD-DBODFF model in pattern detection and its robust generalization capabilities, particularly in precisely classifying unseen data.

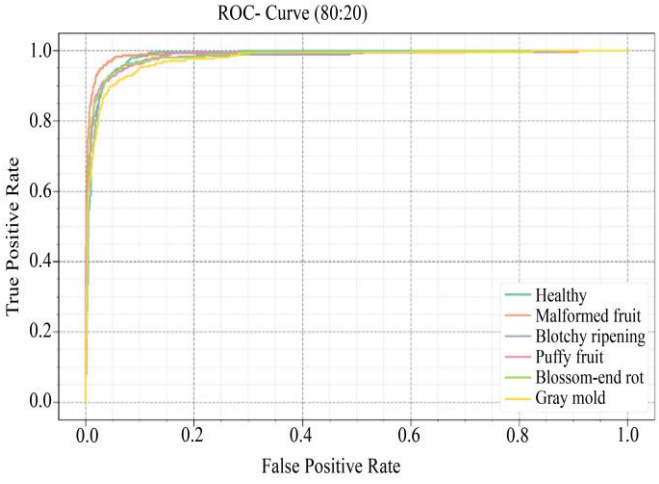

Fig. 9 ROC curve of TFDD-DBODFF model on 80:20 TRAS/TESS

Table 3. Classifier outcome of TFDD-DBODFF technique on 70:30 TRAS/TESS

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAS (70%)					
Healthy	95.77	86.59	88.75	87.65	85.11
Malformed fruit	97.20	94.59	88.33	91.35	89.76
Blotchy ripening	97.35	92.72	91.56	92.14	90.55
Puffy fruit	95.61	86.54	86.82	86.68	84.05
Blossom-end rot	95.50	86.56	85.71	86.13	83.45
Gray mold	96.67	87.69	92.99	90.26	88.31
Average	**96.35**	**89.11**	**89.03**	**89.04**	**86.87**
TESS (30%)					
Healthy	95.93	90.08	83.85	86.85	84.52
Malformed fruit	97.28	90.51	93.23	91.85	90.24
Blotchy ripening	97.53	93.65	90.77	92.19	90.74
Puffy fruit	94.94	83.56	87.77	85.61	82.58
Blossom-end rot	95.19	88.15	83.80	85.92	83.06
Gray mold	96.17	86.21	91.91	88.97	86.72
Average	**96.17**	**88.69**	**88.56**	**88.57**	**86.31**

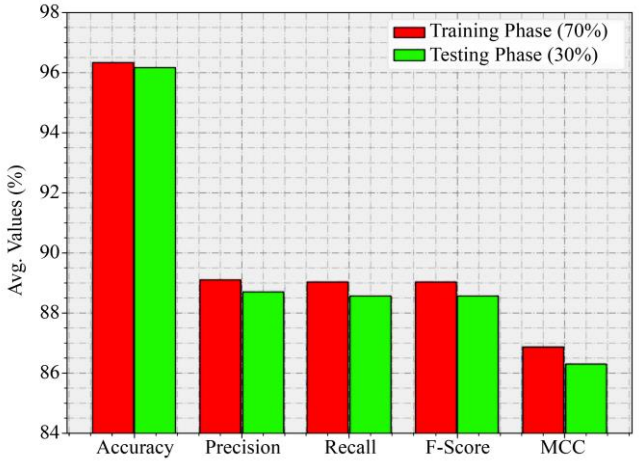

Fig. 10 Average of TFDD-DBODFF technique on 70:30 TRAS/TESS

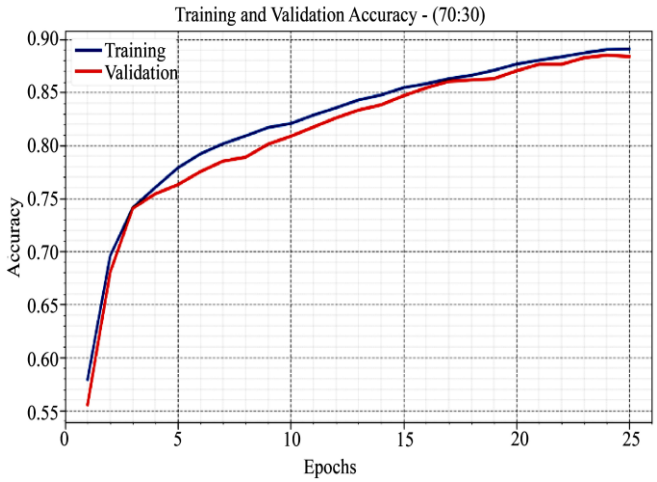

Fig. 11 $Accu_y$ the curve of the TFDD-DBODFF approach on 70:30 TRAS/TESS

Figure 12 shows the outcomes of TRLA/VALL of the TFDD-DBODFF approach over discrete epochs on 70:30 TRAS/TESS. The decreasing TRLA depicts how the TFDD-DBODFF approach optimizes weights and mitigates classification errors on TRA/TES data. The figure portrays the effectualness of the TFDD-DBODFF technique in capturing patterns in both datasets and its ongoing enhancement in reducing the variations between predicted and actual TRA classes. Scrutinizing the PR, as revealed in Figure 13, the outputs confirmed that the TFDD-DBODFF approach gradually realizes maximum PR values under every class on 70:30 TRAS/TESS. It examines the better capabilities of the TFDD-DBODFF method in detecting diverse classes, emphasizing the ability to recognize classes. Additionally, in Figure 14, ROC curves created by the TFDD-DBODFF approach are exhibited in the classification of various labels on 70:30 TRAS/TESS. It presents details into the tradeoff between TPR/FRP across diverse detection thresholds and epochs. The outputs depict that the TFDD-DBODFF methodology attains greater classifier performance across overall classes, underlining its efficiency in solving several classification issues.

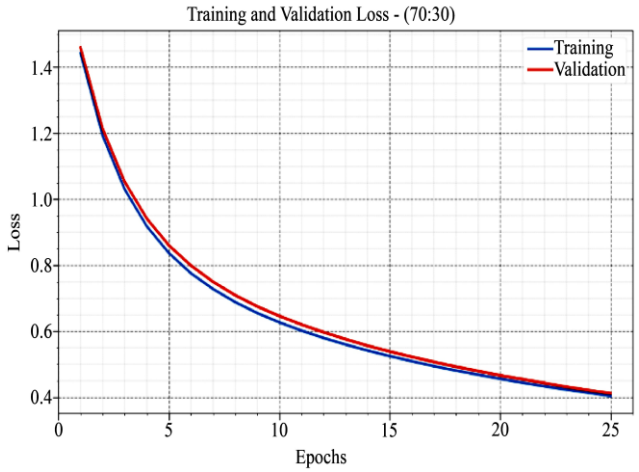

Fig. 12 Loss curve of TFDD-DBODFF approach on 70:30 TRAS/TESS

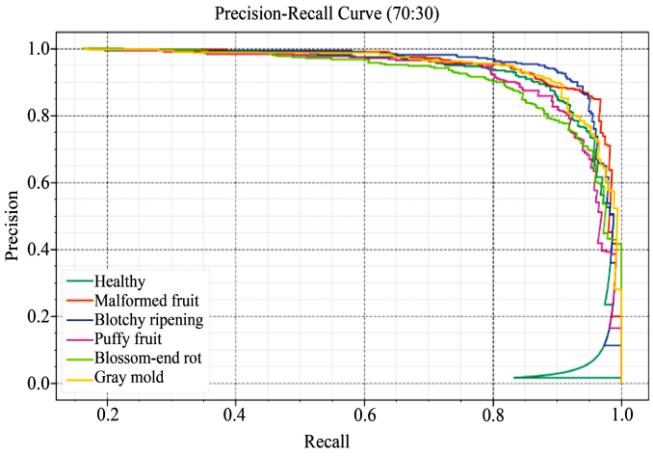

Fig. 13 PR curve of TFDD-DBODFF approach on 70:30 TRAS/TESS

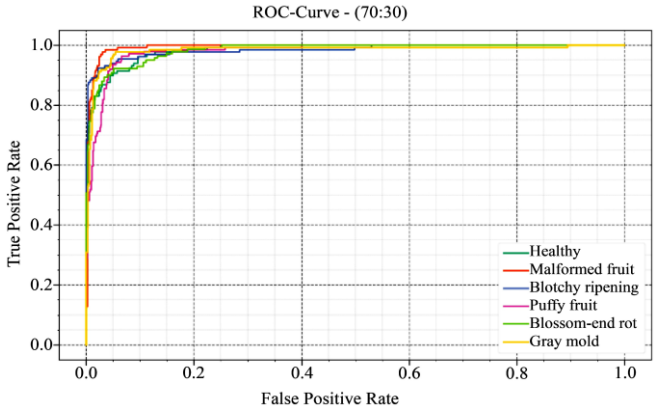

Fig. 14 ROC curve of TFDD-DBODFF approach on 70:30 TRAS/TESS

Table 4. Comparative evaluation of TFDD-DBODFF approach with existing methods

Model	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
Yolov5m	90.59	80.61	80.84	82.14
ResNet-50	93.20	87.14	86.80	87.92
ResNet-101	94.22	83.81	80.00	83.18
EfficientNet-B0	91.93	82.89	86.07	87.54
VGG-16	93.07	85.81	86.30	81.12
MobileNet	91.54	82.18	80.64	82.38
TFDD-DBODFF	96.79	90.47	90.38	90.39

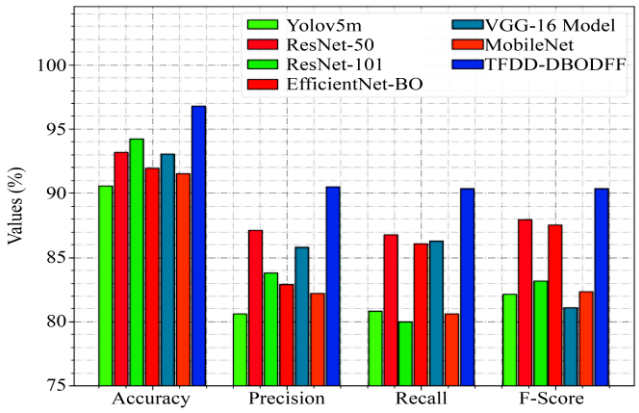

Fig. 15 Comparative analysis of TFDD-DBODFF technique with recent models

The results of the TFDD-DBODFF technique are associated with recent DL methods in Table 4 and Figure 15 [10, 31]. The outputs represent that the EfficientNet-B0, YOLOV5m, and MobileNet models have shown the lowest performance. Furthermore, the ResNet-50, ResNet-101, and VGG-16 models reported closer results.

However the TFDD-DBODFF technique demonstrates significant performance with increased $accu_y$ of 96.79%, $prec_n$ of 90.47%, $reca_l$ of 90.38%, and F_{score} of 90.39%. Thus, the TFDD-DBODFF technique can be used to effectively recognize diseases in tomato fruits.

5. Conclusion

In this study, a new TFDD-DBODFF methodology is introduced. The TFDD-DBODFF model aims to enhance the recognition of the results of tomato fruit diseases. It contains different kinds of procedures involved as preprocessing, feature fusion process, DBO-based hyperparameter tuning, and SAE-based classification process. Primary, the CLAHE-based preprocessing is performed. Besides, the TFDD-DBODFF technique follows a deep feature fusion model containing 3 DL approaches: ResNet, CapsNet, and SqueezeNet. In addition, the DBO model performs the hyperparameter tuning of these DL approaches.

Finally, the classification and detection of tomato fruit diseases take place using the SAE model. The investigational output of the TFDD-DBODFF technique can be examined using a benchmark dataset. The investigational outputs indicate a higher solution for the TFDD-DBODFF technique with existing models in terms of different measures.

The existing models face limitations in scalability and adaptability to diverse datasets, potentially affecting their performance in varying contexts. Future work should concentrate on improving generalization capabilities and integrating advanced techniques for more robust feature extraction and tuning. Additionally, exploring more flexible classification methods could address current limitations and improve overall accuracy.

References

[1] Hafedh Mahmoud Zayani et al., "Deep Learning for Tomato Disease Detection with YOLOv8," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13584-13591, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Anuradha Chug et al., "A Novel Framework for Image-Based Plant Disease Detection Using Hybrid Deep Learning Approach," *Soft Computing*, vol. 27, no. 18, pp. 13613-13638, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Aichen Wang et al., "NVW-YOLOv8s: An Improved YOLOv8s Network for Real-Time Detection and Segmentation of Tomato Fruits at Different Ripeness Stages," *Computers and Electronics in Agriculture*, vol. 219, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Md Shofiqul Islam et al., "Multimodal Hybrid Deep Learning Approach to Detect Tomato Leaf Disease Using Attention Based Dilated Convolution Feature Extractor with Logistic Regression Classification," *Sensors*, vol. 22, no. 16, pp. 1-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Antonio Guerrero-Ibañez, and Angelica Reyes-Muñoz, "Monitoring Tomato Leaf Disease through Convolutional Neural Networks," *Electronics*, vol. 12, no. 1, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] V. Shwetha, Arnav Bhagwat, and Vijaya Laxmi, "LeafSpotNet: A Deep Learning Framework for Detecting Leaf Spot Disease in Jasmine Plants," *Artificial Intelligence in Agriculture*, vol. 12, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mohammed Saeed Alzahrani, and Fawaz Waselallah Alsaade, "Transform and Deep Learning Algorithms for the Early Detection and Recognition of Tomato Leaf Disease," *Agronomy*, vol. 13, no. 5, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] S. Swapna Rani et al., "Development and Evaluation of a Distinctive Cloud-Based Artificial Intelligence System Using Deep Learning Techniques (AISDLT) for Accurate Detection of Tomato Plant Leaf Diseases," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 12s, pp. 538-552, 2024. [Publisher Link]

[9] Marriam Nawaz et al., "A Robust Deep Learning Approach for Tomato Plant Leaf Disease Localization and Classification," *Scientific Reports*, vol. 12, no. 1, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Quoc-Hung Phan et al., "Classification of Tomato Fruit using Yolov5 and Convolutional Neural Network Models," *Plants*, vol. 12, no. 4, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Anu, Kamna Solanki, and Amita Dhankar, "A Novel Multi-Class Deep Learning Approach for Tomato Leaf Disease Detection System," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 6s, pp. 187-196, 2024. [Google Scholar] [Publisher Link]

[12] Rina Bora, Deepa Parasar, and Shrikant Charhate, "A Detection of Tomato Plant Diseases Using Deep Learning MNDLNN Classifier," *Signal, Image and Video Processing*, vol. 17, no. 7, pp. 3255-3263, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Muhammad Shoaib et al., "Deep Learning-Based Segmentation and Classification of Leaf Images for Detection of Tomato Plant Disease," *Frontiers in Plant Science*, vol. 13, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Lobna M. Abouelmagd et al., "An Optimized Capsule Neural Networks for Tomato Leaf Disease Classification," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] N. Aishwarya et al., "Smart Farming for Detection and Identification of Tomato Plant Diseases Using Light Weight Deep Neural Network," *Multimedia Tools and Applications*, vol. 82, no. 12, pp. 18799-18810, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] I. Sheik Arafat, S. Aswath, and S.M. Haji Nishath, "Automatic Early Detection of Tomato Leaf Disease Using IoT and Deep Learning," *Research Article*, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Showmick Guha Paul et al., "A Real-Time Application-Based Convolutional Neural Network Approach for Tomato Leaf Disease Classification," *Array*, vol. 19, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Rajasekaran Thangaraj et al., "A Deep Convolution Neural Network Model Based on Feature Concatenation Approach for Classification of Tomato Leaf Disease," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 18803-18827, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yuanbo Ye et al., "Application of Tswin-F Network Based on Multi-Scale Feature Fusion in Tomato Leaf Lesion Recognition," *Pattern Recognition*, vol. 156, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Himanshu B. Patel, and Nitin J. Patil, "Enhanced CNN for Fruit Disease Detection and Grading Classification Using SSDAE-SVM for Postharvest Fruits," *IEEE Sensors Journal*, vol. 24, no. 5, pp. 6719-3732, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hao Sun et al., "YOLO-FMDI: A Lightweight YOLOv8 Focusing on a Multi-Scale Feature Diffusion Interaction Neck for Tomato Pest and Disease Detection," *Electronics*, vol. 13, no. 15, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Zheng Li et al., "Tomato Leaf Disease Recognition via Optimizing Deep Learning Methods Considering Global Pixel Value Distribution," *Horticulturae*, vol. 9, no. 9, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Pandurangan Indumathi, and Rathinasamy Kumuthaveni, "Coati Optimized Transfer Leaning with Vision Transformer Model for Improving Deep Learner Based Plant Diseases Detection," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 4, pp. 277-288, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Malathi Chilakalapudi, and Sheela Jayachandran, "Multi-Classification of Disease Induced in Plant Leaf Using Chronological Flamingo Search Optimization with Transfer Learning," *PeerJ Computer Science*, vol. 10, pp. 1-29, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Chunduri Madhurya, and Emerson Ajith Jubilson, "YR2S: Efficient Deep Learning Technique for Detecting and Classifying Plant Leaf Diseases," *IEEE Access*, pp. 3790-3804, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Wanni Xu, You-Lei Fu, and Dongmei Zhu, "Resnet and Its Application to Medical Image Processing: Research Progress and Challenges," *Computer Methods and Programs in Biomedicine*, vol. 240, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Asif Raza et al., "Diverse Capsules Network Combining Multiconvolutional Layers for Remote Sensing Image Scene Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5297-5313, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] Coşku Öksüz, Oğuzhan Urhan, and Mehmet Kemal Güllü, "An Integrated Convolutional Neural Network with Attention Guidance for Improved Performance of Medical Image Classification," *Neural Computing and Applications*, vol. 36, no. 4, pp. 2067-2099, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Zhenhui Lu, "An Adaptive Dung Beetle Optimization Algorithm with Golden Sine for Optimizing Numerical Unconstrained

Problems," *Current Journal of Applied Science and Technology*, vol. 43, no. 4, pp. 12-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] Muhammad Sami Ullah et al., "Brainnet: A Fusion-Assisted Novel Optimal Framework of Residual Blocks and Stacked Autoencoders for Multimodal Brain Tumor Classification," *Scientific Reports*, vol. 14, no. 1, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] Qimei Wang et al., "Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques," *Computational Intelligence and Neuroscience*, vol. 20191, no. 1, pp. 1-15, 2019. [CrossRef] [Google Scholar] [Publisher Link]