International Journal of Basic and Applied Sciences, 14 (1) (2025) 156-165

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/pr4acd16
Research paper

A convolutional neural network with average pooling for chickpea disease detection and classification

CH. Srilakshmi [1], Gudepu Sridevi [2], Mummidi Krishnaveni [3], Ch Sekhar [4] *, J. Vamsinath [5], Nagamalli Arasavalli [6], Mirtipati Satish Kumar [7]

[1] Department of Computer Engineering & Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India
[2] Department of Mechanical Engineering, Vignan's Institute of Information Technology, Visakhapatnam, Andhra Pradesh, India
[3] Department of Computer Science and Engineering (AI&ML), Anil Neerukonda Institute of Technology and Sciences(A), Sangivalasa, Visakhapatnam, Andhra Pradesh, India
[4] Department of Computer Science and Engineering (AI&ML), GMR Institute of Technology(A), Rajam, Andhra Pradesh, India
[5] Department of Computer Science and Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, Telangana, India
[6] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, KL (Deemed to be University) Vaddeswaram, Guntur District, Andhra Pradesh, India
[7] Department of Computer Science and Engineering, CENTURION University of Technology and Management, Vizianagaram, Andhra Pradesh, India
**Corresponding author E-mail: sekharch.gmrit@gmail.com*

Received: April 11, 2025, Accepted: April 30, 2025, Published: May 3, 2025

Abstract

Chickpeas rank among the top legumes worldwide, yet diseases can hit them hard, slashing both quality and yield. Accurate disease classification matters a lot for managing them well. In the Proposed research work use a Convolutional Neural Network (CNN) with Average Pooling to identify and categorize diseases of chickpea plants. Primary Data used in this work are Chickpea and Fusarium-22 datasets PrePre-processing was done through demonising to maintain the purity of the data. To balance the classes, Data augmentation is done, which boosts the dataset size, with key features drawn out using the AlexNet architecture, before the CNN-Average Pooling steps in to classify the diseases. The CNN automatically snatches crucial details from chickpea images, while the Average Pooling layer slices down spatial dimensions to interesting patterns that are more useful. This arrangement brings precise disease classification, making it beneficial for agriculture and crop management. Tests show the CNN-Average Pooling hitting 99.60% accuracy on the Chickpea dataset and 99.53% on Fusarium-22, exceeding options like CNN-LSTM, the GLCM-Color Histogram combo, and DenseNet-201

Keywords: *Alexnet; Average Pooling; Chickpea Disease; Convolutional Neural Network; Image Denoising.*

1. Introduction

In emergent nations, agriculture and its products play an essential role in modelling their economies. A significant impact of plant diseases is on the growth, productivity, and financial stability of all countries. In recent years, improvements in digital image processing have significantly contributed to refining agricultural predictions and enhancing productivity for accurate disease detection and monitoring [1-2]. Chickpeas are a valuable crop for sustainable agriculture, helping to preserve soil fertility and improve the main characteristics of the soil, while also reducing pests, diseases, and weeds throughout the rainy season [3]. Chickpeas can be classified into two main types: Daisy and Kabuki, based on distinct morphological traits such as seed coat thickness, color, shape, and size [4]. Mixing different types should be avoided to retain their physico-chemical characteristics [5] [6]. Sometimes, disease causes lesions and harms the shoot of the chickpea plant, affecting all parts of the shoot, including the leaves, stems, and pods [7]. Many background images is required in datasets to develop the composite disease classification system [8].

Unlike other machine learning techniques, deep learning does not require prior arrangement of features [9]. Deep Learning (DL) has enormous significance and greatly contributes to people's daily lives [10] [11]. One of the pillars of precision agriculture is the timely and accurate identification of crop diseases. Most plant diseases are diagnosed manually, which is a challenging and time-consuming process [12]. Thus, there is great demand for an automated, timely with a more accurate model for forecasting and identifying plant diseases. Many tools for accurate and timely disease forecasting in plants have been developed as an outcome of recent advancements in Artificial Intelligence (AI) [13]. The Convolutional Neural Network (CNN) model is designed for automatic feature extraction by learning weight values that reflect the relationship between local patterns, including those found in chickpeas, tomatoes, cucumbers, and bananas [14]. These technologies aim to empower farmers with the ability to quickly identify diseases and propose preventive strategies to minimise crop losses [15]. The main contributions of this research are given below:

- A CNN-Average pooling approach is implemented using the Chickpea and Fusarium-22 datasets to collect plant images from various environments.
- Pre-processing is performed through image denoising, which helps reduce unwanted noise, followed by data augmentation to enhance the dataset size.
- The AlexNet is employed to extract the optimal features, while the proposed method is used for classification. The performance of the implemented approach is analysed in terms of precision, accuracy, f-measure, and recall.

This research paper is further structured as follows: Section 2 reviews the current models, Section 3 provides a in-depth description of the proposed method, Section 4 presents a reasonable analysis of the proposed method’s performance against that of the existing models, and to conclude, Section 5 of the research work.

2. Literature survey

Belay [16] implemented a CNN and Long Short-Term Memory (CNN-LSTM) model for the detection and classification of chickpea diseases. The chickpea dataset was utilized to effectively evaluate the performance of the CNN-LSTM method. This CNN-LSTM method enabled the accurate identification of chickpea crop diseases and reduced the amount of chickpea production lost. However, the CNN-LSTM method required enhancement for more efficient segmentation, as it struggled to identify certain chickpea diseases. Additionally, the CNN-LSTM approach had complex outputs and high processing times, which hindered real-time disease detection.

Abuhayi and Bezabih [17] implemented a hybrid method for the classification of chickpea diseases. This hybrid method integrated the Gray-Level Co-occurrence Matrix (GLCM)-Color Histogram with non-local means and Bilateral filtering. The combined GLCM and Color Histogram models were integrated with the SVM approach to achieve robust outcomes through combined features. However, due to low classification performance for chickpea diseases, this method required alternative feature extraction approaches to improve the classification process. Furthermore, it resulted in high processing times when evaluated with the filtering techniques.

Çetin [18] implemented Principal Component Analysis (PCA) to identify superior cultivars for specific qualities and highlight certain cultivars with traits. Data from different chickpea cultivars were used to evaluate the PCA method, and statistical parameters for seed mass predictions were also used to obtain effective results. Nonetheless, darker background colors were not favorable when selecting images, which made image segmentation challenging. Moreover, PCA involved matrix multiplication, leading to high processing delays.

To help with issues related to producing food and agriculture, Andreas [19] presents 40 studies that used deep learning. It looks at the data type, pre-processing methods, the stated performance outcomes, the kinds of agricultural issues dealt with, and the deep learning models used. The article also contrasts the classification and regression outcomes achieved by deep learning techniques with the outcomes achieved by more conventional approaches. Deep learning algorithms provide better accuracy in agricultural applications, often surpassing standard image processing methods, as highlighted in the review.

The increasing need for alternatives to foreign medications is addressed by Bodduru [20], who investigates the use of machine learning algorithms for predicting medicinal herbs based on symptoms. To detect illnesses and suggest suitable medicinal leaves, the research makes use of three machine learning techniques: Random Forest, Multinomial Naive Bayes, and Gradient Boosting. The Multinomial Naive Bayes algorithm outperformed the rest with a 92% success rate. The results indicate that machine learning, and specifically Multinomial Naive Bayes, can successfully combine conventional Ayurvedic wisdom with cutting-edge scientific understanding, opening the door to safer, more effective therapeutic options.

Hayit [21] implemented a Gray-Level Run-Length Matrix- hue-saturation value-based K-nearest neighbours (GLRLM-HSV-based KNN) method. This method was used for Fusarium wilt disease classification based on the texture and color features of chickpea. The method achieved high classification accuracy for Fusarium wilt in chickpea, particularly at high severity levels. However, due to small dataset, the method got stuck in local minima, resulting in underfitting or overfitting issues. Additionally, extracting statistical features made the GLRLM-HSV-based KNN method computationally expensive.

Hayit [22] implemented a pre-trained DenseNet-201 CNN approach based on a transfer learning model for Fusarium wilt disease classification in chickpeas. The model utilized the Fusarium-22 dataset to assess disease severity. The DenseNet-201-based method achieved better accuracy rates, while the CNN model provided high capability for analyzing and classifying complex image data. However, implementing the pre-trained DenseNet-201 CNN approach was time-consuming and challenging during classification. The complexity of DenseNet-201 increased when it was integrated with the pre-trained model for disease classification.

3. Methodology

In this work, a CNN-Average pooling approach is implemented for chickpea disease detection and classification. Fig. 1 illustrates the block diagram of the implemented method. This work utilizes the chickpea and Fusarium-22 datasets for data collection, along with an image denoising technique for pre-processing and noise removal. Data augmentation is applied to improve the dataset size, and AlexNet is employed to extract features from the augmented data. Finally, the proposed pooling approach is used for the classification of chickpea diseases.

3.1. Dataset

This research utilizes the Chickpea and Fusarium-22 datasets, which are collected from different environments. To compile a comprehensive dataset, chickpea images are captured using a variety of devices, including tablets and smartphones. In Gondar, Ethiopia, 1247 RGB images are taken with a Samsung A10 phone and an ONN tablet, which have 15-megapixel and 13-megapixel cameras, respectively. The Fusarium-22 dataset consists of 3000 images for each class, with a total of 15,000 chickpea plant images. This Fusarium-22 dataset is divided into three groups: training (70%), validation (20%), and testing (10%).
(https://www.kaggle.com/datasets/tolgahayit/fusarium-wilt-disease-in-chickpea-dataset/code).

Images of chickpea plants afflicted with Fusarium wilt, a deadly disease that affects chickpea crops, are collected in the Fusarium-22 dataset. The various environmental factors include temperature (20°C to 40°C), humidity (40% to 80%), soil types (clay, loam, and sand), and irrigation levels (drought-stressed, well-watered). Additionally, the dataset includes images captured under various lighting conditions, such as shading, artificial light, and natural light. In contrast, there are many types of chickpeas, including Kabuli, Desi, Spanish, seedling, vegetative, and reproductive varieties. To document the progression of the disease, images of plants with mild, moderate, and severe Fusarium wilt infections are taken. The Chickpea and Fusarium-22 dataset is enhanced by incorporating these various components, making it possible to create reliable and accurate learning models for studying chickpea plants and diagnosing diseases.

3.2. Pre-processing

After data collection, pre-processing is performed using the image denoising technique. In this work, noise is eliminated using the Perona-Malik (P-M) equation, which links the various image regions' features to diffusion. In addition to noise removal, a diffusion coefficient is applied in all directions to prevent edge smoothing, which varies according to the image gradient. The primary reason for applying the P-M equation is that the edges are not smoothed, thereby enhancing the overall effect, as mathematically expressed in Eq. (1).

$$\frac{\partial u}{\partial t} = \mathrm{div}\left(c(|\nabla u(x,y,t)|)\nabla u(x,y,t)\right) \tag{1}$$

Where, $u_t = r(x,y,t)$ refers to an image obtained at a time t, div refers to the divergence operator with the gradient operator ∇ in the x and y directions. Once the denoising is carried out, the output is fed into the data augmentation phase.

3.3. Data augmentation

Following pre-processing, data augmentation is applied, as a large amount of data is required for DL methods to effectively train and improve their performance. Data augmentation involves generating new training data from the existing dataset, thereby increasing its size. By augmenting the data, a larger and more diverse dataset is created, which enhances the model's ability to generalize and enables it to perform more effectively when exposed to new data. Therefore, if the dataset contains a large amount of diverse and well-labeled data, the model is trained more effectively, resulting in more accurate predictions. Conversely, a small dataset limits the model's performance. In this work, three data augmentation techniques: image rotation, image resizing, and image flipping, are used to expand the dataset. These techniques are explained below.

3.3.1. Image rotation

The process of rotating an image to create a new image is referred to as image rotation.

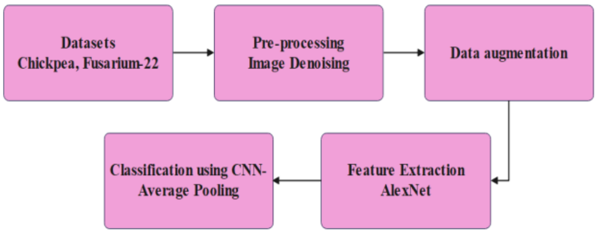


Fig. 1: Block Diagram of Proposed Model.

The pixel values remain unchanged before and after the image is rotated. The matrix formula for rotation is applied when the coordinate origin is selected as the rotation point, as mathematically represented in Eq. (2).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

Where, the rotation angle is selected randomly between from 0° to 360°, and is denoted as θ.

3.3.2. Image resizing

The image resizing technique modifies various pixels in the image, and these matrices are represented in Eq. (3), along the x and y-axes.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

Where, the resized image is denoted as (x', y'), while original image is referred to as (x, y). The scaling factors are denoted as S_y and S_x, along the y and x-axis. Also, 0.5 to 2 are the randomly chosen scaling factors.

3.3.3. Image flip

The vertical flip and horizontal flip matrix expressions are as shown in Eqs. (4) and (5).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

Where the height of the image is represented as h, and its width is denoted as w. Here, data augmentation not only enhances the model's performance but also assists in decreasing overfitting, which is a general issue in DL methods. Then, the data augmentation outcome is given to the feature extraction phase as an input.

3.4. Feature extraction

After data augmentation, feature extraction is carried out to extract features from the datasets. The extraction of features is an effective tool to preserve the significant, informative features for dimensionality reduction. This study uses the pre-trained AlexNet model to extract deep features from the augmented data.

3.4.1. AlexNet

One well-known CNN architecture that gained recognition in the 2012 ImageNet competition is AlexNet, which is an effective tool for image classification. The AlexNet architecture consists of fud, convolutional, and pooling layers. The input pixels are 227 × 227, and the network works by applying 3 × 3 or 5 × 5-pixel filters within the convolution layer of the image. As a result, the activation maps with more effective features are produced and transferred to the following layer. The activation maps have unique features, important for classification. The cost and size of the image are decreased without disturbing any of its features in the pooling layer. Fig. 2 illustrates the architecture of AlexNet.s The following steps outline the algorithm for utilizing and training AlexNet. The AlexNet architecture consists of eight layers, with the first five layers being Convolutional layers, followed by three maximum pooling layers and three completely connected layers.

The input layer of the dimension 224x224x3 is equipped with a convolutional approach containing a 11x11 filter, a stride of 4, and a pooling size of 2. The next layer processes the image with a pixel size of 55x55x96, where 96 refers to the number of filters applied. Following this, a max pooling strategy with a 3x3A A sized filter and a stride of 2 is utilized for enhancement. The next layer with the 27x27x96 dimensions, where 96 represents the number of filters applied. The same convolutional approach with a 5x5x96 filter is employed. The second layer is 27x27x256, meaning there are 256 filters in total, with each filter having the dimensions of 5x5x96 filters. Furthermore, maximum pooling is used again, using a filter of the size 3x3 and a stride of 2. The following layer has dimensions of 13x13x296 with has 256 filters. The following layer is 13x13x384 in size, meaning that there are 384 filters in total, with each filter containing 3x3x256 filters. The next layer, with dimensions of 13×13×384, applies 384 filters, each of which uses a 3×3×256 filter. A 3×3×384 filter is then used in the next convolutional layer. This layer has 384 filters in total, with dimensions 13×13×384. Max pooling is applied again with a 3×3 filter size and a stride of 2. The following layer has dimensions of 6×6×256, which corresponds to 9216 neurons (i.e., 6×6×256 = 9216). There are 36 neurons in each layer.

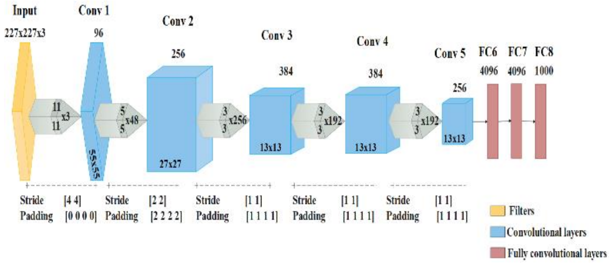


Fig. 2: Architecture of AlexNet.

The next step involves dense connections, where there are 4096 neurons in the first hidden layer, with 9216 connections between each of these neurons. The following hidden layer also has 4096 neurons and 4096×4096 connections fully connected to the neurons in the previous layer. Finally, the output layer contains 1000 neurons, each fully connected to the 100×4096 connections from the previous layer. The ReLU activation function is used, which outputs the input directly if it is positive and outputs zero otherwise. The extracted output features are then passed through the classification phase as input. After augmenting the data, we proceed to feature extraction.

3.5. Classification using CNN-average pooling layer

The CNN method is employed to categorize the extracted features to ensure the model provides outstanding results in several domains, including language processing, diagnosis systems, and image processing. In contrast to Multi-Layer Perceptron (MLP), the CNN uses fewer parameters and neurons, leading to minimal complexity and rapid adaptability. The CNN method is a type of DL and Feed-forward Neural Network (FFNN) employed for classifying diseases affecting chickpeas. The filter's location is determined autonomously due to the convolution operations and the constant detention convention, which lowers the number of parameters. The CNN architecture utilizes three kinds of layers: pooling, fully connected, and convolution layers, which facilitate dimensionality reduction, extraction, and classification. The filter is forwarded to the convolution process, and activation is obtained by adding the input capacity of the activation map, which analyzes the pointwise result of each score. The sliding filter performs a fast distribution of the dot product that is used by both linear and convolution operators. Considering the kernel function w, where x is input, $(x \times w)(a)$ at time t, as expressed in Eq. (6).

$$(x \times w)(a) = \int x(t)w(a - t)da \tag{6}$$

Where, a is R^n for each $n \geq 1$. The parameter t is discrete, as expressed in given in Eq. (7).

$$(x \times w)(a) = \sum_a x(t)w(t - a) \tag{7}$$

A 2D image I is input, and K is a 2D kernel. Equation 8 defines the convolution operation for feature extraction.

$$(I \times K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \tag{8}$$

The ReLU and SoftMax functions are two distinct activation functions used to increase the non-linearity. The ReLU is represented in Eq. (9).

$$\mathrm{ReLU}(x) = \max(0, x)\ x \in R \tag{9}$$

The gradient $\mathrm{ReLU}(x) = 1$ for $x > 0$ and $\mathrm{ReLU} - (x) = 0$ for $x < 0$. The convergence ability of ReLU is better compared to sigmoid nan-linearities. The Softmax layer follows, and it is preferred when the output needs to account for two or more classes, as expressed in Eq. (10).

$$\mathrm{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_i)} \tag{10}$$

The pooling layers are used to rescale the output structure without sacrificing any crucial information for producing an input static. The final layer, known as the Fully Connected (FC) layer, has inputs and outputs, represented by m and n, respectively. The weight matrix $W \in M_{m,n}$ is the output layer parameter, where b is a bias vector, and m and nThere are rows and columns in R^m. The output of the fully connected layer f is given by Eq. (11), where x refers to an input vector that lies within R^n.

$$FC(x) := f(Wx = b) \in R^m \tag{11}$$

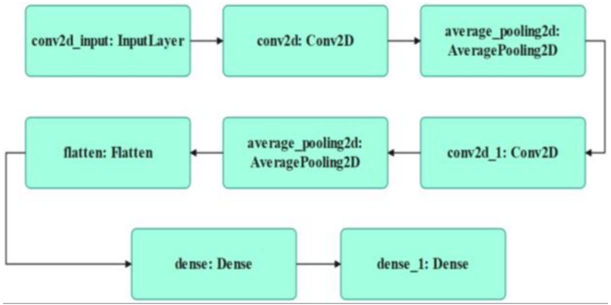


Fig. 3: Two Convolution Layers with Average Pooling Action.

Where, function f is used as the component, and Wx is the matrix product. This fully connected layer is used to address challenges related to classification.

An example of a DL algorithm is CNN, which uses an input image to assign priority to various parts of the image, to differentiate between images based on their attributes. This system utilizes one CNN model with convolutional layers, where each convolutional layer uses 2D convolutional operations. 'ReLU activation' is applied in both convolutional 2D layers. Two Dense layers are used to ensure full connectivity, with 'Sigmoid activation' applied in the second dense layer, and ReLU activation in the first dense layer.

Average Pooling layer: The average element from the feature map's covered area is chosen by a pooling process. By counting all values and passing them to the next layer through average pooling, all values are used for feature mapping and output generation, which is a complete computation. Fig. 3 represents the two-convolution layer with the implemented average pooling action.

4. Results and discussion

The implemented method is trained on chickpea and Fusarium-22 datasets, with the following system requirements: Windows 10 (64-bit) OS, Intel i7 processor, and 16 GB RAM. The implemented approach's effectiveness is calculated in terms of recall, precision, accuracy, and F1-score.

4.1. Evaluation metrics

For the evaluation of medical image analysis, the arithmetic representation of the metrics is given in Eqs. (12), (13), (14), and (15) [23].

$$\mathrm{Accuracy} = \frac{TP+TN}{TN+TP+FN+FP} \tag{12}$$

$$F1 - \mathrm{score} = \frac{2\times TP}{2\times TP+FP+FN} \tag{13}$$

$$\mathrm{Precision} = \frac{TP}{TP+FP} \tag{14}$$

$$\mathrm{Recall} = \frac{TP}{TP+FN} \tag{15}$$

Where, FP indicates false positive, TP denotes true positive, FN denotes false negative, and TN indicates true negative.

4.2. Performance analysis of the chickpea dataset

This section presents a quantitative analysis of the CNN-Average pooling approach in terms of F1-measure, recall, precision, and accuracy, using the chickpea and Fusarium-22 datasets, as displayed in Tables 1, 2, 3, and 4. Table 1 presents the quantitative analysis of various classifiers, without the use of PCA on the Caltech 256 dataset. The implemented CNN-Average pooling model is compared with the state-of-the-art models: Gated Recurrent Unit (GRU), CNN, and LSTM. It is evident from the analysis that when compared to the existing techniques: GRU, LSTM, and CNN, the CNN-Average pooling is more effective in chickpea disease classification.

Table 1 and Fig. 4 display the performance of the proposed approach on the chickpea dataset, without AlexNet. The performance metrics of GRU, LSTM, and CNN are evaluated and compared with the implemented CNN-Average pooling approach. The accomplished results prove that the proposed CNN-Average pooling model offers superior performance in terms of recall, F1-measure, precision, and accuracy with values of about 94.56%, 93.47%, 94.10%, and 95.90%.

Table 2 and Fig. 5 indicate the proposed approach's performance on the chickpea's dataset with AlexNet. The existing approaches of GRU, LSTM, and CNN are evaluated and compared with the implemented CNN-Average pooling approach. The accomplished outcomes display the proposed CNN-Average pooling approach's superiority, with the values of recall, F1-measure, precision, and accuracy respectively being 99.50%, 99.40%, 99.35%, and 99.60%. This proves that the proposed model outperforms existing classifiers. Table 3 presents the analysis of state-of-the-art models for the chickpea dataset in terms of accuracy and processing time.

As shown in Table 3, the proposed CNN-Average pooling model achieved a higher accuracy of 99.60% with a shorter processing time of 31 seconds, outperforming other models such as Vision Transformer, Shifter Window (SwIN) Transformer, ResNet-Transformer, and CNN-Transformer models.

Table 1: Different Classifiers without AlexNet for the Chickpea Dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
GRU	90.12	89.84	88.60	89.76
LSTM	85.43	91.57	90.95	89.93
CNN	92.76	90.95	89.90	91.29
CNN-Average pooling	95.90	94.10	94.56	93.47

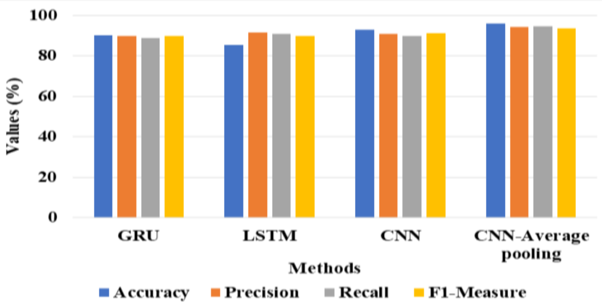


Fig. 4: Performance of Classification without AlexNet for the Chickpea Dataset.

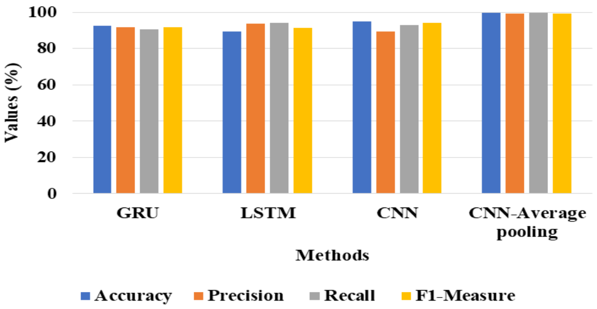


Fig. 5: Performance of Classification with AlexNet for the Chickpea Dataset.

Table 2: Different Classifiers with AlexNet for the Chickpea Dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
GRU	92.49	91.58	90.70	91.63
LSTM	89.47	93.60	94.25	91.37
CNN	94.82	89.40	92.80	93.99
CNN-Average pooling	99.60	99.35	99.50	99.40

Table 3: Analysis of State of the Art Models for Chickpea Dataset

Methods	Accuracy (%)	Processing Time (sec)
Vision Transformer	92.49	58
SwIN Transformer	94.37	53
ResNet-Transformer	96.66	44
CNN-Transformer	97.51	41
CNN-Average pooling	99.60	31

Table 4: Performance Analysis of Training Time and Resource Usage for Chickpea Dataset

Methods	Training Time (min)	Resource usage (mb)
GRU	3.52	13
LSTM	3.45	16
CNN	3.17	12
CNN-Average pooling	3.2	10

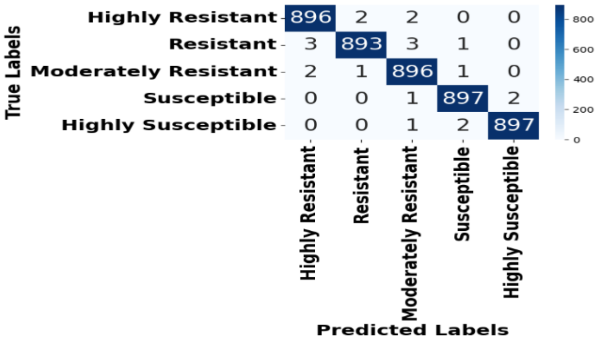
Table 4 displays the performance analysis of training time and resource usage for the chickpea dataset. As shown in Table 4, the proposed CNN-Average pooling model achieved a reduced training time of 3.2 minutes and resource usage of 10 MB, which is lower than that of the other compared models. Fig. 6 presents the confusion matrix for the chickpea dataset.

4.3. Performance analysis of fusarium-22 dataset

Table 5 and Fig. 7 display the proposed approach's performance on the Fusarium-22 dataset, without AlexNet. The performances of GRU, LSTM, and CNN are evaluated and compared with the implemented CNN-Average pooling model. The attained outcomes display that the implemented CNN-Average pooling approach attains better results in terms of recall, F1-measure, precision, and accuracy, with values of about 0.74, 0.83, 0.80, and 94.60% when compared to the other classifiers.

Table 6 and Fig. 8 represent the proposed model's performance on the Fusarium-22 dataset with AlexNet. The performance metrics of GRU, LSTM, and CNN are measured and compared with the implemented CNN-Average pooling approach. The obtained outcomes show that the implemented CNN-Average pooling model achieves improved outcomes in terms of recall, precision, F1-measure, and accuracy with values of 0.98, 0.99, 0.99, and 99.53%, respectively. This renders improved outcomes when compared to the other classifiers.

Table 7 presents the analysis of state-of-the-art models for the Fusarium-22 dataset in terms of accuracy and processing time. As shown in Table 7, the proposed CNN-Average pooling model achieved a higher accuracy of 99.53% with a shorter processing time of 33 seconds, outperforming other models such as Vision Transformer, SwIN Transformer, ResNet-Transformer, and CNN-Transformer. Similarly, Table 8 displays the performance analysis of training time and resource usage for the Fusarium-22 dataset. As shown in Table 8, the proposed CNN-Average pooling model achieved a reduced training time of 3.22 minutes and resource usage of 11 MB, which is lower than that of the other stated models. Fig. 9 presents the confusion matrix for the Fusarium-22 dataset.



True Labels	Highly Resistant	Resistant	Moderately Resistant	Susceptible	Highly Susceptible
Highly Resistant	896	2	2	0	0
Resistant	3	893	3	1	0
Moderately Resistant	2	1	896	1	0
Susceptible	0	0	1	897	2
Highly Susceptible	0	0	1	2	897

Fig. 6: Confusion Matrix for Chickpea Dataset.

Table 5: Different Classifiers without AlexNet for the Fusarium-22 Dataset

Methods	Accuracy (%)	Precision	Recall	F1-Measure
GRU	84.70	0.56	0.67	0.60
LSTM	90.81	0.73	0.62	0.75
CNN	92.87	0.79	0.70	0.78
CNN-Average pooling	94.60	0.80	0.74	0.83

Table 6: Different Classifiers with AlexNet for the Fusarium-22 Dataset

Methods	Accuracy (%)	Precision	Recall	F1-Measure
GRU	92.74	0.68	0.83	0.79
LSTM	90.93	0.55	0.76	0.68
CNN	94.50	0.72	0.69	0.84
CNN-Average pooling	99.53	0.99	0.98	0.99

Table 7: Analysis of State of the Art Models for Fusarium-22 Dataset

Methods	Accuracy (%)	Processing Time (sec)
Vision Transformer	93.48	54
SwIN Transformer	95.39	50
ResNet-Transformer	96.96	43
CNN-Transformer	98.28	38
CNN-Average pooling	99.53	33

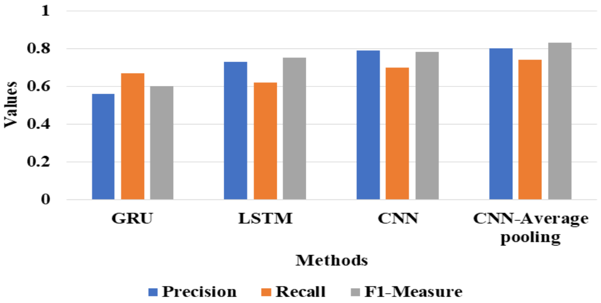


Fig. 7: Performance of Classification without AlexNet for the Fusarium-22 Dataset.

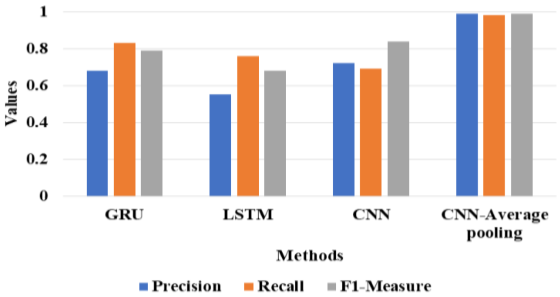


Fig. 8: Performance of Classification with AlexNet for the Fusarium-22 Dataset.

4.4. Ablation study

This section presents a series of ablation experiments to evaluate the contribution of each component in the proposed CNN-Average pooling model. Each component was removed one at a time, and the performance of the resulting model was measured. Table 9 below shows the results of the ablation study on pre-processing techniques for the chickpea and Fusarium-22 datasets.

4.5. Comparative analysis

The results of the proposed CNN-Average pooling model are analyzed in terms of performance metrics. Table 10 compares the conventional models with the proposed model, utilizing the chickpea dataset. It demonstrates that the proposed method outperforms previous models, including CNN-LSTM [16] and Hybrid GLCM-Color Histogram features [17], in terms of accuracy, recall, and precision. Specifically, the CNN-Average pooling method achieves superior results on the chickpea dataset, with accuracy, recall, and precision values of 99.35%, 99.60%, and 99.50%, respectively.

The performance of the implemented method is analyzed in Table 11. A comparative analysis of the existing and proposed models on the Fusarium-22 dataset is presented in Table 11. The table demonstrates that the proposed method outperforms all other methods, including GLRLM-HSV-based KNN [19] and DenseNet-201 CNN [20], in terms of accuracy, recall, and precision. Specifically, the CNN-Average pooling method achieves superior results on the Fusarium-22 dataset, with accuracy, recall, F1-measure, and precision values of 99.53%, 0.99, 0.99, and 0.98, respectively.

4.6. Discussion

The limitations of the standing methods and the qualities of the implemented approach are considered in this section. While the CNN-LSTM [16] struggles with identifying chickpea diseases, it requires a boost for more efficient segmentation for effective disease dealing. The GLRLM-HSV-based KNN [19] is limited by small dataset training, causing it to get stuck in local minima, which can cause issues like underfitting or overfitting. The pre-trained DenseNet-201 CNN [20] approach, though effective, is time-consuming and presents challenges during the classification process. The offered pooling approach is intended to address these issues. In this work, the Chickpea and Fusarium-22 datasets were employed to evaluate the performance of the proposed pooling approach.

Feature vectors were drawn from images using the AlexNet architecture for high-level labelling. The proposed approach enhances the system's ability to accurately categorize chickpea diseases. There are several advantages to using the propsoed approach for chickpea disease classification. The average pooling process reduces overfitting and decreases the feature map size while maintaining critical data.

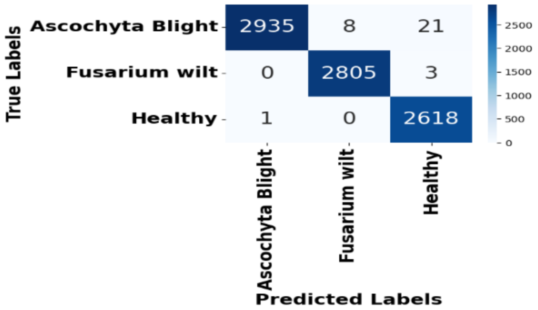


Fig. 9: Confusion Matrix for Fusarium-22 Dataset.

Table 8: Performance Analysis of Training Time and Resource Usage for Fusarium-22 Dataset

Methods	Training Time (min)	Resource usage (mb)
GRU	3.59	12
LSTM	3.43	17
CNN	3.27	13
CNN-Average pooling	3.22	11

Table 9: Ablation Study of Classification Analysis

Methods	Accuracy (%)			
	Without image denoising and data augmentation		With image denoising and data augmentation	
	Chickpea	Fusarium-22	Chickpea	Fusarium-22
GRU	84.34	83.12	90.25	89.37
LSTM	85.18	85.49	91.63	90.86
CNN	85.71	86.97	92.18	91.57
CNN-Average pooling	88.44	88.08	97.12	97.41

Table 10: Implemented Approach Compared with Existing Methods

Author	Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)
Belay [16]	Chickpea dataset	CNN-LSTM	92.55	N/A	N/A
Abuhayi and Bezabih [17]	Chickpea dataset	Hybrid GLCM-Color Histogram features	95.49	93.73	94.01
Proposed	Chickpea dataset	CNN-Average pooling	99.60	99.35	99.50

Table 11: Implemented Approach Compared with Existing Methods

Author	Dataset	Methods	Accuracy (%)	Precision	Recall	F1-Measure
Hayit [19]	Fusarium-22 dataset	GLRLM-HSV-based KNN	93.2	0.94	0.94	0.95
Hayit [20]	Fusarium-22 dataset	DenseNet-201 CNN	90	N/A	N/A	N/A
Proposed	Fusarium-22 dataset	CNN-Average pooling	99.53	0.99	0.98	0.99

This arrangement is effective in particularly for detecting disease patterns in images of various sizes, allowing for robust finding of diverse chickpea diseases. The method accurately detects infection cases and is adaptable to a wide range of disease patterns. Finally, CNN-Average pooling is an efficient and effective methodology for chickpea disease classification, improving both performance and efficiency.

5. Conclusion

In this research work, a new method called CNN-Average pooling is proposed to address the chickpea disease taxonomy challenges. Initially, data from the Chickpea and Fusarium-22 datasets are collected and pre-processed using an image denoising technique. Next, data augmentation is applied to improve the dataset size, followed by the extraction of features using the AlexNet architecture to retrieve the most significant features from the augmented data. Finally, CNN-Average pooling is engaged in the classification process, improving the model's disease classification performance. The results demonstrate that the proposed CNN-Average pooling approach outperforms existing methods in terms of accuracy, recall, precision, and F1-measure. When compared to the state-of-the-art methods: CNN-LSTM [16], Hybrid GLCM-Color Histogram features [17], GLRLM-HSV-based KNN [18], and DenseNet-201 CNN [19], the proposed CNN-Average pooling achieves superior results. Specifically, it reaches the highest accuracy of 99.60% and 99.53% on the Chickpea and Fusarium-22 datasets, respectively. In the future, chickpea disease classification will be further analyzed using other deep learning classifiers to further enhance performance metrics.

Acknowledgement

I would like to thank my colleagues for their support in this research work.

Conflict of interest

The authors declare that there is no conflict of Interest.

Table: Notations

Notations	Definitions
t	Time
$u_t = r(x, y, t)$	Image during time t
x, y	Operators
div	Divergence operator
∇	Gradient operator
θ	Rotation angle
(x', y')	Resized image
(x, y)	Original image
S_y	Scaling factor through y-axis
S_x	Scaling factor through x-axis
w, x	Kernel function
$(x \times w)(a)$	Input of kernel function
I	Image
K	2D kernel
$W \in M_{m,n}$	Weighted matrix
b	Bias vector
m and n	Rows
R	Columns
f	Fully connected layer
FP	False Positive
TP	True Positive
FN	False Negative
TN	True Negative

References

[1] Y.A. Bezabih, B.M. Abuhayi, A.M. Ayalew, and H.A. Asegie, “Classification of Pumpkin Disease by using a Hybrid Approach”, *Smart Agricultural Technology*, Vol. 7, p. 100398, 2024. https://doi.org/10.1016/j.atech.2024.100398.

[2] R.S. Solanki, A. Babbar, and N. Tripathi, “Genetic diversity analysis in kabuli chickpea (Cicer arietinum L.) genotypes based on quantitative traits and molecular markers”, B*angladesh Journal of Botany*, Vol. 51, No. 3, pp. 581-587, 2022. https://doi.org/10.3329/bjb.v51i3.62005.

[3] A. Kirtis, M. Aasim, and R. Katırcı, “Application of artificial neural network and machine learning algorithms for modeling the in vitro regeneration of chickpea (Cicer arietinum L.)”, *Plant Cell, Tissue and Organ Culture (PCTOC),* Vol. 150, No. 1, pp. 141-152, 2022. https://doi.org/10.1007/s11240-022-02255-y.

[4] D. Saha, and A. Manickavasagan, “Chickpea varietal classification using deep convolutional neural networks with transfer learning”, *Journal of Food Process Engineering*, Vol. 45, No. 3, p. e13975, 2022. https://doi.org/10.1111/jfpe.13975.

[5] M.R. Anwar, D.J. Luckett, Y.S. Chauhan, R.H.L. Ip, L. Maphosa, M. Simpson, A. Warren, R. Raman, M.F. Richards, G. Pengilley, K. Hobson, and N. Graham, “Modelling the effects of cold temperature during the reproductive stage on the yield of chickpea (Cicer arietinum L.)”, *International Journal of Biometeorology*, Vol. 66, No. 1, pp. 111-125, 2022. https://doi.org/10.1007/s00484-021-02197-8.

[6] S. Addisu, C. Fininsa, Z. Bekeko, A. Mohammad, A. Kumar, and A. Fikre, “Distribution of Chickpea (Cicer arietinum L.) Ascochyta blight (Didymella rabiei) and analyses of factors affecting disease epidemics in Central Ethiopia”, *European Journal of Plant Pathology*, Vol. 166, No. 4, pp. 425-444, 2023. https://doi.org/10.1007/s10658-023-02672-5.

[7] S. Talasila, K. Rawal, and G. Sethi, “Black gram disease classification using a novel deep convolutional neural network”, *Multimedia Tools and Applications*, Vol. 82, pp. 44309-44333, 2023. https://doi.org/10.1007/s11042-023-15220-4.

[8] Y.M. Abd Algani, O.J.M. Caro, L.M.R. Bravo, C. Kaur, M.S. Al Ansari, and B.K. Bala, “Leaf disease identification and classification using optimized deep learning”, *Measurement: Sensors*, Vol. 25, p. 100643, 2023. https://doi.org/10.1016/j.measen.2022.100643.

[9] M.M. Islam, M.A.A. Adil, M.A. Talukder, M.K.U. Ahamed, M.A. Uddin, M.K. Hasan, S. Sharmin, M.M. Rahman, and S.K. Debnath, “DeepCrop: Deep learning-based crop disease prediction with web application”, *Journal of Agriculture and Food Research,* Vol. 14, p. 100764, 2023. https://doi.org/10.1016/j.jafr.2023.100764.

[10] A. Chug, A. Bhatia, A.P. Singh, and D. Singh, “A novel framework for image-based plant disease detection using hybrid deep learning approach”, *Soft Computing*, Vol. 27, No. 18, pp. 13613-13638, 2023. https://doi.org/10.1007/s00500-022-07177-7.

[11] A. Haridasan, J. Thomas, and E.D. Raj, “Deep learning system for paddy plant disease detection and classification”, *Environmental Monitoring and Assessment,* Vol. 195, No. 1, p. 120, 2023. https://doi.org/10.1007/s10661-022-10656-x.

[12] N. Upadhyay, and N. Gupta, “Diagnosis of fungi affected apple crop disease using improved ResNeXt deep learning model”, *Multimedia Tools and Applications,* Vol. 83, No. 24, pp. 64879-64898, 2024. https://doi.org/10.1007/s11042-023-18094-8.

[13] A. Ahmad, A. El Gamal, and D. Saraswat, “Toward Generalization of Deep Learning-Based Plant Disease Identification Under Controlled and Field Conditions”, *IEEE Access,* Vol. 11, pp. 9042-9057, 2023. https://doi.org/10.1109/ACCESS.2023.3240100.

[14] P. Kaur, S. Harnal, V. Gautam, M.P. Singh, and S.P. Singh, “A novel transfer deep learning method for detection and classification of plant leaf disease”, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 14, No. 9, pp. 12407-12424, 2023. https://doi.org/10.1007/s12652-022-04331-9.

[15] E. Moupojou, A. Tagne, F. Retraint, A. Tadonkemwa, D. Wilfried, H. Tapamo, and M. Nkenlifack, “FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning”, *IEEE Access*, Vol. 11, pp. 35398-35410, 2023. https://doi.org/10.1109/ACCESS.2023.3263042.

[16] A.J. Belay, A.O. Salau, M. Ashagrie, and M.B. Haile, “Development of a chickpea disease detection and classification model using deep learning”, *Informatics in Medicine Unlocked*, Vol. 31, p. 100970, 2022. https://doi.org/10.1016/j.imu.2022.100970.

[17] B.M. Abuhayi, and Y.A. Bezabih, “Chickpea disease classification using hybrid method”, *Smart Agricultural Technology*, Vol. 6, p.100371, 2023. https://doi.org/10.1016/j.atech.2023.100371.

[18] N. Çetin, H. Ozaktan, S. Uzun, O. Uzun, and C.Y. Ciftci, “Machine learning based mass prediction and discrimination of chickpea (Cicer arietinum L.) cultivars”, *Euphytica*, Vol. 219, No. 1, p. 20, 2023. https://doi.org/10.1007/s10681-022-03150-5.

[19] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, Feb. 2018. https://doi.org/10.1016/j.compag.2018.02.016.

[20] B. Keerthana, J. Vamsinath, C. S. Kumari, S. V. S. Appaji, P. P. Rani, and S. Chilukuri, “Machine learning techniques for medicinal leaf prediction and disease identification,” International Journal of Experimental Research and Review, vol. 42, pp. 320–327, Aug. 2024, https://doi.org/10.52756/ijerr.2024.v42.028.

[21] T. Hayit, A. Endes, and F. Hayit, “KNN-based approach for the classification of fusarium wilt disease in chickpea based on color and texture features”, *European Journal of Plant Pathology*, Vol. 168, No. 4, pp. 665-681, 2024. https://doi.org/10.1007/s10658-023-02791-z.

[22] T. Hayit, A. Endes, and F. Hayit, “The severity level classification of Fusarium wilt of chickpea by pre-trained deep learning models”, *Journal of Plant Pathology*, Vol. 106, No. 1, pp. 93-105, 2023. https://doi.org/10.1007/s42161-023-01520-z.

[23] P. K. Sekharamantry, M. S. Rao, Y. Srinivas, and A. Uriti, “PSR-LeafNet: a deep learning framework for identifying medicinal plant leaves using support vector machines,” *Big Data and Cognitive Computing*, vol. 8, no. 12, p. 176, Dec. 2024, https://doi.org/10.3390/bdcc8120176.