IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 January 2025, accepted 6 February 2025, date of publication 13 February 2025, date of current version 19 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3541347

RESEARCH ARTICLE

Multi-Stage Neural Network-Based Ensemble Learning Approach for Wheat Leaf Disease Classification

SAMIA NAWAZ YOUSAFZAI[1], INZAMAM MASHOOD NASIR[2], (Member, IEEE), SARA TEHSIN[2], (Student Member, IEEE), DANIA SALEEM MALIK[3], ISMAIL KESHTA[4], (Member, IEEE), NORMA LATIF FITRIYANI[5], YEONGHYEON GU[5], AND MUHAMMAD SYAFRUDIN[5]

[1]Applied INTelligence Lab (AINTLab), Seoul 05006, Republic of Korea
[2]Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania
[3]Department of Mathematics, HITEC University Taxila, Taxila, Pakistan
[4]Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia
[5]Department of Artificial Intelligence and Data Science, Sejong University, Seoul 05006, Republic of Korea

Corresponding authors: Yeonghyeon Gu (yhgu@sejong.ac.kr) and Muhammad Syafrudin (udin@sejong.ac.kr)

This work was carried out with the funding of ''Cooperative Research Program for Agriculture Science and Technology Development (Project No. RS-2021-RD010360, Development of pests and plant diseases diagnosis using intelligent image recognition)'' Rural Development Administration, Republic of Korea.

ABSTRACT Prompt and accurate classification of wheat leaf diseases and their severity is essential for precise diagnosis, effective pesticide application, efficient disease control, and enhanced wheat production and quality. Nevertheless, the wide range of wheat diseases poses a significant challenge in terms of their identification, especially in intricate agricultural landscapes. The utilization of conventional models has major limitations in wheat disease detection, including dataset-specific performance, overfitting due to limited data, and high computational needs, making deployment in resource-constrained situations difficult. To address these challenges, this research proposes a multi-stage Convolutional Neural Network (CNN) based Ensemble Learning (EL) approach, which utilizes several concurrent CNN models with a bagged EL method to classify wheat leaf diseases. The various stages in the EL approach employ a multi-level framework to enhance feature extraction and capture complex data patterns. This improves the model's emphasis on diseases while reducing the influence of complex backgrounds on disease identification. The proposed method, integrating pretrained CNNs and a bagging ensemble technique, achieved an accuracy of 86.78%, 98.28%, and 99.16% on the three publicly available datasets, outperforming the state-of-the-art models. These results demonstrate the potential of the proposed model for real-time disease diagnosis.

INDEX TERMS Wheat disease detection, ensemble learning, image classification, convolutional neural networks, bagging.

I. INTRODUCTION

Globally, wheat is a widely consumed cereal crop, alongside maize and rice serving as a significant source of food. Wheat provides approximately 20% of the global dietary protein and calorie intake. Several diseases potentially impact both the quantity and quality of wheat crops [1]. Diseases currently account for 26-30% of yearly wheat losses, but without adequate application of protection technologies to manage fields, that number may climb to 70% [2]. Among other diseases, fungal diseases have the greatest wheat yield loss of 15% to 20% worldwide [3]. Wheat production is negatively impacted by a range of fungal diseases, including various forms of rusts, septoria, and powdery mildew (see Fig. 1) [4]. The proliferation of these diseases has significantly hampered wheat farming and resulted in substantial financial losses

The associate editor coordinating the review of this manuscript and approving it for publication was Sawyer Duane Campbell.

for farmers. The management of crop diseases has evolved into a major challenge. Hence, early disease detection and identification emerges as the highly effective method to deal with these diseases, maximize crop yield and improve food security.

Historically, the extraction of characteristics associated with agricultural leaf diseases relied heavily on manual design techniques. Although these methods can successfully accomplish classification, they usually necessitate the segmentation of affected areas or foliage [5]. This procedure exacerbates the burden during the early stages and requires the development of separate feature extraction methods for each condition, resulting in difficulties in differentiating comparable disorders. Furthermore, identification models that are based on conventional machine learning methods not only depend heavily on the extraction of specific features but also require the creation of various recognition models for different crop diseases, which results in a lack of universal applicability [6].

In past decades, the identification and detection of wheat diseases relied heavily on manual techniques, which are not only labor intensive but also subjective, inefficient, and least accurate. The phenomenon of image analysis has been the subject of substantial research in previous studies [7], [8], [9], [10], [11] since it has been recognized as an efficient method for enhancing the representation capabilities of Convolutional Neural Networks (CNNs) in multiple domains. CNNs are also used for object recognition [12], detection of human movement [13], target detection [14] and person identification [15]. With the emergence of technology, the detection of wheat diseases has become increasingly reliant on Machine Learning (ML) [16] and Deep Learning (DL) [17]. Currently, many DL techniques are being widely used for the purpose of disease identification for various plant species [18], [19], [20]. CNNs have also achieved tremendous success in recent times [21], [22], [23], [24], [25]. Recently, researchers have employed CNNs in human action recognition [26], [27], statistical models [28], [29] and other medical imaging domains [30], [31], [32]. Existing DL methodologies utilized CNNs along with several architecture options such as AlexNet and VGG. Furthermore, the advancement of DL techniques for improved disease identification in plants encompassed the implementation of various architectures [33], [34], [35], [36], transfer learning methodologies [37] and Bayesian DL [38].

Most of the recent research has been focused on the task of diagnosing fungi disease while employing DL architectures has witnessed a notable advancement. Nevertheless, there are still areas that require further investigation related to the application of emerging DL architectures, particularly in the context of detecting fungal diseases in wheat. It is noticeable that the techniques employed in earlier studies produced adequate outcomes when applied to a single dataset. However, when these approaches were tested on several datasets, a decline in performance was observed, and therefore, these methodologies are not suited for practical implementation in the agricultural field [39]. Since the detection of wheat diseases in real-world cases is hindered by several challenges, including similar symptoms of different diseases, low-quality images, lighting conditions, and shooting angles negatively impact the overall efficiency of many previously proposed methods [40]. The issue of data imbalance is a significant concern within the field of DL since it can have a substantial impact on the overall performance of the model. The presence of both overfitted and underfitted data complicates this problem. To overcome these challenges, this article proposes a robust multi-stage Ensemble Learning (EL) strategy designed to improve the diagnosis of wheat leaf diseases and achieve better generalization across diverse datasets.

This paper presents a multi-stage Ensemble Learning (EL) strategy, based on CNN, for the diagnosis of wheat leaf diseases. The approach utilizes pretrained CNN models, including InceptionResNetV2, DenseNet201, and a bespoke CNN model. The initial phases of the proposed model are delineated as follows: a) Dividing the selected training dataset into several subsets; b) Training three parallel CNNs using an interactive modeling approach to capture essential characteristics of healthy and diseased wheat leaves, thereby improving the accuracy of feature extraction; c) Utilizing EL for selecting ensemble members and transforming their reliability; and d) Implementing bagged EL to determine the best model for producing the final output.

The organization of the following sections in this paper is as follows: Section II involves a thorough analysis that specifically concentrates on the literature evaluation related to the identification of wheat leaf diseases. Section III primarily functions as an introductory section that outlines the methodology used in this study, offering a thorough review of the novel approaches adopted. Proceeding to Section IV, the experimental data are presented in full, together with a comprehensive analysis that seeks to clarify the significance of the findings. Section V functions as the concluding section, providing a concise summary of important observations and suggesting prospective avenues for future research in the domain of wheat leaf disease diagnosis.

II. LITERATURE REVIEW

Diseases pose a significant challenge to food security [41], thus requiring a notable focus on agricultural research associated with the detection of wheat diseases. Consequently, numerous researchers have adopted DL image recognition techniques within the agricultural domain to successfully identify wheat diseases from various perspectives. AlexNet model was employed for the purpose of classifying various types of wheat diseases, including stem rust, yellow rust, powdery mildew, and healthy. To facilitate the training process, a total of 7062 wheat photos were restricted for training, while an additional 1766 images were reserved for

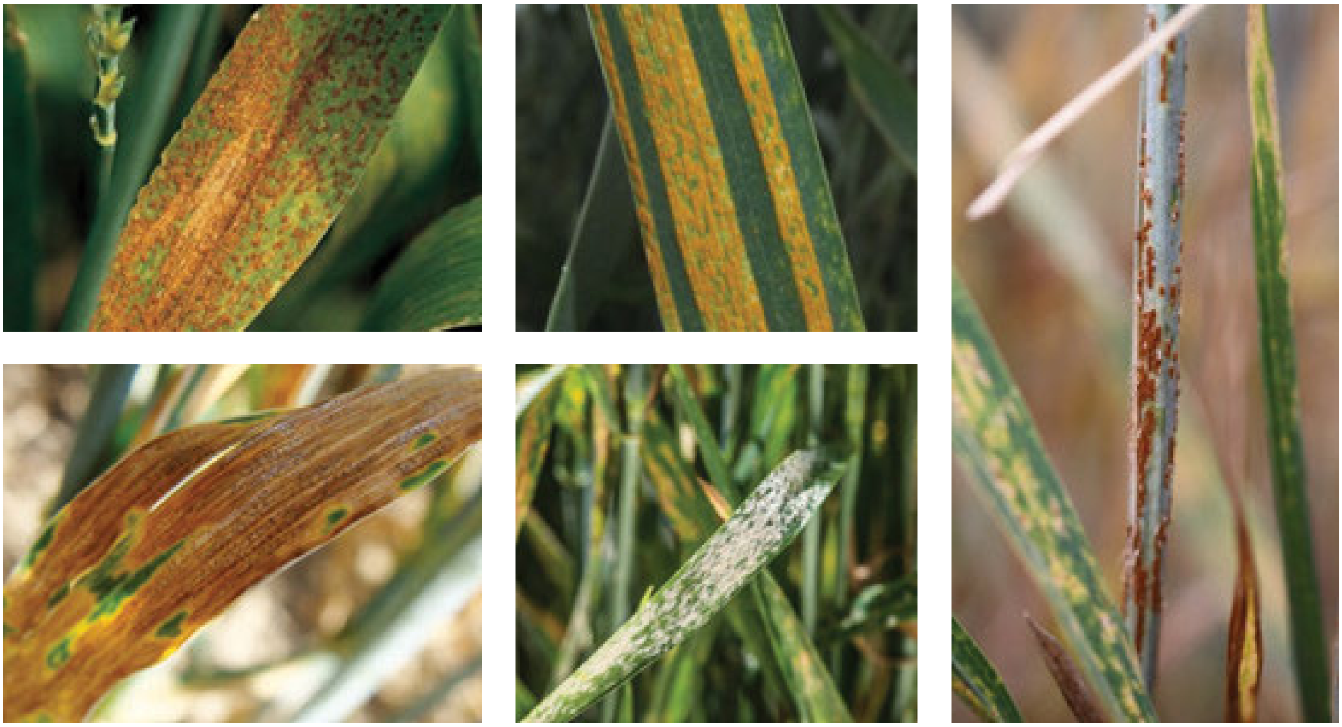


FIGURE 1. Digital images showing various wheat diseases.

testing. They reported a maximum accuracy of 84.54% when the batch size = 16 and the learning rate = 0.0001 with ReLU as an activation function [42].

A deep CNN-based approach with multiple Inception-Resnet layers for feature extraction was employed for wheat rust diseases using Unmanned Aerial Vehicle (UAV) images gained from four wheat fields in China. The results demonstrated model accuracy of 85.00% [43]. Manually created datasets, including leaf rust and tan spot images, were employed for training various models, including GoogLeNet, Support Vector Machines (SVM), and VGG16. The performance of each model varied across distinct subsets of the problem and achieved an average accuracy of 89.5% [44]. A novel approach for wheat disease identification was introduced through the utilization of a differential amplification CNN on a self-constructed dataset with eight kinds of leaf diseases. The average identification accuracy showed an increase of 6.03% with an accuracy of 95.15% when compared with LeNet-5 [45]. Wheat stripe rust was identified using UAV images and the PSPNet semantic segmentation model. To prove efficiency results were compared with RF, SVM, FCN, U-Net, and Backpropagation Neural Network (BPNN) algorithm and showed the highest accuracy of 94.00% [46]. Wheat leaf diseases were identified Elliptical-Maximum Margin Criterion metric learning and segmentation were performed with the Otsu technique. The average accuracy was 94.16% for identifying stripe rust and powdery mildew [47]. A lightweight multiscale CNN model with residual and inception modules and attention methods (CBAM and ECA) to improve feature extraction and reduce background interference. A high accuracy of 98.7% on the test dataset surpasses classic models like AlexNet and VGG16, as well as lightweight architectures like MobileNetV3. This method is ideal for mobile real-time wheat disease diagnosis [48]. SimpleNet is a lightweight CNN model for wheat ear disease detection using convolution and inverted residual blocks supplemented with CBAM and a feature fusion module to increase feature representation and reduce background interference. SimpleNet outperformed lightweight CNN models like VGG16, ResNet50, and MobileNet V3 with 94.1% accuracy on the test dataset [49].

DL techniques were investigated for the classification of wheat spike and leaf diseases with ten classes using the LWDCD2020 dataset. A comparative analysis was conducted with established models such as VGG16 and ResNet50, revealing that the suggested model exhibits superior performance, surpassing these models by 7.01% and 15.92% respectively [50]. EfficientNet framework was developed to classify wheat fungal diseases using the Wheat Fungi Disease dataset (WFD2020) consisting of 2414 images representing five distinct fungal diseases. The application of the EfficientNet architecture delivered a notable accuracy rate of 94.20% [39]. Multi-task learning was employed to enhance the VGG16 model in conjunction with the utilization of a pretrained model on ImageNET for transfer learning. This approach was developed to identify two distinct wheat leaf diseases on a self-collected dataset consisting of 40 photos for each disease [51]. The classification of wheat rust diseases was conducted through a deep CNN technique on a publicly accessible CGIAR dataset and achieved an accuracy of

97.16% with dropout rate = 50%, batch size = 64, learning momentum = 0.9, and ADAM optimizer [52]. The efficiency of different DL models was evaluated using open-source data from mundi.com for the identification of wheat diseases namely stripe rust, leaf rust, and stem rust. They reported that the VGG19 model achieved the highest accuracy [53].

EL was employed for the identification of three wheat rust classes, namely leaf rust, stem rust, and healthy, using snapshot ensembling, bagging, and stochastic gradient descent with warm restarts. The authors found that state-of-the-art CNNs like ResNet-101, ResNet 152, VGG, DenseNet-169, and DenseNet-201 improve wheat rust detection by 8% to 32% [54]. Comparative analysis was performed to evaluate the performance of the MobileNet model in relation to EfficientNet for detecting wheat leaf diseases stripe rust and Septoria with healthy class and performance was slightly better in the case of MobileNet with 97.00% accuracy [55]. A lightweight DL architecture that uses spatial channel attention and pruning to extend the model's receptive was designed to classify wheat rust diseases with multispectral images captured by UAV. The suggested Efficient Dual Flow UNet (DF-UNET) model minimizes computing cost while matching state-of-the-art accuracy [56]. ML-based framework with several steps, including data collection from different fields, preprocessing, and segmentation, was applied to train the model for the classification of stipe and brown rust diseases in wheat crops [57]. A methodology that investigates two CNNs for the purpose of feature extraction followed by feedback blocks employed to train the optimized features. The suggested methodology demonstrates a remarkable capability for classifying wheat diseases across several datasets including LWDCD 2020, CGIAR, and Plant Diseases [58]. CNN-based wheat leaf disease detection and classification approach was presented to improve accuracy and efficiency, and multiple image augmentation methods were utilized to eliminate overfitting and achieved 94.00% testing accuracy [59]. Wheat rust disease classification on data collected from several fields was performed by training two transfer learning models (Xception and ResNet-50) and subsequently compared in terms of prediction time, accuracy, and memory utilization. The superior performing model, based on ResNet50 architecture with an accuracy of 96.00%, was subsequently implemented on embedded devices to conduct field testing for evaluation [60]. A transfer and ensemble learning framework for diagnosing eight wheat leaf diseases was performed, achieving 98.08% accuracy, outperforming models like CNN, SimpleNet, and VGG16 [61].

Prior studies in wheat disease detection face limitations such as dataset-specific performance, overfitting due to limited data, and high computational demands, making deployment in resource-constrained environments challenging [54], [55], [59]. Additionally, the lack of cross-dataset validation and reliance on transfer learning restrict their generalizability to diverse diseases, environmental conditions, and data variations [58], [60]. Building on these limitations, our work proposes a multi-stage CNN-based Ensemble Learning (EL) approach that integrates pretrained CNNs with a bagging ensemble technique to enhance feature extraction and reduce overfitting. By addressing computational efficiency and ensuring robust cross-dataset performance, the proposed method achieves high accuracy on three publicly available datasets, outperforming state-of-the-art models while providing scalability for real-world applications.

III. PROPOSED METHODOLOGY

The objective of this research is to create a systematic structure for automatically categorizing diseases affecting wheat leaves. This will be accomplished by creating a combined model. The proposed ensemble model consists of two main components. The first component involves feature extraction, which includes preprocessing the dataset and extracting deep features. The second component incorporates a classifier that generates classifications of wheat leaf diseases within the proposed EL framework. The distinct stages of the Ensemble Learning (EL) framework referred to as ‘‘multi-stage’’ involve a systematic workflow for feature extraction, allowing the model to capture detailed low-level features alongside broader, high-level patterns. This ensures a more thorough and effective understanding of the data. The specific steps involved in implementing this framework will be explained in the following sections.

A. SELECTED CNN MODELS

DL models have gained significant popularity for image recognition and classification due to their increasing availability, versatility, and strong predictive capabilities. Therefore, the suggested method employs a DL approach for the detection and classification of wheat leaf diseases through image analysis. In this study, seven widely recognized pretrained CNN models, namely VGG19 (Vg) [62], MobileNetV2 (Mo) [63], ResNet50 (Re) [64], DenseNet201 (De) [65], InceptionResNetV2 (In) [66] and NASNetLarge (Na) [67], were applied to accurately detect leaf diseases.

1) VGG19

VGG19 architecture was introduced in 2014. VGG19 is an enhanced version of its preceding models [68]. It is a CNN consisting of 19 layers including 16 convolutional layers and 3 fully connected layers. The purpose is to classify images into 1000 distinct object categories. The VGG19 model has been trained using the ImageNet database, which consists of one million images categorized into one thousand distinct categories. To preserve the spatial resolution of the input image, each convolutional layer used a kernel size of 3 × 3 with a stride of 1 pixel. The initial 16 convolutional layers are considered for the purpose of extracting features, while the subsequent 3 layers are dedicated to the task of classification. These layers are divided into 5 groups, each of which is followed by a max-pooling layer. This model takes an input image with dimensions of 224 by 224 and returns the label of the object in the image. VGG19 employs a uniform structure of convolutional layers, offering interpretability and ease of implementation. VGG19 serves as a reliable benchmark model, demonstrating baseline performance in extracting spatial and texture features critical for disease detection.

2) RESNET50

ResNet50 was first time introduced in the year 2015. It is a member of the residual network family, which employs a technique known as residual mapping to avoid degradation problems. The degradation problem occurs when increasing the depth of a neural design leads to a decrease in the accuracy of the network [69]. The ResNet50 architecture consists of a total of 48 convolutional layers, along with one Max-Pool layer and one Average-Pool layer. These convolutional layers are organized into 5 blocks. Each of these blocks is comprised of a collection of residual blocks. The inclusion of residual blocks in the architecture facilitates the preservation of valuable information from preceding layers, hence enhancing the network's ability to acquire more effective representations of the input data. Residual networks have found extensive applications due to their easy architecture for optimization, which enables enhanced accuracy as the network depth increases.

3) DENSENET201

DenseNet201 is a 201-layer CNN architecture that has dense connections across layers. These connections are established using Dense Blocks, which connect all layers directly with each other. The DenseNet201 architecture incorporates the integration of supplementary inputs from prior layers at each layer. As output (feature maps) of all preceding layers are received by the convolutional layer as input and propagates its feature maps to all following levels. By utilizing the concatenation method, the neural network transmits its own feature-maps to subsequent higher layers. Consequently, every subsequent layer acquires the cumulative knowledge of all prior layers. This is mostly due to the design choice of connecting all preceding layers to each subsequent layer, resulting in compact and thinner architecture. DenseNet201 reuses features via dense connectivity, reducing redundancy and improving learning efficiency. Despite its depth (201 layers), it has a low parameter count (20.0M) and small size (77 MB).

4) INCEPTIONRESNETV2

InceptionResNetV2 model comprises a total of 164 layers. Its design includes global average pooling 2D, a dense layer, batch normalization, an activation function, a dropout layer, and another dense layer for image classification. Inception-ResNetV2, based on CNN architecture, incorporates residual connection. The InceptionResNetV2 architecture integrates the multi-level feature extraction capabilities of the Inception network with the residual connections found in the ResNet model, in order to improve gradient flow and facilitate network convergence [70]. This hybrid DL model integrates the advantages of a residual network with the distinctive characteristics of the Inception network's multi-convolutional core. The layer is provided with a 299 × 299 image as input and produces output possibilities. The utilization of residual connections in deep structures serves the purpose of reducing training time and mitigating the degradation issue [71], [72], [73].

5) MOBILENET

MobileNet architecture was specifically developed to help reduce the computational cost associated with CNNs. This was achieved by employing an inverted bottleneck approach in place of conventional 2D convolutional layers. In order to enhance computational efficiency, the MobileNetV2 architecture integrates three depth-wise separable convolutions, hence reducing the computational expenses of the network [74]. The size of the input image is 224 × 224 × 3. The images undergo a 2D convolutional layer and thereafter pass a sequence of bottleneck layers, resulting in dimensions of 7 × 7 × 320. Subsequently, the input undergoes a final 2D convolutional layer and an average pooling layer, resulting in dimensions of 1 × 1 × 1280. This vector is then forwarded to the last stage of the process. The implementation of SSD Light was aimed at cost reduction through the reduction of factors involved in the process. The fusion of SSD Light and MobileNetV2 yielded comparable accuracy while utilizing a reduced number of parameters.

6) NASNET

Neural Architecture Search network, commonly known as NASNet, was developed by the Google Brain Team. Its primary objective is to employ reinforcement learning techniques in order to identify the most optimal CNN architecture [75]. This model has been trained on a vast dataset consisting of millions of photos from the ImageNet database. The input image dimensions for this network are 331 × 331. The fundamental unit of this structure comprises two primary functionalities namely normal and reduction cells. Convolutional cells that yield a feature map of the same dimensions are commonly denoted as normal cells, but convolutional cells that yield a feature map with reduced width and height by a factor of two are commonly referred to as reduction cells. The controller Recurrent Neural Network (RNN) exclusively explores the architectures present in the Normal and Reduction Cells. The cells found in normal or reduction cells are composed of many blocks, with each block functioning as a CNN model. It is built by Neural Architecture Search (NAS), which automatically selects the optimum architectural configurations for performance. Despite its huge parameter count (88.9M), it has state-of-the-art accuracy for complex tasks.

TABLE 1. Characteristics of Pre-trained CNN models.

Model	Depth	Size (MB)	Input Size	Parameters (M)	Accuracy (%)	
					Top-1	**Top-5**
ResNet50 (Re)	50	96	224×224×3	25.6	74.9	92.1
InseptionResNetV2 (In)	164	209	299×299×3	55.9	80.3	95.3
DenseNet201 (De)	201	77	224×224×3	20.0	77.3	93.6
VGG19 (Vg)	19	535	224×224×3	143.6	71.3	90.0
NasNetLarge (Na)	-	332	331×331×3	88.9	82.5	96.0
MobileNetV2 (Mo)	53	13	299×299×3	3.5	71.3	90.1

B. SUMMARY OF SELECTED PRE-TRAINED MODELS

The selected pre-trained models exhibit varying structural characteristics due to differences in layer arrangement and nature. The choice of feature extraction layers and the specific features extracted per image vary across different models. 4096 features for the single image are extracted from the fc7 layer in the VGG19 model. For MobileNetV2 and NASNetLarge models 1280 and 4032 features are extracted by global_average_pooling2d_1 and global_average_pooling2d_2 layers, respectively. 2048, 1920 and 1536 features are extracted through avg_pool layer for ResNet50, DenseNet201 and InceptionResNetV2, respectively. CNN models with size, depth, input size, parameters, and accuracy are given in Table 1.

C. NEURAL NETWORK ENSEMBLE APPROACH

In the context of classification, a designation of a six-stage EL model, emphasizing its reliance on reliability-based neural networks is presented in this section. The neural network ensemble's fundamental concept emerged from the utilization of all the useful data concealed in network neural classifiers, each of which can enhance generalization. In the presented multistage network neural ensemble model, the bagging sampling technique is utilized for the generation of distinct training sets to ensure abundant data for training. Subsequently, distinct neural classifiers are individually trained utilizing these varied training datasets. Consequently, the classification outcomes and reliability values for each neural network classifier are determined. Afterward, to refine the ensemble, a maximize decorrelation technique is utilized for the selection of relevant members from several trained classifiers of the neural network. After this, the conversion technique is applied to reliability values, which helps to restrict them in unit intervals to prevent a scenario where a classifier member classifier having a high absolute value frequently controls the ensemble's final decision. Lastly, the ensemble members are consolidated according to specific criteria, output is based on reliability measures. The resultant outcome is referred to as the ensemble's output. The systematic representation of the multistage neural reliability network ensemble's learning model is given in Fig. 2.

D. ORIGINAL DATASET PARTITIONING

Given the lack of data set on certain problems of data analysis techniques that involves bagging are utilized. In bagging the samples are created by varying the subsets data selection or perturbating training sets. Bagging was introduced in 1996 and is now widely used in data sampling for the field of ML. The technique of bagging data sampling is explained in Algorithm 1, in the given technique the original data sampling set of size P is represented by DS, where N represents the set of training data that contain n elements. The bagging technique efficiency is based on the ability to construct a reasonably large training set, attributed to its

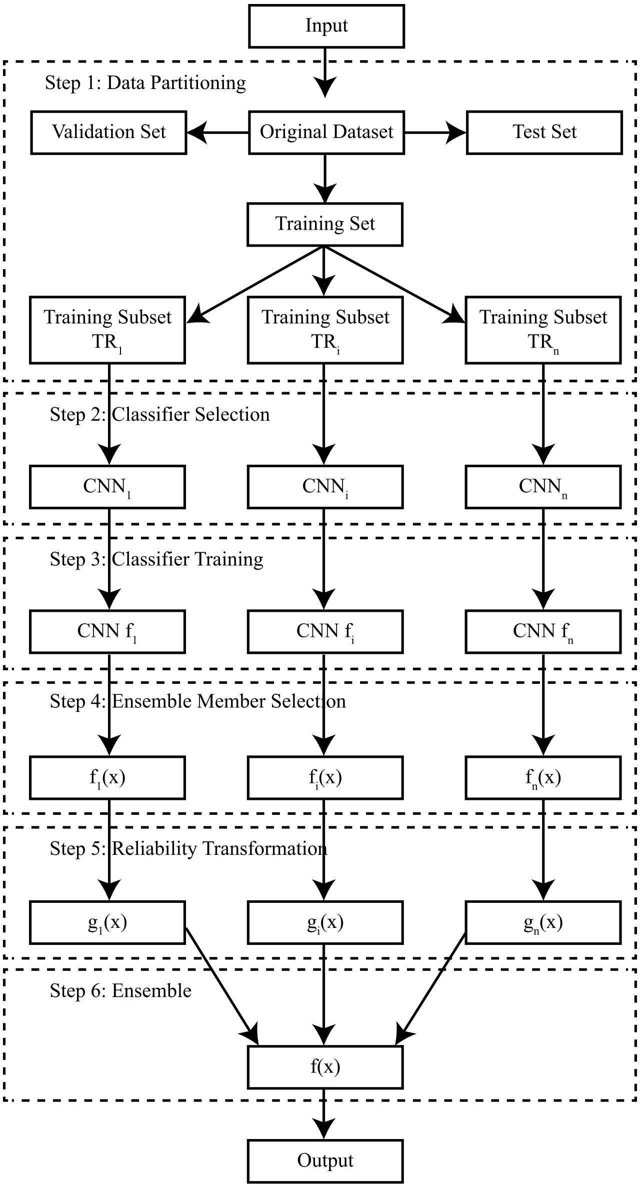


FIGURE 2. Multi-stage neural network EL approach.

use of random sampling with replacement. Consequently, bagging is a suitable technique for machine-learning. Within this study, the bagging technique is utilized to generate varied training data subsets through bootstrap sampling, particularly in situations of data scarcity. Each subset was generated by random sampling with replacement from the original dataset, ensuring that each subset has the same size as the original dataset but may contain duplicate samples. This randomness introduces diversity among the models trained on these subsets.

```
Algorithm 1 Technique of Bagging Data Sampling
Input: Original Dataset $DS$, Partitioning Factor $P = 0.7$
Output: The generated new training subsets ( )
for $t = 1$ to $n$
  for $i = 1$ to $N$
    $RandRow = P \times rand()$
    if $RandRow \leq P$
      $P_t(i, AllCols = DS(RandRow, AllCols)$
    endif
  Next $i$
Next $t$
Output The final training subsets $(R_1, TR_2, \ldots, TR_n)$
```

E. SELECTING DISTINCT NEURAL NETWORK MEMBERS

A sufficient and necessary condition for an ensemble classifier could be more precise than its individual members if classifiers are diverse and accurate. An efficient ensemble classifier made up of various models with a lot of disagreement is generally more likely to perform well in terms of generalization when it comes to the bias-variance concept of trade-off. Thus, a key consideration is how to create a diversified model. In demand of neural network techniques, numerous techniques have been explored to produce ensemble members with various errors. Fundamentally, these methodologies depend on altering parameters associated with neural network designs and their training. Notably, the primary methods cover four key aspects:

1) VARIED INITIAL CONDITIONS

Ensemble members exhibit diversity by modifying initial conditions, encompassing factors that include learning rates, random weights, and momentum rates during training.

2) ARCHITECTURAL VARIABILITY

To create distinct neural networks having varied architectures is achieved by modifying the number of nodes and hidden layers in each layer.

3) DIVERSIFICATION OF TRAINING DATA

Attaining diverse training sets and, consequently, distinct network generations can be accomplished by data resampling and preprocessing, which has six methodologies to attain diverse sets of training data: bagging, noise injection cross-validation, stacking, and boosting input decimation.

4) DIVERSE TRAINING METHODOLOGY

Opting for diverse core learning techniques facilitates the generation of different ensemble members. For instance, when dealing with a multilayered feedforward network, alternatives include the steep-descent technique and a range of other training techniques.

Choosing the third technique in the proposed study is based on the diversity achieved in the prior phase, which results in a multitude of distinct training data subsets. Moreover, a three-layer BPNN was introduced [76]. This choice is informed by the fact that a three-layer BPNN, incorporating a transfer identity function in the output unit and transfer logistic functions in middle-layer units, can proficiently estimate any function that is continuous with an adequate number of middle-layer units. The use of diverse training datasets will result in the generation of diverse neural network classifiers.

F. CONFIDENCE VALUE AND NEURAL NETWORK LEARNING PROCESS

After the generation of the neural networks classifier, in the next phase, a variety of datasets are utilized in training the neural networks. In our work, the adopted BPNN is a subclass of backpropagation supervised error learning technique that adopts the neural-network associative memory structure. Generally, the backpropagation learning technique comprises two steps: one is known as the backpropagation phase and the other one is a forward-propagation phase. Consider a data set that comprises s samples, each of which is represented by input vector set $W_i = (w_i1, w_i2, w_i3, \ldots, w_im)$ and output target vector $U_i = (u_i1, u_i2, u_i3, \ldots, u_in)$, where $1 \leq i \leq s$.

In the forward-propagation phase, the input layer is fed by input vector W_i which results in output vector $V_i = (v_i1, v_i2, v_i3, \ldots, v_in)$ determined by present weight vector X. The main aim is to reduce the error function K explicitly defined as:

$$K = \sum_{i=1}^{s} \sum_{j=1}^{n} \frac{(v_{ij} - u_{ij})^2}{2} \tag{1}$$

The achievement of accurate input vector mappings to corresponding output vectors involves the modification of the weight vector X. In the back-propagation phase, a gradient decent in weight space X is executed to identify ideal solutions. The change Δx_{ij} in both magnitude and direction is computed as:

$$\Delta x_{ij} = -\frac{\partial K}{\partial x_{ij}} \epsilon \tag{2}$$

where $\epsilon \in (0, 1)$, represents a learning parameter that controls the convergence rate of the algorithm. In the second phase, the squared error calculated by the formulation given in (2), is retroactively propagated by output units to input units. This propagation path at each level is determined by weight changes. In every iteration of the backpropagation technique, the execution of both phases involves until the error-function K converges.

In the classification task domain, the neural networks are commonly trained by an in-sample set of data, while the verification is done by an out-sample set of data. The model parameters iterative refinement that includes node biases and connection weights is carried out for the minimization of error-function K. Essentially, the final output model of the Feed-Forward Neural Network (FFNN) is given as:

$$v = g(w) = \alpha_0 + \sum_{j=1}^{q}\left(x_j\varphi\left(\alpha_j + \sum_{i=1}^{p} x_{ij}w_i\right)\right) \tag{3}$$

Here, α_j, $j = 0, 1, \ldots, q$ denote the bias of $j - th$ unit, whereas $x_ij(i = 1, 2, 3, \ldots, p, j = 1, 2, 3, \ldots, q)$ is the representation of connection weights between the model layers. The layer hidden transfer function is represented by $g(\bullet)$, input nodes number is signified by p and hidden nodes number is signified by q. Through neural network training, model parameters given in (3), are determined. As a result, the neural network classifier is represented as:

$$G(w) = \text{sign}\left(\alpha_0 + \sum_{j=1}^{q}\left(x_j\varphi\left(\alpha_j + \sum_{i=1}^{p} x_{ij}w_i\right)\right)\right) \tag{4}$$

In the proposed work, the prime focus is on utilizing output value $g(w)$ of neural networks as classification measurement score level rather than directly employing the classification results $G(w)$. For the classification of wheat diseases, parameter α_0 is used to influence the true positive percentage. Furthermore, the output neural network value $g(w)$ served as an effective reliability level indicator in ensemble classifiers. The increase in $g(w)$ results in the rise of predicting confidence of neural network classifiers. As a result, the use of $g(w)$ as a reliability measure facilitates the ensemble members' integration. By this technique, at the measurement level, decision fusion has been enabled.

G. SELECTION OF SUITABLE ENSEMBLE MEMBERS

After training is completed, each individual neural classifier generates its specific outcome. However, dealing with a substantial number of individual members requires selectively picking a representative subset to enhance ensemble efficiency. Notably, the ensemble model does not adhere to the "the more, the better" principle. Moreover, this emphasizes the diversity needed in neural network classifiers for ensemble effective learning. In the proposed technique, the maximization decorrelation technique is employed to precisely determine the appropriate number of ensemble members. As previously noted, the root of the maximization decorrelation method lies in the fundamental principle of promoting variety within an ensemble model. The maximization decorrelation helps to minimize the correlation between the chosen classifiers. If there exists p neural classifiers i.e., $g_1, g_2, \ldots, g_p$ that generates n forecast values, the p predictors error matrix i.e., $k_1, k_2, \ldots, k_p$ is defined as:

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1p} \\ k_{21} & k_{22} & \cdots & k_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{np} \end{bmatrix}_{n\times p} \tag{5}$$

Utilizing the above matrix information, the mean, covariance, and variance of K are evaluated as:

$$Mean : \bar{k}_i = \frac{1}{n}\sum_{l=1}^{n} k_{li} \quad (i = 1, 2, 3, \ldots, p) \tag{6}$$

$$Variance : \nu_{ii} = \frac{1}{n}\sum_{l=1}^{n}(k_{li} - \bar{k}_i)^2 \quad (i = 1, 2, 3, \ldots, p) \tag{7}$$

$$Covariance : \nu_{ij} = \frac{1}{n}\sum_{l=1}^{n}(k_{li} - \bar{k}_i)(k_{lj} - \bar{k}_j) \quad (i, j = 1, 2, 3, \ldots, p) \tag{8}$$

From (7) & (8), a variance-covariance matrix is obtained by:

$$\nu_{(p\times p)} = (\nu_{ij}) \tag{9}$$

Using the covariance-variance and correlation matrix R that are calculated by:

$$R = (r_{ij}) \tag{10}$$

$$r_{ij} = \frac{\nu_{ij}}{\sqrt{\nu_{ii}\nu_{jj}}} \tag{11}$$

Here r_{ij} represents the correlation coefficients indicating the correlation degree between classifiers g_i and g_j. Then, plural coefficient correlation, $\rho_{g(i|(g_1,g_2,\ldots,g_{i-1},g_{i+1},\ldots,g_p))}$ between g_i and $p - 1$ classifiers that could be computed by (10) & (11). For ease, abbreviate $\rho_{g(i|(g_1,g_2,\ldots,g_{i-1},g_{i+1},\ldots,g_p))}$ by ρ_i that represents correlation degree between g_i and $(g_1, g_2, \ldots, g_{i-1}, g_{i+1}, \ldots, g_p)$. The correlation plural coefficient is calculated by matrix correlation R such that:

$$R \xrightarrow{\perp} \begin{bmatrix} R_{(-i)} & r_i \\ r_i^T & 1 \end{bmatrix} \tag{12}$$

where, R_{-i} is the representation of correlation deleted matrix, and $r_{ii} = 1, i = (1, 2, \ldots, p)$ then plural coefficient correlation could be computed by:

$$\rho_i^2 = r_i^T R_{(-i)}^T r_i \tag{13}$$

For θ threshold, which is pre-specified, if $\rho_i^2 > \theta$, so ρ_i classifier could be eliminated from ρ classifiers. Suppose that the value of the ρ_i classifier could be preserved. Commonly, the maximization decorrelation technique could be summarized in the following stages:

(a) Computing the matrices of covariance-variance ν_{ij} and matrix correlation R utilizing (9) & (10).

(b) The coefficients of plural correlation ρ_i could be calculated from (13), for $i^t h$-classifier where $i = 1, 2, 3, \ldots, \rho$.

(c) For threshold prespecified θ, if $\rho_i < \theta$, then i^th-classifier would be removed from ρ classifiers. Similarly, if the value of $\rho_i > \theta$, then the value for i^th-classifier would be retained.

To retain the classifiers, the whole process of (1), (2) & (3) is iteratively performed to get better results.

H. VALUE RELIABILITY TRANSFORMATION

Utilizing the neural classifier outcomes in the preceding phase acts as a measure of reliability. It's important to emphasize that the value of reliability is confined to the range $(-\infty, \infty)$. Nevertheless, a significant limitation of this confidence measure is that the ensemble classifiers having substantial absolute values frequently utilize significant impact on the ultimate outcome of the ensemble model. To address this drawback a basic strategy is applied that involves the transformation of output values to achieve mean value zero and standard deviation i.e.:

$$z_i^+(w) = \frac{g_i(w) - \mu}{\sigma} \tag{14}$$

Here, σ and μ are the representation of Standard Deviation (SD) and mean of classifier pooled output. If classifiers dissimilarity measures output, reversing the sign of the given output should precede this transformation. To simplify the given process, the transformation of confidence values in the interval having the range between 0 and 1 gives better results. To represent the probability measure in neural outputs, logistic functions play a vital role in the context of neural networks. Hence, it is easily utilized as a scale function for reliability measuring transformation as given below:

$$z_i^+(w) = \frac{1}{1 + e^{-g_i(w)}} \tag{15}$$

In the classification of binary techniques, if a class of positive reliability has degree representation as $z_i^+(w)$, then a class of negative reliability has degree representation as:

$$z_i^-(w) = 1 - z_i^+(w) \tag{16}$$

The results of reliability transformed values depicted that numerous classifiers can be incorporated into ensemble neural classifier outputs as demonstrated in the succeeding section.

I. INTEGRATION OF NUMEROUS CLASSIFIERS INTO ENSEMBLE OUTPUTS

Based on the preceding stages' work, suitable ensemble member number sets could be accumulated. The main task is the combination of elected members set with classifiers aggregated in a suitable strategy ensemble. Commonly, in literature, the strategies of the ensemble are of rank and abstract levels. The main three ensembles' techniques are average weighting, rank, and majority vote. To classify problems, the ensemble approach used for this purpose is Majority vote because it is easily implemented. The voting technique of ensemble members is utilized to get the outcome. Mostly, half ensembles are required for accepted results agreement as ensemble outcome instead of every generalized network's accuracy and diversity. The majority voting drawback is that sometimes its disregard for neural networks possibility that lies in the minority might, at times, deliver correct results. At the stage of ensemble, the ensemble's main motivation, like diversity existence, has been ignored. Also, at an abstract level, the ensemble strategy-only class is a majority vote.

Low-level classifiers of ensemble members utilized the rank technique and it helped in producing not just a single output but also given choices that are ranked according to their probabilities. Hence, the high-level class of classifiers is chosen from those sets of classes that have additional information that is not given in specific low-level classifiers. As mentioned earlier, at rank levels, the fusion strategy class is rank strategy. In the ensemble's final decision, the weighted average is determined by the ensemble's individual members and every output member has a weight value. Every ensemble member is assigned a portion of the total weight, which is fixed at one, depending on their respective performances or diversity. Despite being a form of ensemble strategy at the measurement level, acquiring optimal weights for each ensemble member is a complex task in the context of classification challenges. In these challenges, here is a strategy based on ensemble reliability to get an output of the ensemble at the measurement stage. Here five strategies are presented that could be used in the integration of ensemble individual members:

$$G(w) = \begin{cases} 1 & \text{if } \max(z_i^+(w)) \geq \max(z_i^-(w)) \\ -1 & \text{otherwise} \end{cases} \tag{17}$$

$$G(w) = \begin{cases} 1 & \text{if } \min(z_i^+(w)) \geq \min(z_i^-(w)) \\ -1 & \text{otherwise} \end{cases} \tag{18}$$

$$G(w) = \begin{cases} 1 & \text{if } median(z_i^+(w)) \geq median(z_i^-(w)) \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

$$G(w) = \begin{cases} 1 & \text{if } \sum_{i=1,2,\ldots,m} z_i^+(w) \geq \sum_{i=1,2,\ldots,m} z_i^-(w) \\ -1 & \text{otherwise} \end{cases} \tag{20}$$

$$G(w) = \begin{cases} 1 & \text{if } \prod_{i=1,2,\ldots,m} z_i^+(w) \geq \prod_{i=1,2,\ldots,m} z_i^-(w) \\ -1 & \text{otherwise} \end{cases} \tag{21}$$

The summarized conclusion of the neural networks multistage ensemble reliability learning model is given as:

(a) Original set of data is portioned into training n-datasets as: $TR_1, TR_2, \ldots, TR_n$

(b) Train individual n models of neural networks via distinct training sets of data, i.e., $TR_1, TR_2, \ldots, TR_n$, then by this $n - distinct$ neural classifiers (ensemble member) are obtained.

(c) Select decorrelated m neural network classifiers from n classifiers (neural) obtained in step 2, by utilizing the technique of maximization decorrelation.
(d) Utilizing (3), for selection of m output results of neural classifiers for new sample (unlabeled) w, $g_1(w), g_2(w), \ldots, g_m(w)$, as given in Fig. 1.
(e) Transform the output of degree reliability utilizing (15) & (16) to obtain positive and negatives classes represented as: $z_1{}^{+}(w), z_2{}^{+}(w), \ldots, z_m{}^{+}(w)$ and $z_1{}^{-}(w), z_2{}^{-}(w), \ldots, z_m{}^{-}(w)$.
(f) Use neural multiple classifiers to generate output aggregated in reliability term values utilizing (17) - (21).

IV. EXPERIMENTAL RESULTS

A series of experiments conducted under various parameters effectively verify the proposed model's performance. This section presents experimental results on various selected datasets and compares them to existing methodologies.

A. DATASETS AND EXPERIMENTAL SETUP

Several datasets concerning wheat leaf diseases are made accessible to the public, to facilitate the development of algorithms capable of effectively classifying these diseases. The proposed technique is validated on three different wheat disease datasets. A publicly available Wheat Disease Images Small Dataset (D1) [77] a subset of the full dataset is used containing 999 images. These images are of various major diseases including "Yellow Rust", "Brown Rust", "Septoria", "Mildew" and "Healthy". The images were obtained in a realistic and uncontrolled growth condition from various locations across the UK and Ireland. Images were taken using a smartphone and digital camera; therefore, varied sizes of images are present in the dataset.

Kaggle dataset namely Wheat Leaf dataset (D2)[1] is employed. The wheat leaf dataset is composed of 407 images which are divided into three categories with the assistance of plant pathologists, namely "Stripe Rust (Yellow Rust)," "Septoria," and "Healthy" wheat leaf. The data was collected in an uncontrolled environment in a real wheat farm in Holeta, Ethiopia. Canon EOS 5D Mark III a high-resolution digital camera was used for capturing the images having the capability to effectively depict the intricate details of the leaf. The Wheat Disease dataset (D3)[2] used in this study comprises 2942 images of wheat disease that were retrieved from the Kaggle repository. The wheat disease dataset is divided into three classes, "Brown Rust", "Yellow Rust" and "Healthy".

Each experiment is conducted on NVIDIA GeForce RTX 3080Ti with a Boost Clock of 1.77GHz and standard memory of 12 GB GDDR6X. Python 3.11.4 and TensorFlow v2.14.0 are used to train, test, and validate the proposed technique. The selected pretrained models undergo transfer learning, starting with an initial learning rate of 0.0001. The entire process consists of 100 epochs and has a total momentum of 0.45. The datasets are divided into training, testing, and validation sets using several ratios, which are discussed in detail in section IV-C. The performance is assessed using seven evaluation metrics such as Accuracy (Acc), Precision (Pre), Sensitivity (Sen), Specificity (Spe), F1-score (F1), Training Time (TT), and Prediction Time (PT). The experimental results are repeated at least five times under identical conditions to compute the average results, which are provided in the next section.

B. CLASSIFICATION RESULTS

This section shows the results of classification after training on 70% of the dataset. The efficiency of the proposed EL model on the D1 dataset is validated through a comparison of the results with pretrained models. Of the six pretrained models, De model has performed the best with an average accuracy of 77.00%, precision of 81.00%, and sensitivity of 73.00%. However, this model takes a high training time of 205 minutes and an average of 2.10 seconds to predict an input image, which limits its effectiveness. Followed by De, Mo achieved second highest accuracy of 74.00%, precision of 73.00%, and sensitivity of 71.00% with TT of 84 minutes and PT of 0.81 seconds. Re attained the lowest accuracy of 34.00% among all pretrained models, while Na had the highest TT and PT of 928 minutes and 10.02 seconds, respectively. However, the custom CNN model obtained slightly improved results with an average accuracy of 78.47%, precision of 82.91%, and sensitivity of 76.85%. The custom CNN classifier takes 100 minutes to train and 3.89 seconds to predict. Bagged EL is performed on three best-performing models including custom CNN, De, and Mo. This model has achieved an accuracy of 79.02%, precision of 84.46%, and sensitivity of 78.12% which is higher than single De, Mo, and custom CNN. For the D1 dataset, the experimental results of all the models are shown in Table 2.

To verify the efficiency of the proposed model, a comparison of the results is performed with selected models on D2 dataset. Among the six pretrained models, the De model has superior performance, with an average accuracy of 93.29%, precision of 93.35%, and sensitivity of 93.35%. However, this model has a high training time of 183 minutes and an average prediction time of 1.81 seconds per input image. Mo has achieved the second-greatest accuracy of 92.94%. It has a precision of 93.02% and a sensitivity of 93.02%. The TT required is 78 minutes, and the PT is 0.73 seconds. Re attained the lowest level of accuracy, measuring 77.65%, compared to all other pretrained models. On the other hand, Na demonstrated the greatest TT of 833 minutes and PT of 8.45 seconds. The custom CNN model yields optimal outcomes among all pretrained models, with an average accuracy of 94.27%, precision of 94.89%, and sensitivity of 93.99%. The customized CNN classifier requires 89 minutes for training and 0.86 seconds for prediction. While proposed EL model outperformed all other models when applied to three top-performing models. It achieved an accuracy of

[1] https://www.kaggle.com/datasets/olyadgetch/wheat-leaf-dataset/data
[2] https://www.kaggle.com/datasets/buffyhridoy/wheat-disease

TABLE 2. Results of CNN models on D1 dataset.

Models	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
ResNet50 (Re)	34.00	23.00	13.00	95.50	16.33	271 ± 23	2.98 ± 0.97
InseptionResNetV2 (In)	65.00	69.00	60.00	93.75	63.00	167 ± 17	1.73 ± 0.81
DenseNet201 (De)	77.00	81.00	73.00	95.75	75.67	205 ± 14	2.10 ± 0.86
VGG19 (Vg)	67.39	51.45	31.88	94.37	38.41	257 ± 22	2.50 ± 0.92
NasNetLarge (Na)	68.63	67.65	60.78	92.89	63.07	928 ± 41	10.02 ± 1.61
MobileNetV2 (Mo)	74.00	73.00	71.00	94.00	71.67	84 ± 11	0.81 ± 0.17
Custom CNN	78.47	82.91	76.85	95.22	78.77	100 ± 27	3.89 ± 1.12
EL **(Proposed)**	**79.02**	**84.46**	**78.12**	**96.42**	**79.87**	102 ±15	0.83 ± 0.13

TABLE 3. Results of CNN models on D2 dataset.

Models	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
ResNet50 (Re)	77.65	79.07	74.42	91.28	75.97	262 ± 21	0.62 ± 0.24
InseptionResNetV2 (In)	90.59	90.70	90.70	95.93	90.70	135 ± 17	0.43 ± 0.16
DenseNet201 (De)	93.29	93.35	93.35	95.67	93.35	183 ± 28	0.81 ± 0.33
VGG19 (Vg)	89.41	88.37	87.21	94.77	87.60	216 ± 22	0.42 ± 0.25
NasNetLarge (Na)	91.76	91.86	90.70	95.93	91.09	833 ± 32	0.45 ± 0.31
MobileNetV2 (Mo)	92.94	93.02	93.02	96.51	93.02	**78 ± 8**	0.72 ± 0.1
Custom CNN	94.27	94.89	93.99	95.03	94.32	89 ± 41	0.86 ± 0.13
EL **(Proposed)**	**96.48**	**96.89**	**96.21**	**96.98**	**96.08**	99 ± 16	**0.33 ± 0.5**

96.48%, precision of 96.89%, and sensitivity of 96.21% in TT of 172 minutes and 0.33 seconds to predict. The experimental findings of all models for the D2 dataset are presented in Table 3.

The proposed model's performance on the D3 dataset is validated through the comparison of the results against pre-trained models. Again, the De model outperformed the other six pretrained models, with an average accuracy of 96.57%, precision of 96.89%, and sensitivity of 95.62%. However, this model requires a prolonged training time of 271 minutes and takes an average of 2.69 seconds to predict an input image, reducing its relevance. Custom CNN has the second most significant accuracy with 95.44%, precision of 95.78%, and sensitivity of 95.18% with TT of 104 minutes and PT of 1.05 seconds. Among all models, Re has the lowest accuracy of 79.28%, while Mo has the lowest TT and PT of 88 minutes and 0.82 seconds, respectively. Bagged EL has been performed with three models including custom CNN. This model achieved an accuracy of 97.16%, precision of 97.49%, and sensitivity of 96.86% which is higher than all other pretrained as well as custom CNN models. EL model takes a significant TT of 112 minutes and 1.05 seconds to predict. Since all the chosen classifiers were considered and tested in this experiment, the EL approach attained the best accuracy, thus subsequent experiments are performed on the proposed EL model. Table 4 displays the experimental findings of all models for the D3 dataset.

C. DISCUSSION

1) ABLATION ANALYSIS

Once a model with the best performance is selected in the first experiment, that is, the EL model, this model is trained on various learning rates. On the D1 dataset, the proposed EL method achieved an average accuracy of 83.34%, precision of 85.70%, and sensitivity of 81.45% with an initial learning rate of 0.000001. The second-best average accuracy of 80.96%, precision of 82.67%, and sensitivity of 81.02% was achieved with an initial learning rate of 0.00001. The lowest average TT was 95 minutes, and the lowest average PT of 0.94 seconds was taken by a learning rate of 0.01, but it could only achieve 76.78% accuracy. It is noteworthy that the proposed model achieved the highest accuracy with a 0.00001 learning rate. Table 5 demonstrates all experimental results with different initial learning rates on the D1 dataset.

After selecting the best-performing technique from the first experiment and the best initial learning rate from the second experiment. As mentioned earlier, the proposed EL approach has performed better on D1 when 0.000001 is selected as the learning rate. During this experiment, selected evaluation matrices are used to note the performance for each training-testing-validation data split. A data split ratio of 70-15-15 has achieved the best average accuracy of 86.78%, best precision of 88.39%, and best sensitivity of 84.52%. This classifier requires 108 minutes for training and 0.51 seconds for predicting an input image. The second-best average accuracy of 83.34% is obtained with an 80-10-10 ratio of data split, which took 114 minutes to train and 0.68 seconds to predict. The lowest accuracy of 65.82% is noted for 50-25-25 at TT 97 minutes. Table 6 represents the impact of different data split ratios on the D1 dataset.

The same experiments were performed on the D2 dataset on the proposed model. With an initial learning rate of 0.000001, the EL model achieved an average accuracy of 97.32%, precision of 97.94%, and sensitivity of 97.12%. Followed by a learning rate of 0.00001, accuracy of 96.98%, precision of 97.91%, and sensitivity of 97.99% were achieved. The learning rate of 0.01 achieved the lowest average TT of 91 minutes and the lowest average PT of 0.94 seconds, however it only achieved 93.98% accuracy. Table 7 displays all experimental outcomes with various initial learning rates on the D2 dataset.

The most effective combination from the previous experiments is selected to be trained on various data split ratios to achieve the best outcomes. In this experiment, certain

TABLE 4. Results of CNN models on D3 dataset.

Models	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
ResNet50 (Re)	79.28	80.74	76.48	93.39	77.12	328 ± 34	3.21 ± 0.44
InseptionResNetV2 (In)	94.32	96.89	91.63	97.47	93.54	232 ± 29	2.46 ± 0.31
DenseNet201 (De)	96.57	96.89	95.62	98.12	97.69	271 ± 26	2.69 ± 0.37
VGG19 (Vg)	90.98	88.78	88.78	92.84	89.97	357 ± 36	3.68 ± 0.41
NasNetLarge (Na)	94.42	95.17	93.29	97.54	94.13	1140 ± 47	12.01 ± 1.23
MobileNetV2 (Mo)	93.97	94.28	93.40	93.35	93.61	88 ± 9	0.82 ± 0.1
Custom CNN	95.44	95.78	95.18	95.99	95.23	104 ± 38	1.13 ± 0.38
EL **(Proposed)**	**97.16**	**97.49**	**96.86**	**97.52**	**96.74**	112 ± 14	1.05 ± 0.19

TABLE 5. Results of different initial learning rates on D1 dataset.

Learning Rates	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
0.01	76.78	77.27	74.53	90.00	75.21	**95 ± 16**	0.94 ± 0.19
0.001	78.69	78.85	78.00	94.37	76.67	98 ± 12	0.87 ± 0.11
0.0001	79.02	84.46	79.12	96.42	79.87	102 ± 15	0.83 ± 0.13
0.00001	80.96	82.67	81.02	97.89	82.78	109 ± 14	0.73 ± 0.13
0.000001	**83.34**	**85.70**	**81.45**	**98.63**	**84.28**	114 ± 17	**0.68 ± 0.12**

TABLE 6. Results of training-testing-validation data split on D1 dataset.

Data Split	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
50-25-25	65.82	69.34	60.39	83.00	63.00	**97 ± 18**	0.83 ± 0.11
60-20-20	74.42	73.65	71.00	85.89	71.37	99 ± 16	0.71 ± 0.21
70-15-15	**86.78**	**88.39**	**84.52**	**98.95**	**85.24**	108 ± 19	**0.51 ± 0.06**
80-10-10	83.34	85.70	81.45	98.63	84.28	114 ± 17	0.68 ± 0.12

TABLE 7. Results of different initial learning rates on D2 dataset.

Learning Rate	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
0.01	93.98	95.17	95.39	95.54	94.82	91 ± 14	0.94 ± 0.15
0.001	95.38	95.48	95.48	97.63	95.25	94 ± 13	1.03 ± 0.11
0.0001	96.48	96.89	96.21	96.98	96.08	99 ± 16	0.33 ± 0.5
0.00001	96.98	97.91	98.07	97.99	97.14	102 ± 19	1.01 ± 0.21
0.000001	**97.32**	**97.94**	**97.12**	**97.76**	**97.04**	112 ± 23	1.13 ± 0.22

evaluation matrices are employed to record the performance of each training-testing-validation data split. The data split ratio of 70-15-15 delivered the highest average accuracy of 98.28%, the highest precision of 99.53%, and the highest sensitivity of 98.68%. The training process takes 109 minutes, whereas the prediction of an input image only requires 1.11 seconds. The model achieved the second highest accuracy of 97.32%, precision of 97.94%, and sensitivity of 97.12% with a data split ratio of 80-10-10. The accuracy of 83.65% is observed to be the lowest for a 50-25-25 ratio at TT of 87 min. Table 8 illustrates the effects of various data split ratios on the D2 dataset.

EL model with the highest performance in the initial experiment undergoes training using different learning rates on D3 datasets. The classifier achieved the highest average accuracy of 98.76%, precision of 98.43%, and sensitivity of 98.15%, using an initial learning rate of 0.000001 in TT of 125 minutes and PT of 1.00 seconds. An average accuracy of 98.02%, precision of 99.11%, and sensitivity of 98.32% were attained with an initial learning rate of 0.00001. The learning rate of 0.01 achieved the lowest average PT of 0.99 seconds and the lowest average TT of 101 minutes. However, despite these low times, it only achieved an accuracy of 95.12%. The proposed model demonstrated exceptional performance, achieving the greatest accuracy when trained with a learning rate of 0.00001. Table 9 presents a comprehensive overview of the experimental outcomes obtained using various initial learning rates on the D3 dataset.

As already mentioned, the proposed EL model exhibited superior performance on D3 when the learning rate was set to 0.000001. For training purposes, the optimal combination is chosen across various data division ratios to achieve the most favorable outcomes. The optimal data division ratio of 70-15-15 produced the highest sensitivity of 98.86%, precision of 99.49%, and average accuracy of 99.16%. Training this data takes 131 minutes, while prediction takes 0.94 seconds. An average accuracy of 98.76% for the 80-10-10 data split ratio model required 125 minutes to train and 1.00 seconds to predict. The condition with the lowest accuracy, 82.63%, is for 50-25-25 at TT 104 minutes and PT of 1.01 seconds. The results of varying data split ratios are illustrated in Table 10.

2) CRITICAL ANALYSIS

The models displayed in Tables 2 to 10 illustrate the comparative performance of various designs, their efficiencies, and the characteristics of the specified datasets. Models such as DenseNet201, NasNetLarge, and InceptionResNetV2 demonstrated superior accuracy and enhanced F1 scores due to their ability to extract more complex features. The densely interconnected layers of DenseNet201 enhanced feature reuse and generalization, whereas NasNetLarge benefited

TABLE 8. Results of training-testing-validation data split on D2 dataset.

Data Split	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
50-25-25	83.65	85.07	81.54	89.28	82.97	87 ± 13	0.91 ± 0.14
60-20-20	91.87	91.96	90.73	94.38	91.15	98 ± 18	1.04 ± 0.2
70-15-15	**98.28**	**99.53**	**98.68**	**98.74**	**98.13**	109 ± 17	1.11 ± 0.15
80-10-10	97.32	97.94	97.12	97.76	97.04	112 ± 23	1.13 ± 0.22

TABLE 9. Results of different initial learning rates on D3 dataset.

Learning Rate	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
0.01	95.12	95.87	96.27	96.85	94.87	101 ± 13	0.99 ± 0.1
0.001	96.99	97.68	97.14	97.78	96.66	108 ± 15	1.01 ± 0.21
0.0001	97.16	97.49	96.86	97.52	96.74	112 ± 14	1.05 ± 0.19
0.00001	98.02	99.11	98.32	98.42	98.52	121 ± 16	1.02 ± 0.17
0.000001	**98.76**	**98.43**	**98.15**	**98.71**	**98.55**	125 ± 13	1.00 ± 0.2

TABLE 10. Results of training-testing-validation data split on D3 dataset.

Data Split	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1 (%)	TT (m)	PT (s)
50-25-25	82.63	83.07	83.70	94.53	82.99	104 ± 11	1.01 ± 0.14
60-20-20	89.71	88.73	87.21	96.01	88.54	119 ± 14	1.11 ± 0.12
70-15-15	**99.16**	**99.49**	**98.86**	**99.52**	**98.74**	131 ± 12	0.94 ± 0.18
80-10-10	98.76	98.43	98.15	98.71	98.55	125 ± 13	1.00 ± 0.2

from NAS, which optimized the model's architecture for superior performance. The trade-off was apparent in the TT and processing time PT of both models, rendering them less efficient when resources were nearly depleted. MobileNetV2, on the contrary, necessitated low TT and PT while remaining suitable for real-time field deployment. The areas beyond the background were diminished, while feature extraction and robustness were improved, leading to the suggested EL outperforming each separate architecture when integrated. The model's excellent sensitivity and accuracy were further boosted by precise learning rates, such as 0.00001. Data divisions of 70-15-15 demonstrated superior generalization efficacy compared to skewed ratios like 80-10-10.

CNN models for wheat disease classification show a trade-off between accuracy and computational efficiency. Advanced architectures like InceptionResNetV2 and NasNet-Large may capture complicated patterns with great precision and sensitivity, but their processing costs make them unsuitable for real-time deployment. For resource-constrained applications, lightweight models like MobileNetV2 provide competitive accuracy and fast processing. DenseNet201 reuses and generalizes features well but uses more resources. By leveraging complementary strengths, the proposed EL model outperforms individual architectures in accuracy, sensitivity, and specificity while maintaining moderate computational requirements, making it a robust and scalable wheat disease classification model.

The distinctions between individual CNN designs and the EL method are apparent in its challenges with environmental unpredictability and data imbalance, highlighting the robustness of its approach. Numerous factors, like picture quality and background complexity, typically influence the efficacy of classical models, contributing to environmental unpredictability that is challenging to manage. Integrative techniques that combine multiple CNNs with diverse feature extraction capabilities enhance the EL method, which aids in addressing environmental unpredictability. Moreover, the incorporation of Bagging into the EL framework enhances the model's robustness by training on several data subsets, hence minimizing the impact of biased or non-representative datasets. For example, MobileNetV2 or ResNet50 may maintain generalization despite an unbalanced dataset, as the elastic conjoint technique leverages ensemble variability to enhance cross-class generalization. Spatial and channel attention modules can emphasize the significance of certain illness subtleties while mitigating noise interference from the complex background, aligning with other pre-trained CNN attention processes inside EL frameworks. The integration of multi-level feature extraction with bagging guarantees great accuracy and precision, while maintaining consistent performance across datasets with varying distributions.

The proposed EL approach has important practical consequences for the detection of wheat diseases in various agricultural settings and environments. The observation and diagnosis of multiple wheat diseases can be done accurately and expeditiously with the aid of disease monitoring systems, which are EL optimally trained for, as they yield tremendous accuracy across different data sets. This technique can enhance crop productivity and promote food security by balancing pesticide usage and improving disease control measures. Nonetheless, more resources for training and inference based on the ensemble structure require additional resources as they increase the computational complexity and have higher costs than lightweight models. Moreover, the heterogeneity in environmental conditions, light conditions, and image quality in real-life applications of the systems is likely to introduce some problems that will need effective preprocessing methods and thoughtful optimization of the system to alleviate these issues.

V. CONCLUSION

This research presents an efficient model for classifying wheat diseases. The approach is designed to identify these diseases quickly and effectively during complicated field settings. The model is notable for its reduced parameter count and computing needs, which allow for adaptive feature refinement on the input feature map. It does this by combining three pre-trained CNN models within an EL framework. Experimental comparisons confirm the practicality of the proposed model, providing a dependable and accurate technical remedy for identifying and detecting wheat diseases. While the proposed model demonstrates high accuracy, its computational requirements remain a barrier for real-time applications in resource-constrained settings. Future work will focus on optimizing the model for lightweight deployment. Nevertheless, it is recognized that the proposed model requires a greater computing burden in comparison to traditional lightweight models, therefore requiring additional enhancements in future endeavors. The study emphasizes the influence of image quality on the performance of models, emphasizing the necessity for improvements in crop disease image dataset and overall image quality. Future efforts will focus on creating a comprehensive and reliable dataset for wheat diseases, incorporating insect detection as it relates to wheat diseases, and integrating disease severity detection to help farmers make informed decisions about pesticide use to reduce pollution.

ACKNOWLEDGMENT

The authors express their appreciation to HITEC University Taxila, Kaunas University of Technology, AlMaarefa University, Applied INTelligence Lab (AINTLab), and Sejong University for their invaluable assistance and support during this research. Their contributions have significantly enhanced the caliber and breadth of this work.

(Samia Nawaz Yousafzai and Norma Latif Fitriyani are co-first authors.)

REFERENCES

[1] R. N. Strange and P. R. Scott, "Plant disease: A threat to global food security," *Annu. Rev. Phytopathol.*, vol. 43, no. 1, pp. 83–116, Sep. 2005.

[2] Á. Mesterházy, J. Oláh, and J. Popp, "Losses in the grain supply chain: Causes and solutions," *Sustainability*, vol. 12, no. 6, p. 2342, Mar. 2020.

[3] Y. Gao, H. Wang, M. Li, and W.-H. Su, "Automatic tandem dual BlendMask networks for severity assessment of wheat fusarium head blight," *Agriculture*, vol. 12, no. 9, p. 1493, Sep. 2022.

[4] M. Figueroa, K. E. Hammond-Kosack, and P. S. Solomon, "A review of wheat diseases—A field perspective," *Mol. Plant Pathol.*, vol. 19, no. 6, pp. 1523–1536, Jun. 2018.

[5] Y. Chai, S. Senay, D. Horvath, and P. Pardey, "Multi-peril pathogen risks to global wheat production: A probabilistic loss and investment assessment," *Frontiers Plant Sci.*, vol. 13, Oct. 2022, Art. no. 1034600.

[6] T. Kloppe, W. Boshoff, Z. Pretorius, D. Lesch, B. Akin, A. Morgounov, V. Shamanin, P. Kuhnem, P. Murphy, and C. Cowger, "Virulence of Blumeria graminis f. Sp. tritici in Brazil, South Africa, Turkey, Russia, and Australia," *Frontiers Plant Sci.*, vol. 13, Aug. 2022, Art. no. 954958.

[7] S. Tehsin, S. Rehman, M. O. B. Saeed, F. Riaz, A. Hassan, M. Abbas, R. Young, and M. S. Alam, "Self-organizing hierarchical particle swarm optimization of correlation filters for object recognition," *IEEE Access*, vol. 5, pp. 24495–24502, 2017.

[8] S. Tehsin, S. Rehman, A. Bilal, Q. Chaudry, O. Saeed, M. Abbas, and R. Young, "Comparative analysis of zero aliasing logarithmic mapped optimal trade-off correlation filter," *Proc. SPIE*, vol. 10203, pp. 22–37, May 2017.

[9] S. Tehsin, S. Rehman, F. Riaz, O. Saeed, A. Hassan, M. Khan, and M. S. Alam, "Fully invariant wavelet enhanced minimum average correlation energy filter for object recognition in cluttered and occluded environments," *Proc. SPIE*, vol. 10203, pp. 28–39, May 2017.

[10] S. Tehsin, Y. Asfia, N. Akbar, F. Riaz, S. Rehman, and R. Young, "Selection of CPU scheduling dynamically through machine learning," *Proc. SPIE*, vol. 11400, pp. 67–72, Apr. 2020.

[11] S. M. Saad, A. Bilal, S. Tehsin, and S. Rehman, "Spoof detection for fake biometric images using feature-based techniques," *Proc. SPIE*, vol. 11525, pp. 342–349, Nov. 2020.

[12] S. Tehsin, S. Rehman, A. B. Awan, Q. Chaudry, M. Abbas, R. Young, and A. Asif, "Improved maximum average correlation height filter with adaptive log base selection for object recognition," *Proc. SPIE*, vol. 9845, pp. 29–41, Apr. 2016.

[13] N. Akbar, S. Tehsin, A. Bilal, S. Rubab, S. Rehman, and R. Young, "Detection of moving human using optimized correlation filters in homogeneous environments," *Proc. SPIE*, vol. 11400, pp. 73–79, May 2020.

[14] N. Akbar, S. Tehsin, H. Ur Rehman, S. Rehman, and R. Young, "Hardware design of correlation filters for target detection," *Proc. SPIE*, vol. 10995, pp. 71–79, May 2019.

[15] Y. Asfia, S. Tehsin, A. Shahzeen, and U. S. Khan, "Visual person identification device using raspberry Pi," in *Proc. 25th Conf. FRUCT Assoc.*, 2019, pp. 1–7.

[16] A. Dixit and S. Nema, "Wheat leaf disease detection using machine learning method—A review," *Int. J. Comput. Sci. Mob. Comput.*, vol. 7, no. 5, pp. 124–129, May 2018.

[17] D. Kumar and V. Kukreja, "Deep learning in wheat diseases classification: A systematic review," *Multimedia Tools Appl.*, vol. 81, no. 7, pp. 10143–10187, Mar. 2022.

[18] S. H. Lee, H. Goëu, P. Bonnet, and A. Joly, "New perspectives on plant disease characterization based on deep learning," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105220.

[19] M. Nagaraju and P. Chawla, "Systematic review of deep learning techniques in plant disease detection," *Int. J. Syst. Assurance Eng. Manage.*, vol. 11, no. 3, pp. 547–560, Jun. 2020.

[20] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, Oct. 2020.

[21] I. M. Nasir, M. Raza, J. H. Shah, S.-H. Wang, U. Tariq, and M. A. Khan, "HAREDNet: A deep learning based architecture for autonomous video surveillance by recognizing human actions," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107805.

[22] I. M. Nasir, M. Rashid, J. H. Shah, M. Sharif, M. Y. H. Awan, and M. H. Alkinani, "An optimized approach for breast cancer classification for histopathological images based on hybrid feature set," *Current Med. Imag. Formerly Current Med. Imag. Reviews)*, vol. 17, no. 1, pp. 136–147, Mar. 2021.

[23] I. M. Nasir, M. Raza, J. H. Shah, M. Attique Khan, and A. Rehman, "Human action recognition using machine learning in uncontrolled environment," in *Proc. 1st Int. Conf. Artif. Intell. Data Analytics (CAIDA)*, Apr. 2021, pp. 182–187.

[24] I. M. Nasir, A. Bibi, J. H. Shah, M. A. Khan, M. Sharif, K. Iqbal, Y. Nam, and S. Kadry, "Deep learning-based classification of fruit diseases: An application for precision agriculture," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 1949–1962, 2021.

[25] M. A. Khan, I. M. Nasir, M. Sharif, M. Alhaisoni, S. Kadry, S. A. C. Bukhari, and Y. Nam, "A blockchain based framework for stomach abnormalities recognition," *Comput., Mater. Continua*, vol. 67, no. 1, pp. 141–158, 2021.

[26] I. M. Nasir, M. Raza, J. H. Shah, M. A. Khan, Y.-C. Nam, and Y. Nam, "Improved shark smell optimization algorithm for human action recognition," *Comput., Mater. Continua*, vol. 76, no. 3, pp. 2667–2684, 2023.

[27] I. M. Nasir, M. Raza, S. M. Ulyah, J. H. Shah, N. L. Fitriyani, and M. Syafrudin, "ENGA: Elastic Net-based genetic algorithm for human action recognition," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120311.

[28] J. Tariq, A. Alfalou, A. Ijaz, H. Ali, I. Ashraf, H. Rahman, A. Armghan, I. Mashood, and S. Rehman, "Fast intra mode selection in HEVC using statistical model," *Comput., Mater. Continua*, vol. 70, no. 2, pp. 3903–3918, 2022.

[29] I. Mushtaq, M. Umer, M. Imran, I. M. Nasir, G. Muhammad, and M. Shorfuzzaman, "Customer prioritization for medical supply chain during COVID-19 pandemic," *Comput., Mater. Continua*, vol. 70, no. 1, pp. 59–72, 2022.

[30] I. M. Nasir, M. A. Khan, M. Yasmin, J. H. Shah, M. Gabryel, R. Scherer, and R. Damasevicius, "Pearson correlation-based feature selection for document classification using balanced training," *Sensors*, vol. 20, no. 23, p. 6793, Nov. 2020.

[31] I. M. Nasir, M. A. Khan, A. Armghan, and M. Y. Javed, "SCNN: A secure convolutional neural network using blockchain," in *Proc. 2nd Int. Conf. Comput. Inf. Sci. (ICCIS)*, Oct. 2020, pp. 1–5.

[32] I. M. Nasir, M. A. Khan, M. Alhaisoni, T. Saba, A. Rehman, and T. Iqbal, "A hybrid deep learning architecture for the classification of superhero fashion products: An application for medical-tech classification," *Comput. Model. Eng. Sci.*, vol. 124, no. 3, pp. 1017–1033, 2020.

[33] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, p. 1319, Oct. 2020.

[34] D. Argüeso, A. Picon, U. Irusta, A. Medela, M. G. San-Emeterio, A. Bereciartua, and A. Alvarez-Gila, "Few-shot learning approach for plant disease classification using images taken in the field," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105542.

[35] P. Zhang, L. Yang, and D. Li, "EfficientNet-B4-ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment," *Comput. Electron. Agricult.*, vol. 176, Sep. 2020, Art. no. 105652.

[36] P. Goncharov, A. Uzhinskiy, G. Ososkov, A. Nechaevskiy, and J. Zudikhina, "Deep Siamese networks for plant disease detection," in *Proc. EPJ Web Conf.*, vol. 226, 2020, pp. 1–4.

[37] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105393.

[38] S. Hernández and J. L. López, "Uncertainty quantification for plant disease detection using Bayesian deep learning," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106597.

[39] M. A. Genaev, E. S. Skolotneva, E. I. Gultyaeva, E. A. Orlova, N. P. Bechtold, and D. A. Afonnikov, "Image-based wheat fungi diseases identification by deep learning," *Plants*, vol. 10, no. 8, p. 1500, Jul. 2021.

[40] T. Dong, X. Ma, B. Huang, W. Zhong, Q. Han, Q. Wu, and Y. Tang, "Wheat disease recognition method based on the SC-ConvNeXt network model," *Sci. Rep.*, vol. 14, no. 1, p. 32040, Dec. 2024.

[41] A. Adedoja, P. A. Owolawi, and T. Mapayi, "Deep learning based on NASNet for plant disease recognition using leave images," in *Proc. Int. Conf. Adv. Big Data, Comput. Data Commun. Syst.*, Aug. 2019, pp. 1–5.

[42] A. Hussain, M. Ahmad, I. A. Mughal, and H. Ali, "Automatic disease detection in wheat crop using convolution neural network," in *Proc. 4th Int. Conf. Next Generat. Comput.*, 2018, pp. 1–4.

[43] X. Zhang, L. Han, Y. Dong, Y. Shi, W. Huang, L. Han, P. González-Moreno, H. Ma, H. Ye, and T. Sobeih, "A deep learning-based approach for automated yellow rust disease detection from high-resolution hyperspectral UAV images," *Remote Sens.*, vol. 11, no. 13, p. 1554, Jun. 2019.

[44] N. Jahan, P. Flores, Z. Liu, A. Friskop, J. Mathew, and Z. Zhang, "Detecting and distinguishing wheat diseases using image processing and machine learning algorithms," in *Proc. ASABE Annu. Int. Virtual Meeting*, 2020, pp. 1–9.

[45] M. Dong, S. Mu, A. Shi, W. Mu, and W. Sun, "Novel method for identifying wheat leaf disease images based on differential amplification convolutional neural network," *Int. J. Agricult. Biol. Eng.*, vol. 13, no. 4, pp. 205–210, 2020.

[46] Q. Pan, M. Gao, P. Wu, J. Yan, and S. Li, "A deep-learning-based approach for wheat yellow rust disease recognition from unmanned aerial vehicle images," *Sensors*, vol. 21, no. 19, p. 6540, Sep. 2021.

[47] W. Bao, J. Zhao, G. Hu, D. Zhang, L. Huang, and D. Liang, "Identification of wheat leaf diseases and their severity based on elliptical-maximum margin criterion metric learning," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100526.

[48] X. Fang, T. Zhen, and Z. Li, "Lightweight multiscale CNN model for wheat disease detection," *Appl. Sci.*, vol. 13, no. 9, p. 5801, May 2023.

[49] W. Bao, X. Yang, D. Liang, G. Hu, and X. Yang, "Lightweight convolutional neural network model for field wheat ear disease identification," *Comput. Electron. Agricult.*, vol. 189, Oct. 2021, Art. no. 106367.

[50] L. Goyal, C. M. Sharma, A. Singh, and P. K. Singh, "Leaf and spike wheat disease detection & classification using an improved deep convolutional architecture," *Informat. Med. Unlocked*, vol. 25, Jan. 2021, Art. no. 100642.

[51] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of Rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Comput. Electron. Agricult.*, vol. 186, Jul. 2021, Art. no. 106184.

[52] V. Kukreja and D. Kumar, "Automatic classification of wheat rust diseases using deep convolutional neural networks," in *Proc. 9th Int. Conf. Rel., Infocom. Technol. Optim. (Trends Future Directions) (ICRITO)*, Sep. 2021, pp. 1–6.

[53] T. Aboneh, A. Rorissa, R. Srinivasagan, and A. Gemechu, "Computer vision framework for wheat disease identification and classification using Jetson GPU infrastructure," *Technologies*, vol. 9, no. 3, p. 47, Jul. 2021.

[54] Q. Pan, M. Gao, P. Wu, J. Yan, and M. A. E. AbdelRahman, "Image classification of wheat rust based on ensemble learning," *Sensors*, vol. 22, no. 16, p. 6047, Aug. 2022.

[55] S. Akbar, K. T. Ahmad, M. K. Abid, and N. Aslam, "Wheat disease detection for yield management using IoT and deep learning techniques," *VFAST Trans. Softw. Eng.*, vol. 10, no. 3, pp. 80–89, Sep. 2022.

[56] T. Zhang, Z. Yang, Z. Xu, and J. Li, "Wheat yellow rust severity detection by efficient DF-UNet and UAV multispectral imagery," *IEEE Sensors J.*, vol. 22, no. 9, pp. 9057–9068, May 2022.

[57] H. Khan, I. U. Haq, M. Munsif, Mustaqeem, S. U. Khan, and M. Y. Lee, "Automated wheat diseases classification framework using advanced machine learning technique," *Agriculture*, vol. 12, no. 8, p. 1226, Aug. 2022.

[58] L. Xu, B. Cao, F. Zhao, S. Ning, P. Xu, W. Zhang, and X. Hou, "Wheat leaf disease identification based on deep learning algorithms," *Physiol. Mol. Plant Pathol.*, vol. 123, Jan. 2023, Art. no. 101940.

[59] O. Jouini, K. Sethom, and R. Bouallegue, "Wheat leaf disease detection using CNN in smart agriculture," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2023, pp. 1660–1665.

[60] U. Shafi, R. Mumtaz, M. D. M. Qureshi, Z. Mahmood, S. K. Tanveer, I. U. Haq, and S. M. H. Zaidi, "Embedded AI for wheat yellow rust infection type classification," *IEEE Access*, vol. 11, pp. 23726–23738, 2023.

[61] S. Saraswat, S. Batra, P. P. Neog, E. L. Sharma, P. P. Kumar, and A. K. Pandey, "An efficient diagnostic approach for multi-class classification of wheat leaf disease using deep transfer and ensemble learning," in *Proc. 2nd Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT)*, Jan. 2024, pp. 544–551.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[63] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[65] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[66] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 1–7.

[67] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[68] K. Bansal, R. K. Batla, Y. Kumar, and J. Shafi, "Artificial intelligence techniques in health informatics for oral cancer detection," in *Connected E-Health: Integrated IoT and Cloud Computing*. Cham, Switzerland: Springer, 2022, pp. 255–279.

[69] D. AlSaeed and S. F. Omar, "Brain MRI analysis for Alzheimer's disease diagnosis using CNN-based feature extraction and machine learning," *Sensors*, vol. 22, no. 8, p. 2911, Apr. 2022.

[70] Y. Kumar and S. Gupta, "Deep transfer learning approaches to predict glaucoma, cataract, choroidal neovascularization, diabetic macular edema, DRUSEN and healthy eyes: An experimental review," *Arch. Comput. Methods Eng.*, vol. 30, no. 1, pp. 521–541, Jan. 2023.

[71] K. Bansal, R. K. Bathla, and Y. Kumar, "Deep transfer learning techniques with hybrid optimization in early prediction and diagnosis of different types of oral cancer," *Soft Comput.*, vol. 26, no. 21, pp. 11153–11184, Sep. 2022.

[72] A. Koul, R. K. Bawa, and Y. Kumar, "Artificial intelligence techniques to predict the airway disorders illness: A systematic review," *Arch. Comput. Methods Eng.*, vol. 30, no. 2, pp. 831–864, Mar. 2023.

[73] N. Chaplot, D. Pandey, Y. Kumar, and P. S. Sisodia, "A comprehensive analysis of artificial intelligence techniques for the prediction and prognosis of genetic disorders using various gene disorders," *Arch. Comput. Methods Eng.*, vol. 30, no. 5, pp. 3301–3323, Jun. 2023.

[74] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique," *Sustainability*, vol. 15, no. 3, p. 1906, Jan. 2023.

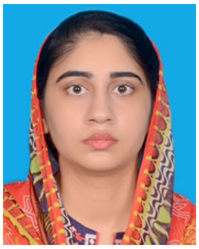
[75] Y. Tolkach, T. Dohmgörgen, M. Toma, and G. Kristiansen, "High-accuracy prostate cancer pathology using deep learning," *Nature Mach. Intell.*, vol. 2, no. 7, pp. 411–418, Jul. 2020.

[76] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[77] M. Long, M. Hartley, R. J. Morris, and J. K. M. Brown, "Classification of wheat diseases using deep learning networks with field and glasshouse images," *Plant Pathol.*, vol. 72, no. 3, pp. 536–547, Apr. 2023.

SAMIA NAWAZ YOUSAFZAI received the B.S. degree in software engineering from HITEC University, Taxila, Pakistan, in 2024. She is currently working on multiple projects related to medical imaging, agricultural imaging, sentiment analysis, and fake news classification. Her research interests include encompass deep learning, digital image processing, computer vision, and natural language processing.

INZAMAM MASHOOD NASIR (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in computer science from COMSATS University Islamabad, Pakistan, in 2012, 2016, and 2023, respectively. He is currently working on privacy-preserving techniques by embedding blockchain and IoTs with machine-learning techniques for real-world applications. His most recent projects are based on federated learning, explainable AI, and different enhancement techniques to improve the efficiency of models in real-world applications. He is a big fan of nature-inspired algorithms, thus one of his key interests is bio-inspired optimization algorithms. His research interests include machine learning (ML) for medical imaging, agricultural imaging, and hyperspectral imaging.



SARA TEHSIN (Student Member, IEEE) received the bachelor's degree in computer engineering from The Islamia University of Bahawalpur, in 2013, and the master's degree in computer engineering from the National University of Sciences and Technology, Islamabad, in 2016. She is currently pursuing the Ph.D. degree with Kaunas University of Technology, Lithuania. Her most recent projects are based on integrating explainability in transformers for improved detection of brain tumors. She is actively involved in the Washington Accord accreditation procedure, where she has as an experience of over five years. Her research interests include machine learning (ML) for medical imaging, agricultural imaging, and hyperspectral imaging. She has attended several national and international conferences.

DANIA SALEEM MALIK received the Ph.D. degree in mathematics from Quaid-i-Azam University, Islamabad, Pakistan, in 2021. She is currently a Lecturer with the Department of Mathematics, HITEC University, Taxila, Pakistan. Her research interests include large-scale coding theory, cryptography, and information theory.

ISMAIL KESHTA (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in computer science and engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2009, 2011, and 2016, respectively. In 2011, he was a Lecturer with Princess Nourah Bint Abdulrahman University and Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. He was a Lecturer with the Computer Engineering Department, KFUPM, from 2012 to 2016. He is currently an Assistant Professor with the Department of Computer Science and Information Systems, Almaarefa University, Riyadh. His research interests include software process improvement, modeling, and intelligent systems.

NORMA LATIF FITRIYANI received the B.S. degree from UIN Sunan Kalijaga Yogyakarta, Indonesia, in 2014, the M.S. degree from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2016, and the Ph.D. degree from Dongguk University, Seoul, South Korea, in 2021. She has been an Assistant Professor with Sejong University, since 2022. With over seven years of experience in data science, statistical and machine learning, informatics, and image processing, her research has been published in journal articles, such as COMPAG, ESWA, AEJ, *Mathematics*, *Sustainable Development*, *Food Control*, and IEEE Access. She also serves as a guest editor and a reviewer for several international peer-reviewed journals.

YEONGHYEON GU received the B.S., M.E., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, South Korea, in 2004, 2006, and 2014, respectively. He is currently the Director of the AI Convergence Research Center and a Professor with the Department of Artificial Intelligence and Data Science, Sejong University. His research interests include AI, NLP, meta-learning, transfer learning, and related fields.

MUHAMMAD SYAFRUDIN received the B.S. degree from UIN Sunan Kalijaga Yogyakarta, Indonesia, in 2013, and the Ph.D. degree from Dongguk University, Seoul, South Korea, in 2019. He was an Assistant Professor with Dongguk University, from 2019 to 2022. He has been an Assistant Professor with Sejong University, since 2022. With over ten years of experience in industrial artificial intelligence, data intelligence, industrial analytics, and informatics. His research has been published in journals, such as ESWA, AEJ, COMPAG, *Food Control*, *Sustainable Development*, *Mathematics*, and IEEE Access. He is listed among Stanford/Elsevier's Top 2% Scientists for 2024 in Artificial Intelligence and Image Processing. He serves as a guest editor and a reviewer for Scopus- and SCI-indexed journals.