Recognition of Diseases of Maize crop using Deep Learning Models

Md. Ashraful Haque[1], Sudeep Marwaha[*1], Chandan Kumar Deb[1], Sapna Nigam[1] and Alka Arora[1]

ICAR-Indian Agricultural Statistics Research Institute, New Delhi-110012

[*]**Corresponding Author:**
Sudeep Marwaha
sudeep@icar.gov.in

Abstract:

Disease attack on crops is one of the most serious threats to the global food supply chain. A proper, comprehensive and systematic solution is required for the early recognition of diseases and to reduce the overall crop loss. In this regard, deep learning techniques (especially convolutional neural networks (CNNs/ConvNets)) are being successfully applied for automatically recognizing the diseases of crops using digital images. This study proposes a novel 15-layer deep convolutional neural network (CNN) model for recognizing the diseases of maize crop. Around 3,852 images of maize crop were collected from the PlantVillage data-repository. This dataset contains leaf images of three diseases viz. Grey Leaf Spot (GLS), Common Rust (CR) and Northern Corn Leaf Blight (NCLB) as well as the healthy ones. The proposed model showed significant results for recognizing the unseen diseased images of the maize crop. We also employed a few popular pre-trained networks in the transfer learning approach for training on the maize dataset. We presented the comparative performance analysis between the proposed model and the pre-trained models in the result section of the manuscript. The experimental findings reported that our proposed model showed 3.2% higher prediction performance with 3x lesser trainable parameters than the best-performing pre-trained network (i.e. DenseNet121). The overall performance analysis reported that the proposed CNN model is very effective in identifying the images of maize diseases and also performs quite better than the popular pre-trained models.

Keywords: Deep Learning, Convolutional Neural Networks, Disease recognition, Maize crop

1 Introduction:

By 2050, the global population is expected to rise by almost 2 billion people from 7.7 billion to 9.7 billion [1]. Therefore, sustainable food security is needed to be maintained to meet the requirements of such a large population. Globally, maize is considered one of the major cereal crops after rice and wheat [2]. It potentially supports the overall food supply chain by providing food as well as feed for humans and livestock across the world. During the year 2020, maize was grown in around ~197 mha area across 170 countries in the world and yielded at the rate of 5.82 t/ha [3]. Spite of the high productivity, maize crop is highly vulnerable to a variety of diseases during its whole growing season. Till date, around 112 diseases have been reported from different regions of the world in maize crop [4]. These diseases have the potential of causing moderate to severe damage to the overall production of maize crop. According to the reports, every year around 4-14% of the total production of maize is damaged due to the attack of disease-causing pathogens alone [5]. In order to address this issue, diseases should be properly diagnosed/recognized before applying any management practices to the crop. Conventionally, diseases are diagnosed either through the visual inspection of the damage in fields or by performing laboratory experiments on the damaged plant parts of the crop by domain experts [6]. However, these traditional approaches have few inherent limitations and aren't always feasible. These approaches require highly qualified staff and involve a significant amount of time to complete the desired tasks. Hence, an effective and precise disease diagnosis approach is the hour of need in the current scenario.

In recent years, computer vision field of computer science has been blessed with the tremendous success of deep learning-based techniques, especially convolutional neural networks (CNNs/ConvNets). The CNNs/ConvNets have introduced automation in the field of image recognition [7]. The CNNs are capable of extracting the inherent and significant features from a large number of images and classifying the extracted features into their respective classes. In the last couple of years, recognition of images using deep learning techniques has also gained huge popularity in the agriculture sector [8]. In this respect, disease recognition in crops using the symptomatic images of diseases has emerged as a suitable candidate for this area of computer science. Therefore, the CNNs/ConvNets are considered as the state-of-the-art framework for the automated diagnosis of diseases of crops using digital images [9]. In the present study, a developed a novel deep CNN model to recognize/identify the diseases of maize crop from the PlantVillage data-repository [10] (publicly available image data repository of diseases of several crops). The major contribution of this study is mentioned as follows:

- We proposed a 15-layer deep convolutional neural network model that is simple and lightweight in nature. The proposed model was trained and tested on the diseased images of maize crop from the PlantVillage data repository.
- In order to showcase the effectiveness of our proposed model, we applied the transfer learning approach using a few popular pre-trained models on the maize dataset and compared their classification performance with our proposed model.

The remainder of the article is arranged as follows: In the next section, we briefly discussed the related works done on maize crop. Next, we discussed the proposed approach for recognition of the diseases of maize crop using the convolutional neural networks. After that, we provided experimental details of model implementation, details of data collection and pre-processing and applied evaluation metrics. Then, we presented and discussed the experimental results, prediction performance proposed model and comparative performance analysis between several pre-trained models done using the maize dataset. Finally, we summarised and concluded the overall study in the last section.

2 Related works

Several researchers across the world are performing experiments for recognizing the digital image of diseases of maize crops by applying deep learning techniques. Mohanty et al. [11] worked on the diagnosis of 26 diseases of 14 different crops using deep learning techniques. They used the publicly available image data-repository 'PlantVillage'[10] which contains more than 50 thousand images of several crops such as apple, cherry, maize, potato, tomato etc. They applied transfer learning on two most popular deep CNN models viz. AlexNet [12] and GoogleNet [13] for their experiment. Dechant et al. [14] trained a computational pipeline of CNNs for the identification of northern corn leaf blight (NLB) disease of maize. They trained and validated the proposed CNN model on the in-field images of NLB disease captured in non-destructive mode. Their proposed 3-stage CNN model identifies the NLB disease symptoms with 96.7% accuracy. However, they focused on only one identifying only one disease of maize and the use of 3-stages of CNN makes it slightly expensive in terms of computation. Zhang et al. [15] applied two state-of-the-art models viz. GoogleNet [13] and Cifar-10 for recognizing 9 diseases of maize crop. They collected the image dataset from the PlantVillage data repository and several internet sources. They applied the transfer learning approach to reduce the computational overload of the state-of-the art models and achieved more than 98% accuracy. Sibiya et al. [16] proposed a custom CNN model to identify and classify images of diseases of maize crop. They collected the symptomatic images of maize crop from different agricultural fields. They used a java based implementation method for implementing the proposed CNN model. However, their proposed model achieved an overall classification of 92.85% on the test dataset. Priyadharshini et al. [17] developed a simple CNN model inspired by LeNet [18] architecture for classifying diseases of maize crop collected from the PlantVillage dataset. They applied the PCA whitening technique to pre-process the images before training and achieved around 98% accuracy. Lv et al. [19] proposed the improved version of the AlexNet [12] model for classifying images (diseases, insect-pest and nutrient deficiencies) of maize crop. The enhanced feature learning capability of AlexNet model by incorporating one multi-scale convolution operation provided significant results on test images. Waheed et al. [20] proposed an optimized CNN model for classifying the images of maize crop into respective disease categories. Their proposed model is based on the architectural framework of the DenseNet [21] model. Haque et al. [9] have used the popular deep CNN network 'GoogleNet' for identifying the MLB disease of maize crop. They collected images of healthy and MLB diseased leaves of maize from agricultural fields in a non-destructive manner with complex backgrounds. They achieved around 99% accuracy while identifying the MLB disease of maize. Chen et al. [22] developed a lightweight CNN model for the identification of 8 diseases of maize crop. They collected 466 images of maize diseases from agricultural fields of Fujian Province, China. They incorporated an attention module in the DenseNet architecture to propose the novel CNN model, which achieved around 95% classification accuracy.

The studies in the available literature reported that most of the works had shown significant results in identifying the diseases of crops. However, in some cases, the significant results were achieved at the cost of higher number of trainable parameters or computation time which adversely affects the model's overall detection performance. Therefore, to reduce the computational complexity of the conventional disease identification model and improve the model's detection accuracy, we proposed a computationally lightweight CNN model for recognizing the diseases of maize crop. We applied 15 layers of stacked convolution and pooling operations to extract the most inherent features from the images. In addition, we incorporated batch normalization and ReLU operation in each of the convolutional operations of our proposed network to normalize the input feature maps for efficient feature extraction and reduce the computational parameters in the network. We also employed global average pooling instead of fully connected layers, drastically reducing the number of parameters with enhanced detection capability.

3 Proposed approach for maize disease recognition:

3.1 Overview

We depicted the overall flow of the proposed maize disease recognition approach using the convolutional neural networks (ConvNets/CNNs) in Fig. 1. Firstly, we collected the digital images of maize crop from the PlantVillage data-repository and stored them in our storage disks. Next, we applied different image pre-processing and augmentation techniques to this maize image dataset. Then, we split the maize image dataset into two groups such as training and testing sets. After that, we trained our proposed CNN model as well as the pre-trained models by using the training set of the maize image dataset. After the completion of model training, all the models were evaluated based on their prediction performance on the test set. Finally, the best model was selected for the maize disease recognition approach.

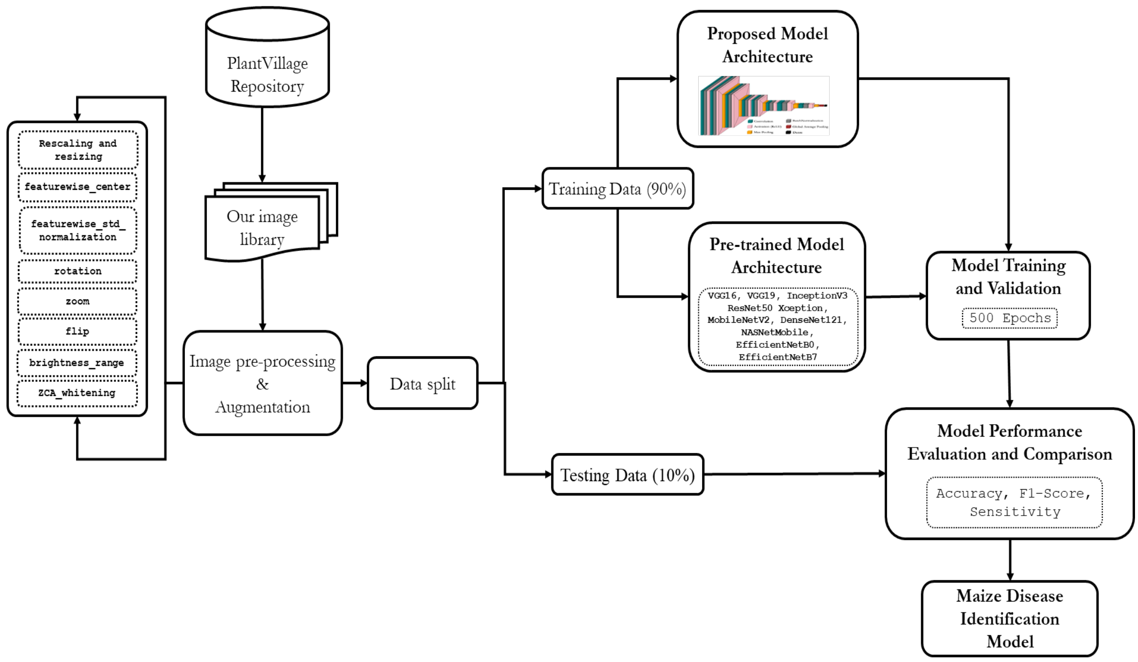

Fig. 1 Overall flow of the maize disease recognition approach

3.2 Convolutional Neural Networks

Generally, convolutional neural networks (CNNs/ConvNets) are the highly specialized form of supervised artificial neural networks [17]. The concept of CNNs/ Convnets is inspired by the working principle of the biological visual cortex [26]. As shown in Fig. 2 (a, b), CNNs are composed of two major functional modules, such as the 'feature extraction module' and the 'classifier module' [27]. The feature extraction module is made up of multiple layers of linear transformations, i.e. convolution and pooling layer, which automates the extraction of the promising features from the images. The classifier module (also known as fully-connected layer or dense layer) then processes the extracted features to classify them into their respective classes. The details of the different layers of CNNs/Convnets are described in the following sections.

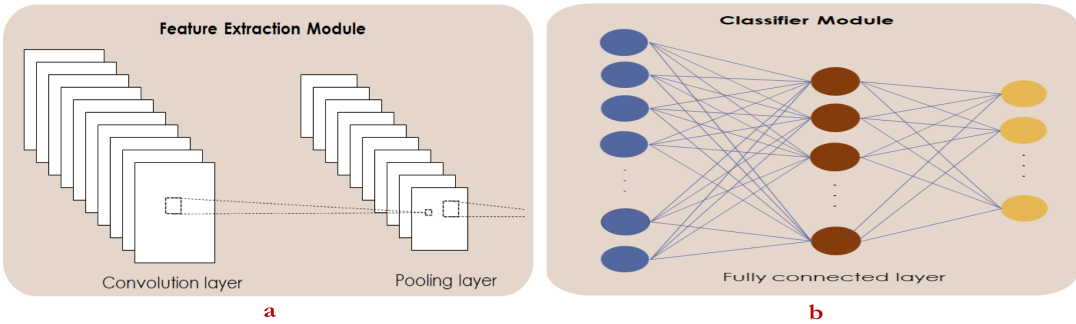

Fig. 2 General framework of convolutional neural networks (CNNs/Convnets) a) feature extraction module and b) classifier module

3.2.1 Convolution

In CNNs, the most significant and essential layer is the convolution layer. The CNNs learn the elementary low-level features of the images such as edges, curves, lines, endpoints, textures, corners and so on through the convolution operation [18]. The convolution layer involves the convolution operation, a mathematical operation (denoted by $*$) on two functions whose output determines how the shape of one function will change by the other one. In the case of image classification, the convolution layer convolves a set of filters (or kernels) through the pixels of the images and generates a set of features (or activation) maps [28]. The convolution operation performs sums of products between the filter (kernel) elements and the input activation maps. These sums of products of the feature maps are activated by the given non-linear activation functions applied in the networks [7]. The major advantages of the convolution operation are such as it provides sparse connectivity between the neurons, shares the parameters among several neurons and imposes translation invariance in the images. The overall convolution operation can be expressed by the equation shown in eq. 1:

$$z_k^l = \mathrm{g}(\sum_k x_k^l \,.\, W_k^l + b_k^l) \qquad \text{eq. 1}$$

Where, z_k^l denotes the output feature map of k-th input at l-th layer of the model; x_k^l denotes the k-th input feature map at l-th layer of the model; W_k^l and b_k^l denotes the weights and bias at the l-th layer of the model and $g(.)$ is the non-linear activation function.

3.2.2 Pooling

The pooling operation is the subsampling process in which input feature/activation maps are down-sampled i.e. the dimension of the inputs are reduced spatially. Pooling makes the representations in the input feature maps smaller but more manageable [26]. The output units of the pooling layer are connected to the outputs of a smaller number of neurons in the preceding layer of the network located within a narrower receptive field. It works individually on each of the input feature maps to generate a set of new pooled feature maps. Moreover, the down-sampling operation on the input feature maps in the pooling layer minimizes the number of trainable parameters and the network's computational load [29]. In the case of image classification using CNNs, mainly two types of pooling operation are applied viz. max-pooling and average pooling. In max-pooling operation, a max-filter (maxkernel) is applied over the input feature maps (see in Fig. 3) and in the average-pooling, an average-filter is applied on the feature maps [9].

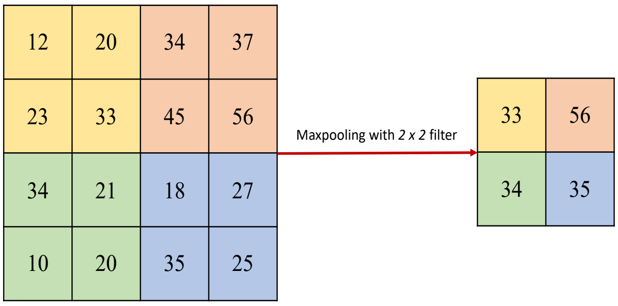

Fig. 3 Depiction of max-pooling operation on the feature maps in the CNNs/Convnets

3.2.3 Fully-connected (Dense) Layer

The fully-connected (dense) layer is generally a classifier in CNNs/ ConvNets that classifies the learned feature maps into corresponding class labels of the dataset under study. This layer generally resembles the traditional artificial neural networks (or MLPs) structurally as well as functionally as shown in Fig. 2(b). In the dense layer, all the neurons of any layer are associated with every neuron of other layers in the network precisely similar to the MLPs [9]. Hence, the outputs nodes are computed through the dot multiplications between the inputs vectors and the weight vectors followed by the bias term [17]. Finally, the dense layer's last layer has as many nodes as there are class labels in the dataset under study.

3.2.4 Activation Functions

The activation functions boost the efficiency of the CNN models significantly. These functions are applied to the output of the convolution operation before passing them to the next layer [30]. The activation function computes the weighted sum of the neurons to determine whether a neuron in the network should be activated or not. In case of CNNs/ConvNets, ReLU (Rectifier Linear Unit) function is used to activate the neurons in the convolution layer. ReLU is a non-linear transformation function that speed-up the network’s convergence capability during training [19]. It activates a specific neuron in the network when its value exceeds a certain threshold (e.g. 0 threshold value). The ReLU function is defined by the following equation (eq. 2):

$$ReLU(x_k) = \begin{cases} x_k, & \text{if, } x_k > 0 \\ 0, & \text{if, } x_k \leq 0 \end{cases} \qquad \text{eq. 2}$$

3.3 Proposed Convolutional Neural Network Model:

In this research work, we developed a novel deep convolutional neural network (deep CNN) model to recognize the diseased images of maize crop. The proposed deep CNN model is a 15-layer deep network consisting of 8 convolution layers, 5 max-pooling layers, 1 global average pooling layer and 1 dense layer as shown in Fig. 4. Each convolution layer was coupled with batch normalization (BN) layer and a ReLU activation function. A global average pooling (GAP) module was incorporated between the last max-pooling layer and the dense layer. Finally, a softmax function was incorporated into the dense layer to classify the images into the respective classes along with their probability values.

The proposed CNN model takes the images as input with the dimension of $227 \times 227 \times 3$ and outputs a vector of size 1×4 representing the probability values of 4 classes of the maize dataset. Initially, the input images were processed by convolution and pooling layers to generate a series of feature maps (feature detectors) which were the close representations of input images. In each convolution layer, kernels/filters of size 3 × 3 with stride 1 were used to generate the feature maps. Straightway, the generated feature maps were normalized by the BN layer and passed through the ReLU activation function. The BN was applied to normalize the input batch of images with mean 0 and standard deviations 1 and the ReLU function added nonlinearity into the feature maps. In the maxpool layers, we applied 3 × 3 filters with stride 2, which drastically reduces the spatial dimension of the feature maps without losing any crucial information from the images. In this way, 8 convolutions (*cbr_1 to cbr_8*) and 5 maxpool *(pool_1 to pool_5*) operations were stacked together for the extraction of inherent features from the input images. Next, to process the generated inherent feature maps from the *pool_5* layer and to create the feature vectors, a GAP layer was applied instead of a fully connected layer. The GAP layer has the potential to enhance the prediction capability of the model without any extra training parameters [31]. At last, the GAP layer-generated feature vectors were processed by the softmax function in dense layer and output the probability values of respective classes. A layer-wise detailed description of the proposed CNN network is presented in Table 1. Our proposed CNN model consists of average 2.5 million parameters which are trained during model training.

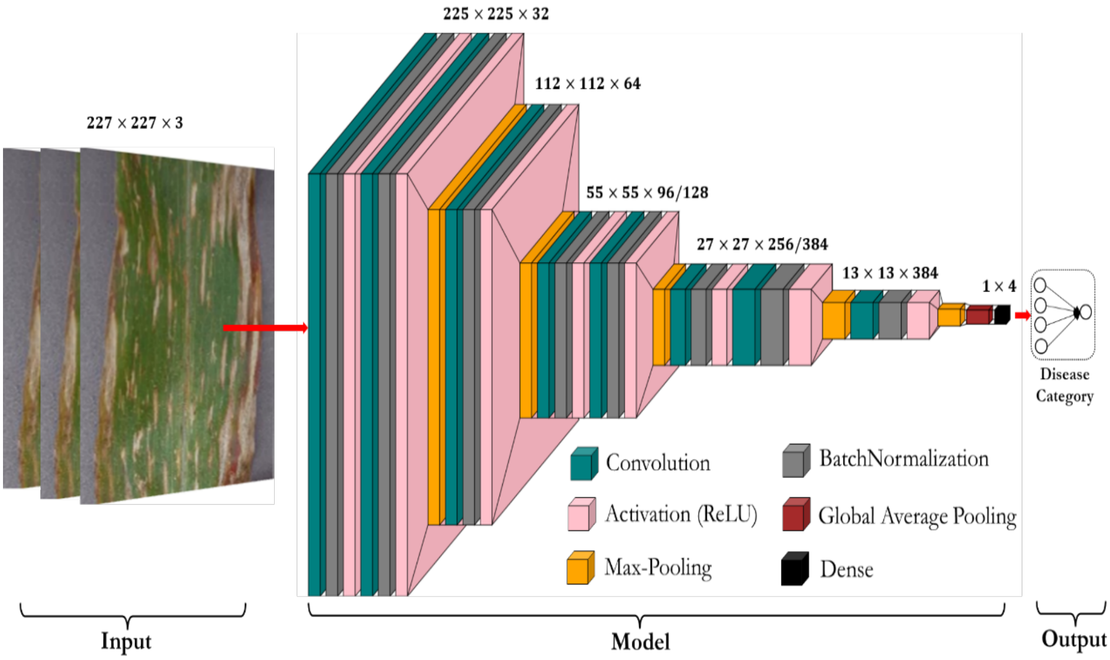

Fig. 4 Architectural view of the proposed deep CNN model for maize disease recognition

Table 1: Layer-wise configuration of the proposed model

Name	Layer	Kernel size	Stride	Output shape	Parameters
input	Input	-	-	227 x 227 x 3	0
cbr_1	Conv + BN + ReLU	3 x 3	1	225 x 225 x32	864 + 96 + 0
cbr_2	Conv + BN + ReLU	3 x 3	1	225 x 225 x32	9,216 + 96 + 0
pool_1	Max-Pooling	3 x 3	2	112 x 112 x 32	0
cnr_3	Conv + BN + ReLU	3 x 3	1	112 x 112 x 64	2,048 + 192 + 0
pool_2	Max-Pooling	3 x 3	2	55 x 55 x 64	0
cbr_4	Conv + BN + ReLU	3 x 3	1	55 x 55 x 96	55,296 + 288 + 0

cbr_5	Conv + BN + ReLU	3 x 3	1	55 x 55 x 128	12,288 + 384 +0
pool_3	Max-Pooling	3 x 3	2	27 x 27 x 128	0
cbr_6	Conv + BN + ReLU	3 x 3	1	27 x 27 x 256	2,94,912 + 768 + 0
cbr_7	Conv + BN + ReLU	3 x 3	1	27 x 27 x 384	8,84,736 + 1152 + 0
pool_4	Max-Pooling	3 x 3	2	13 x 13 x 384	0
cbr_8	Conv + BN + ReLU	3 x 3	1	13 x 13 x 384	13,27,104 + 1152 + 0
pool_5	Max-Pooling	3 x 3	2	6 x 6 x 384	0
gap	Global average pooling	-	-	384	0
dense	Dense	-	-	1 x 4	1540

3.4 Transfer learning approach:

The transfer learning approach in deep learning refers to the process of applying previously learned knowledge to a new task. According to this concept, machine learning or deep learning models are trained and tested on a bulky image dataset and their learned weights are applied to train and make inferences on some other image dataset. The main advantage of the transfer learning approach is that it significantly reduces the longer training time of the models [32]. Therefore, in this work several state-of-the-art deep CNN models were used, those were trained on the ImageNet dataset [33] and their learned pre-trained weights are available in the TensorFlow environment. We employed 10 popular state-of-the-art deep learning models viz. VGG16 [34], VGG19 [34], InceptionV3 [35], ResNet50 [36], Xception[37], MobileNetV2 [38], DenseNet121[21], NASNetMobile [39], EfficientNetB0 [40] and EfficientNetB7 [40] to train and classify our maize dataset in transfer learning approach.

4. Experimental Setup:

4.1 Implementation and hyperparameter setting:

The proposed model was implemented by python programming language using the TensorFlow environment. Tensorflow is an open-source environment for deep learning tasks, provided by Google [41]. The experiments in this work were conducted on NVIDIA DGX GPU servers equipped with 512GB RAM and 8 high-speed Tesla V100 graphics processing units of 32GB each. The proposed CNN model was compiled with the ‘Adam’ optimization function and ‘categorical cross entropy’ loss function. During the training phase of the model the default learning rate of the ‘Adam’ optimization function i.e. 0.001 was used with the fixed learning rate policy. The details of the hyperparameters of the proposed CNN model are presented in Table 2**.**

Table 2: Hyperparameters of the proposed model

Name	**Hyper Parameters**
Optimization Algorithm	Adam
Loss Function	Categorical Cross Entropy
Base Learning Rate	0.001
Momentum	0.9
Weight Decay	0.004
Epochs	500
Batch size	32

4.2 Data acquisition and pre-processing:

In this work, we collected around 3,852 digital images of diseases of maize crop from the PlantVillage data repository [10]. The images were downloaded from https://github.com/spMohanty/PlantVillage-Dataset. The images in the PlantVillage data repository were captured under controlled conditions i.e. captured with uniform background. This data-repository contains images of three diseases of maize crop viz. Grey Leaf Spot (GLS), Common Rust (CR), and Northern Corn Leaf Blight (NCLB) along with healthy ones. Example images of disease symptoms for each class are presented in Fig. 5. These are very serious foliar diseases occurring in maize crop and significant concerning maize productivity. A detailed description of the diseases and their symptomatic characteristics and a summary of the image dataset of the maize crop is provided in Table 3.

Table 3: Details of the maize dataset collected from the PlantVillage data-repository

Category	**Causal organism**	**Symptoms**	**# of Images**
Grey Leaf Spot (GLS)	*Cercospora zeae-maydis*	Individual rectangular or circular, brown to tan color necrotic lesions, parallel to the leaf veins [23]	513
Common Rust (CR)	*Puccinia sorghii Schw*	Oval to elongated rust-colored to cinnamon brown colored pustules sparsely scattered on both leaf surfaces [24]	1,192
Northern Leaf Blight (NCLB)	*Exserohilum turcicum (Pass) Leon.& Sugs.*	Symptoms start as small elliptical spots on the lower leaves which turn greenish cigar-shaped and bigger with time [25]	985
Healthy	-	-	1,162
		Total	**3,852**

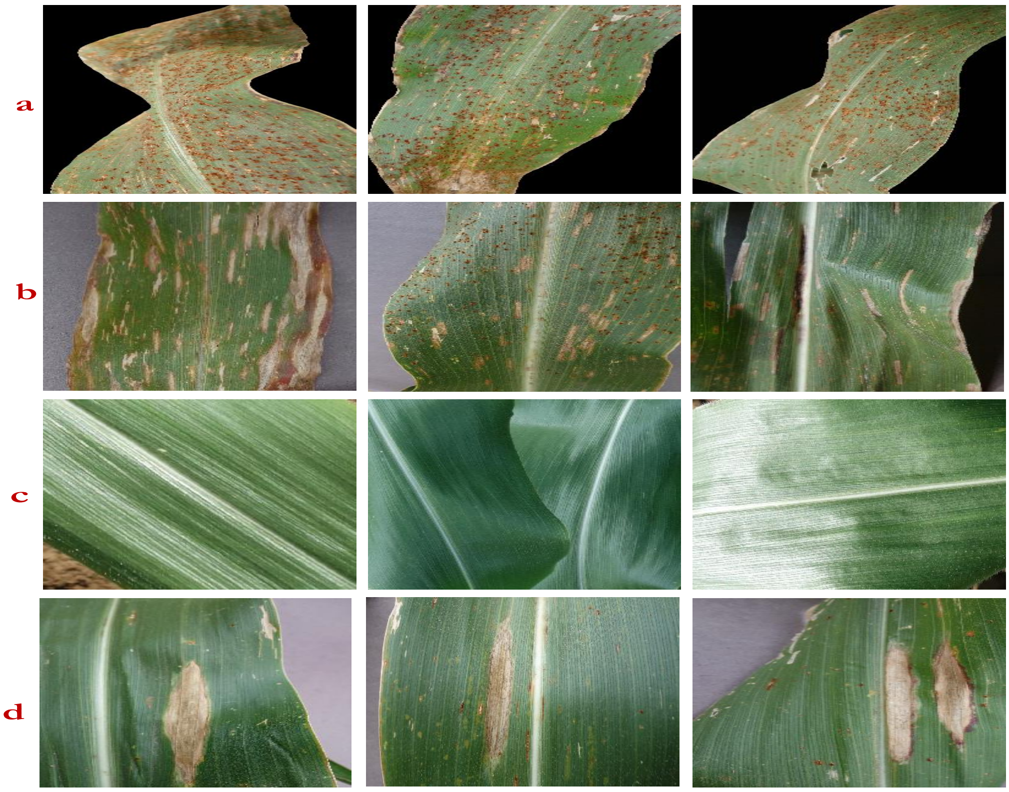

Fig. 5 Example images of diseases of maize crop: a) Common Rust (CR), b) Grey Leaf Spot (GLS), c) Healthy and d) Northern Corn Leaf Blight (NCLB)

The images were annotated by putting them in respective disease-named folders in the storage disks. All the images were resized to 227×227 pixel size for better interpretation by the proposed model. We used the *ImageDataGenerator* package in Keras to create the data tensors for each batch of images. To reduce the chance of the model's overfitting, we implemented a runtime mode of image augmentation technique provided by the *ImageDataGenerator* package. In this mode of augmentation, each batch of images was augmented during the model training time. The main advantage of this technique is that the images were augmented at the model runtime only, hence no extra disk storage was required for storing the augmented images. A detailed description of the image augmentation techniques is given in Table 4.

Table 4: Description of the applied image augmentation

Techniques	Details
featurewise_center	mean of the dataset is set to 0
featurewise_std_normalization	inputs are divided by the standard deviation of the dataset
rotation	rotating within the range of 20 degrees
shear	Shearing within the range of 0.2
zoom	Zooming within the range of 0.2
flip	flip the images horizontally and vertically
brightness_range	increase brightness within the range of [1.5, 2.0]
ZCA_whitening	ZCA whitening is applied with the default epsilon value

4.3 Performance Evaluation and comparison metrics:

In this work, confusion matrix and other associated metrics viz. classification accuracy, specificity, recall (sensitivity), Precision, F1-Score and Matthews Correlation Coefficients (MCC) were used to evaluate the prediction performance of the proposed model. The confusion matrix is the tabular representation of the model's prediction performance on the testing dataset. The rows entries of the confusion matrix denote the instance of the actual classes, whereas the columns represent the model's prediction instances. The correct predictions of the proposed models concerning the actual class labels are denoted by the diagonal elements, while the incorrect predictions are denoted by the off-diagonal elements of the confusion matrix. It provides four variables such as true negatives (TN), false positives (FP), false negatives (FN) and true positives (TP). These variables were used to compute the evaluation metrics as follows:

$$\text{Classification Acuracy} = \frac{(\text{True Positives (TP)}+\text{True Negeatives (TN)})}{(\text{Number of samples in the dataset})}$$

$$\text{Specificity} = \frac{(\text{True Negatives (TN)})}{(\text{False Positives (FP)}+\text{True Negatives (TN)})}$$

$$\text{Recall (Sensitivity)} = \frac{(\text{True Positives (TP)})}{(\text{ False Negeatives(FN)+ True Postives (TP)})}$$

$$\text{Precision} = \frac{(\text{True Positives (TP)})}{(\text{False positives (FP) + True Positives (TP)})}$$

$$\text{F1} - \text{Score} = 2 * \frac{(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})}$$

$$\text{Mathews Correlation Coefficient (MCC)} = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

5 Results and Discussion:

5.1 Prediction performance of proposed model:

The whole image dataset of the maize crop was partitioned into two distinct sets such as training and testing datasets in 90:10 ratio. There were 3464 images in the training set (GLS - 461; CS - 1,072; NCLB - 886 and Healthy - 1045) and the testing set contains 388 images (GLS - 52; CS - 120; NCLB - 99 and Healthy - 117) respectively. The proposed deep CNN model was trained and validated for 500 epochs with a batch size of 32 images per epoch. The training vs validation accuracy and training vs validation loss of the proposed model over the epochs are shown in Fig. 6 (a,b). Here, the trends of the accuracy curve in both training and validation sets reported a significant surge with the increments of training epochs. On the other side, loss values for both training and validation sets were gradually decreased as training iterations increased. The experimental findings on the testing set by the proposed deep CNN model reported a classification accuracy of 99.1% with the loss of 0.077 and an F1-score of 97.49%. The results implied that the proposed model performed admirably on the testing set. The reported results are far better than the random guessing of the images for recognition. The higher testing accuracy, as well as higher F1-score, implies the efficiency and efficacy of the proposed model in the recognition of images of maize diseases.

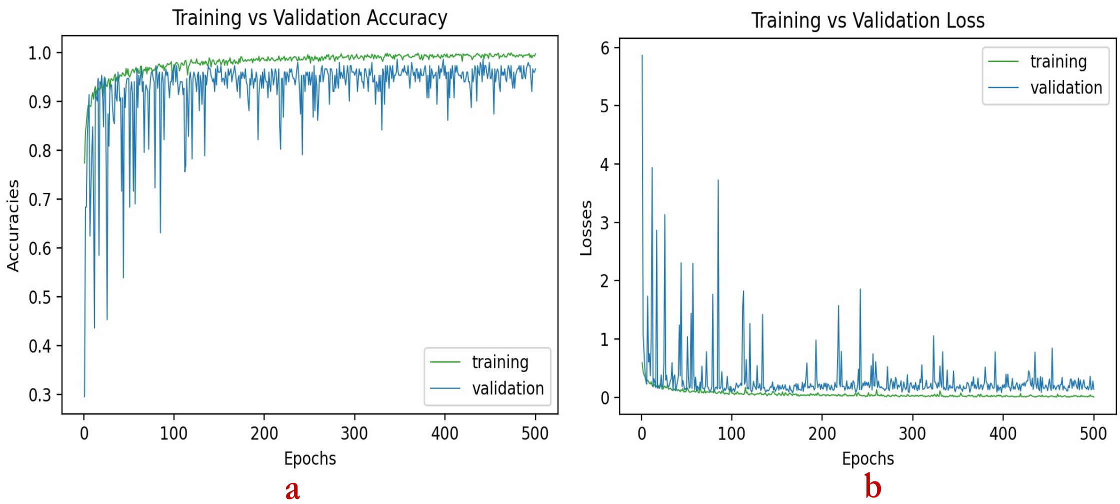

Fig. 6 Performance of the proposed model: a) trends of training and validation accuracy and b) trends of training and validation loss

We computed the confusion matrix and relevant evaluation matrices of the proposed model on the testing set and presented their results in Fig. 7 and Table 5. It is quite evident from Fig. 7 that the proposed model's prediction performance was extraordinary in case of Healthy and CR as all the test images of these two classes were predicted correctly by the model. The model showed a slightly moderate performance for the images of NCLB class as 2 NCLB images were predicted as GLS. Lastly, the model showed a little poor performance for the GLS class as compared to NCLB class where 5 GLS images were predicted as NCLB and 1 as healthy. As a whole, out of 388 images, our model predicted 380 images correctly and 8 images were wrongly predicted which is quite a significant prediction result shown by the proposed model.

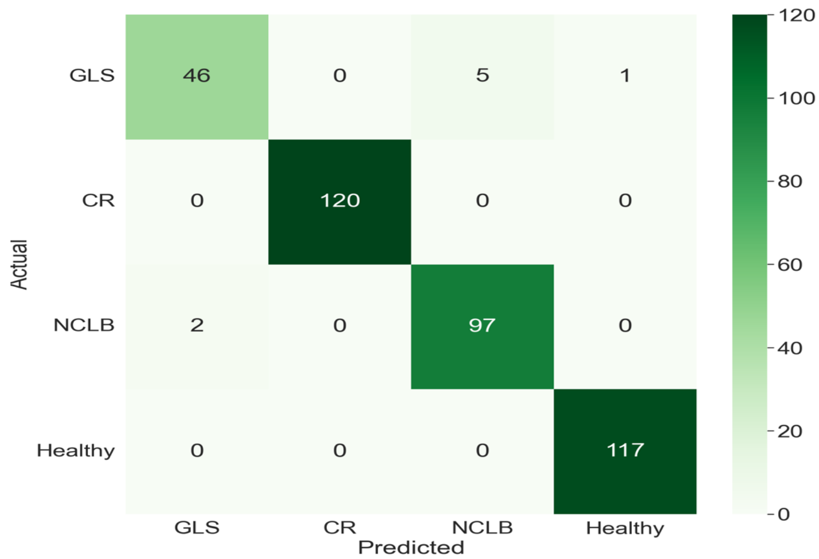

Fig. 7 Confusion matrix of the proposed CNN model on the testing set

From Table 5**,** we can get class-wise insights of the model's prediction performance on the testing set. The classification accuracy and specificity achieved for all the classes were more than 98% which implies quite good prediction results for the model. However, the GLS class has comparatively less sensitivity due to higher false negatives and NCLB class has comparatively less precision as there were more false positives. However, the class-wise analysis suffers from the imbalanced class problem, that's why the previously discussed metrics (i.e. classification accuracy, specificity, sensitivity and precision) aren't sufficient to assess the model's performance. Therefore, we also obtained the class-wise F1-scores of the proposed model on the testing set as presented in Table 5. If we see the class-wise F1-scores of the model, it can be observed that CR and Healthy classes achieved more than 99% of the F1-score, but the GLS and NCLB have lower values. According to the F1-score, the model is 93.88% accurate for identifying the GLS images from the non-GLS images and 96.52% accurate for identifying the NCLB images from the non-NCLB ones. However, the proposed CNN model obtained an average F1-score of 97.49%, which supports the significance of the model in recognizing the unknown images of maize crop.

Next, we calculated MCC to obtain the correlation between the predicted and actual labels of each class of the dataset (as shown in Table 5). The results showed that the CR, Healthy and NCLB classes are having correlation values of more than 0. 95 between the actual labels and the predicted labels. However, the GLS showed comparatively lesser MCC values. Here, it can be said that the model could identify the non-GLS images for the given non-GLS images (i.e. CR, Healthy and NCLB) with high probability but significantly less efficient at recognizing the GLS images for given GLS images. The possible reason for this result may be the lesser number of images in GLS class as compared to the other classes. This can be overcome by incorporating more images or generating artificial images for the GLS class.

Table 5: Class-wise prediction performance of the proposed CNN model on the maize dataset

Category	TP	FP	FN	TN	Accuracy (%)	Specificity (%)	Sensitivity (%)	Precision (%)	F1 score (%)	MCC
GLS	46	0	6	336	98.45	100	88.46	100	93.88	0.9323
CR	120	0	0	268	100	100	100	100	100	1.0000

NCLB	97	5	2	284	98.20	98.27	97.98	95.1	96.52	0.9532
Healthy	117	1	0	270	99.74	99.63	100	99.15	99.57	0.9939
Average					99.1	99.48	96.61	98.56	97.49	0.9698

5.2 Grad-CAM visualization of activation maps

We have plotted the Grad-CAM [42] to visualize the activation maps of the last convolution layer involved in the proposed model for classifying the images of diseased maize. The Grad-CAM maps of the diseased images were generated by creating the heatmaps[1] of the feature maps from last convolution layer (i.e *cbn_8* layer) and superimposing them with the original input images. Here, we used the "*jet*" colormap of the matplotlib[2] python package for creating the heatmaps of the feature maps. In Grad-CAM maps, the yellow area represented the activation areas of images where the model paid the most attention while classifying the images, whereas the blue and red areas had no impact on the model's classification process as shown in Fig 8 (a, b). Therefore, we can

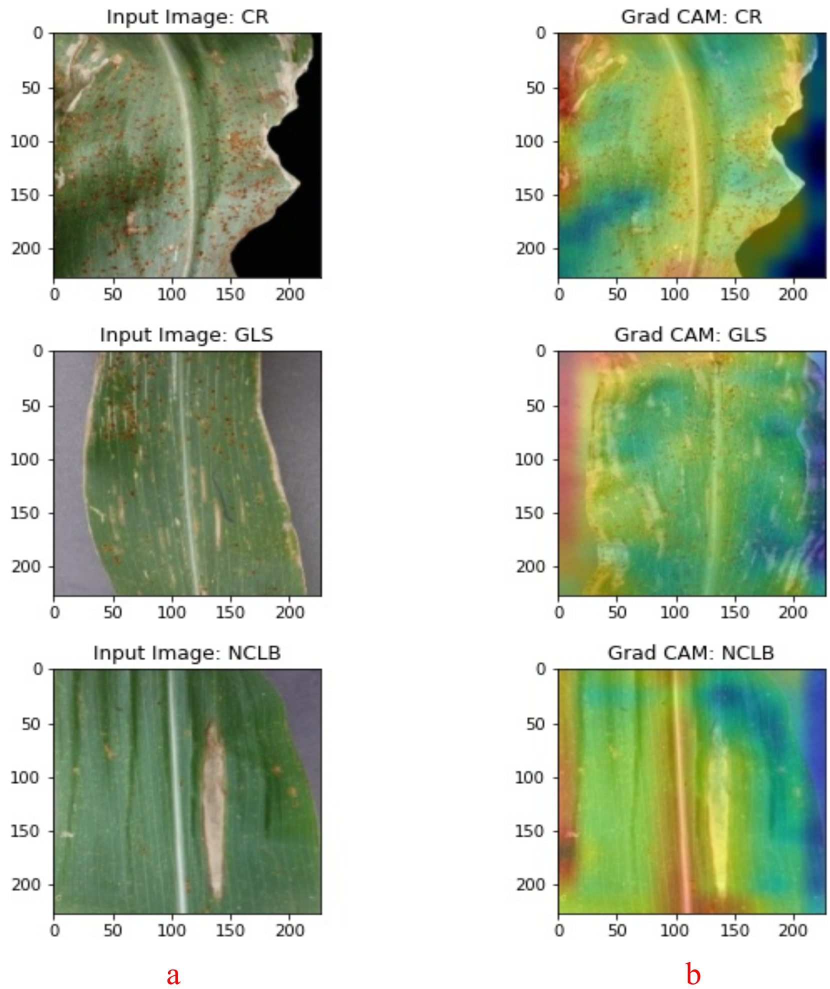

Fig. 8 Grad-CAM visualization of the activations of the proposed model on the maize dataset. a) original input images of 3 diseases of maize respectively Common Rust, Grey leaf Spot and Northern Corn Leaf Blight b) GAD-CAM maps obtained by superimposing the heatmaps (using '***jet***' colormap) of the activations of last convolution layers with original images of the dataset.

---

[1] A heatmap is a graphical representation of data in matrix form where each value of a matrix is represented as different shades of single color model.

[2] Matplotlib is a comprehensive python package for data visualization

observe that our model was able to capture the inherent features from the underlying images effectively and classified the images based on those inherent features. Hence, it is clear that the model made the predictions by identifying the correlated features from the images rather than some random background features or noises.

5.3 Prediction performance of transfer learning approach:

We also applied the transfer learning approach in the images of maize crop using 10 different popular CNN architectures viz. VGG16, VGG19, InceptionV3, ResNet50, Xception, MobileNetV2, DenseNet121, NASNetMobile, EfficientNetB0 and EfficientNetB7. These networks were trained and validated using the large 'ImageNet' data [33] and their learned weights were applied to classify the images of maize crop. In this study, only the last (top) layer of these models was replaced with a GAP layer and a fully-connected or dense layer containing 4 nodes and the remaining layers were kept frozen. The pre-trained models were trained and validated on the images of maize crop in exactly same configurations as our proposed CNN model. The experimental results showed that the pre-trained models were good at classifying the testing set with more than 90% accuracy except for the EfficientNetB0 and EfficientNetB7 models as presented in Table 6. Here we can see that the Xception network showed the highest accuracy (i.e. 95.42%) for classifying the images of maize diseases, whereas the EfficientNetB0 model obtained the lowest classification accuracy (i.e. 69.93%). Now, while looking at the F1-scores of the pre-trained models in Table 6, it could be noted that DenseNet121 model achieved the highest F1-score of 94.37% among the pre-trained models which was at par with its classification accuracy (i.e. 94.12%). Whereas, Xception model achieved the F1-score of 90.28% only which is quite lower than its classification accuracy (i.e. 95.42%) which implies that the DenseNet121 model was more efficient than the Xception model. And again EfficientNetB0 and EfficientNetB7 models achieve significantly lower F1-scores. Therefore, it can be concluded that based on the overall prediction performance DenseNet121 was comparatively better at performing the classification of the maize dataset than other pre-trained models.

Table 6: Overall prediction performance of the pre-trained models on the maize dataset

Model	**Testing Accuracy (%)**	**Testing Loss**	**Average Precision (%)**	**Average Sensitivity (%)**	**Average F1-Score (%)**
VGG16	92.16	0.1753	90.60	90.24	90.39
VGG19	93.46	0.2249	92.37	93.06	92.69
InceptionV3	93.50	0.1473	88.81	89.45	89.06
ResNet50	89.54	0.3331	87.61	86.98	87.22
Xception	**95.42**	0.1349	90.00	90.68	90.28
MobileNetV2	92.81	0.3569	92.44	92.87	92.63
DenseNet121	**94.12**	0.1863	94.66	94.15	**94.37**
NASNetMobile	91.50	0.3507	90.60	87.16	88.04
EfficientNetB0	69.93	1.2900	54.73	61.98	58.07
EfficientNetB7	74.51	1.2640	50.67	57.55	53.65

5.4 Comparative performance analysis with pre-trained networks

Next, we performed a detailed comparative analysis of the prediction performances between the proposed CNN model and pre-trained models and discussed them in the below sub-sections:

Classification accuracy and loss: In Fig 9 (a, b) we presented the classification accuracies and the losses of all the models on the testing set. Here, it can be observed that our proposed CNN model acquired the classification accuracy of 99.02%, which was significantly higher than the best performing pre-trained model i.e. DenseNet121 model. Even our proposed CNN model obtained a very low testing loss (i.e. 0.077) as compared to the pre-trained models. Therefore, it is evident that our proposed CNN model performed comparatively better than state-of-the-art pre-trained models both in terms of the overall classification accuracy and obtained testing loss.

Average F1-scores: Now, as our dataset was unbalanced, classification accuracy wouldn't be justified to make the comparative analysis. Therefore, we presented the comparative analysis of all the models in terms of average precision, sensitivity and F1-score in Fig. 9 (c, d). A closer look at the average F1-score depicts that our proposed CNN model achieved higher F1-score (i.e. 97.49%) than that of the best-performing pre-trained models. It implies that the proposed CNN model was better at predicting the unknown images of each maize disease class than the applied pre-trained models. These results also support that our proposed model is less prone to errors/mistakes while classifying the diseased images of maize crop into respective classes.

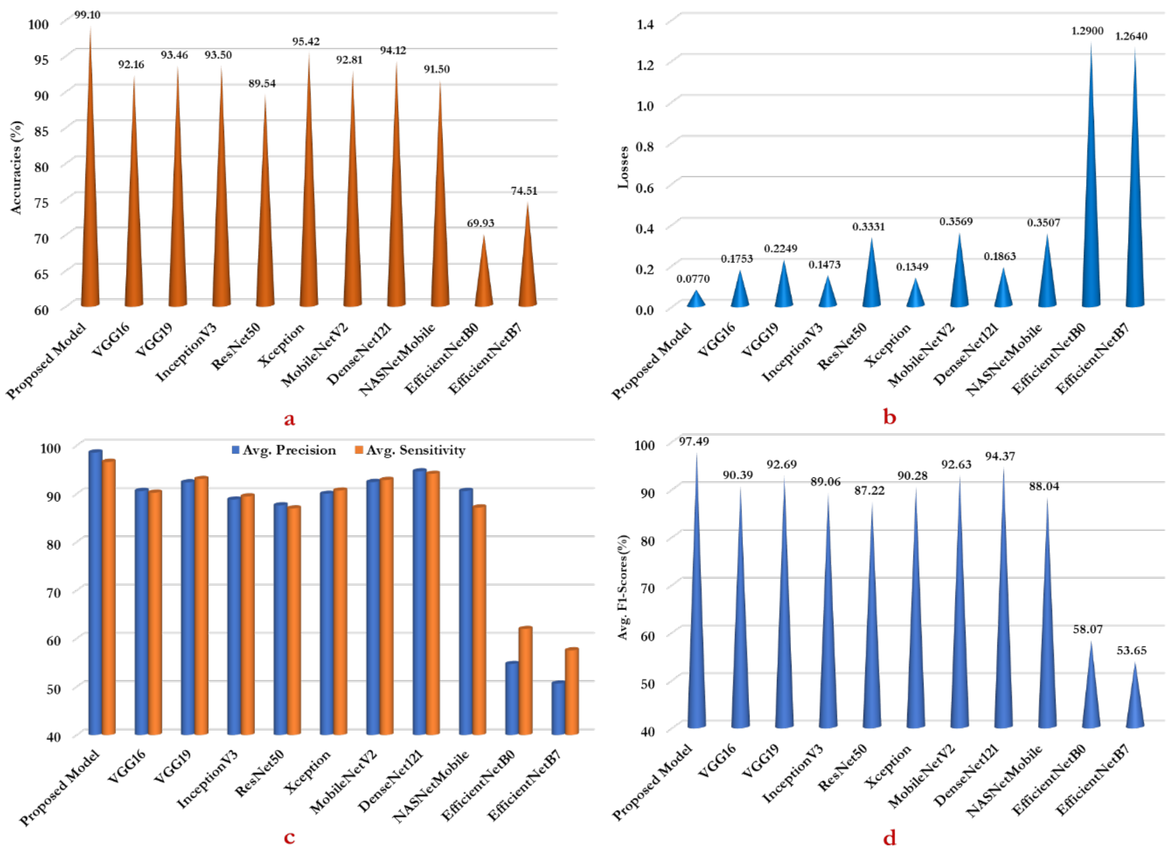

Fig. 9 Comparative performance analysis between proposed CNN model and pre-trained models in terms of a) classification accuracy and b) testing loss c) Average precision and sensitivity and d) Average F1-score

Computational complexity analysis: In this experiment, we also measured the computational complexity of the models with respect to the '*training time per epochs*' and '*number of training parameters*'. In Fig. 10 (a) and 10 (b), we can observe that our proposed CNN had lesser training parameters of around 2.5M and the training time per epoch was also quite low i.e. only 13s. Whereas, most of the state-of-the-art pre-trained models have higher training parameters as well as the training time per epoch. This supports the fact that our proposed model was less complex than the pre-trained models yet provides better prediction performance. Therefore, the overall comparative analysis of the models, it can be concluded that our proposed CNN model is quite effective in classifying and identifying the diseased images of maize crop.

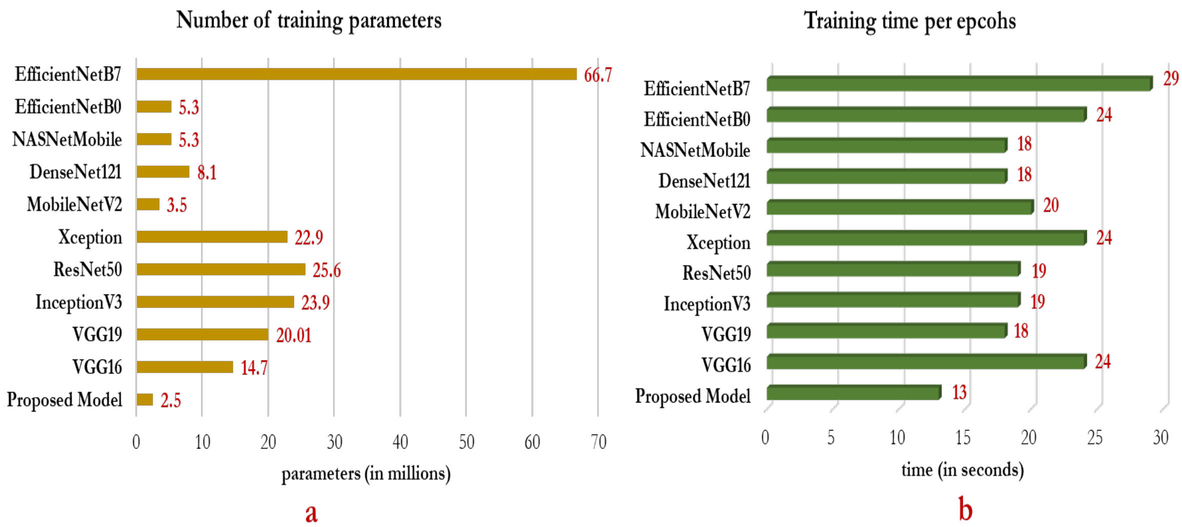

Fig. 10 Comparative computational cost analysis in terms of: a) *number of training parameters* and b) *training time per epochs*

5.5 Comparison with the previous works

In this section, we compared the classification performance of our proposed model with the approaches available in the literature [15–17, 19, 20] and presented a comparative analysis in Table 7. Here, we considered those studies that used the PlantVillage data-repository for model development. We divided the table into two parts where in Part-A we presented the works having the same number of classes in the dataset as ours and, in Part-B, we presented the works with more number of classes in the datasets. However, it can be observed that our proposed method obtained the highest classification accuracy as compared to other works available in the literatures. Therefore, it can be concluded that our proposed 15-layer CNN model is best in recognizing the images of diseases in maize crop.

Table 7: Comparison between proposed approach and approaches available in the literature

References	Classes	Dataset	Models	Results
Our work	4 class	PlantVillage	Custom CNN	Accuracy: 99.1%
Part-A: Approaches with 4 class datasets				
Sibiya et al. [16]	4 class	PlantVillage	Custom CNN	Accuracy: 92.85%
Priyadarshini et al. [17]	4 class	PlantVillage	Modified LeNet	Accuracy: 97.89%
Waheed et al. [20]	4 class	PlantVillage	Modified DenseNet	Accuracy: 98.06%

Part-B: Approaches with more than 4 class datasets				
Zhang et al. [15]	9 class	PlantVillage and Internet sources	GoogleNet and Cifar10	Accuracy: GoogleNet: 98.9% Cifar10: 98.8%.
Lv et al.[19]	7 class	PlantVillage, Global AI-challenge data and Internet sources	Modified AlexNet	Accuracy: 98.62%

6. Conclusion

In this research work, we proposed a novel 15-layer deep convolutional neural network model to identify the images of diseases of the maize crop. The proposed model was able to classify the images of maize into three disease categories such as grey leaf spot (GLS), common rust (CR), and northern corn leaf blight (NCLB) along with healthy ones. We used the maize dataset from the PlantVillage data repository to train, validate and test our proposed model. To avoid the overfitting issue of the model, we applied an online data augmentation technique in which a batch of images was augmented during the runtime of the model training. We trained our proposed model with 90% (1,376 images) of the whole data and tested on the remaining 10% data. The experimental results of the proposed model were quite satisfactory for classifying the unseen images of maize data. The proposed model obtained an overall classification accuracy of 99.10% along with F1-score of 97.49% in testing set of the maize dataset. Furthermore, the Grad-CAM visualization of the activation maps states that the model classifies the images based on the inherent features into their respective classes. We also applied the transfer learning approach to a few state-of-the-art models and presented comparative performance analysis between the proposed and pre-trained models. The comparative analysis showcases the effectiveness of our proposed CNN model over the pre-trained models. From the overall performance analysis of the proposed model, it can be concluded that our proposed model sufficiently captures the promising features of the input images for classifying the diseases of maize crop. Therefore, automated recognition of diseases of the maize crop is feasible using the proposed CNN model and will ultimately support the proper crop management practices.

However, the proposed model was applicable only for the images captured on uniform background images. Therefore, in the future course of study, the proposed model could be trained and evaluated on the diseased images captured in the normal background conditions to address field-level crop monitoring. Additionally, several diseases of other important crops could also be targeted for training & testing with the proposed model for the betterment of the farmers and to support the overall global food supply chain.

Author's Contribution

MH, SM and AA conceived the study. MH conducted the experiments and implemented the models described. MH, CK and SN analysed the results and wrote the manuscript.

Competing Interests

The author(s) declare no competing interests

Acknowledgment

This study has been supported by the National Agricultural Science Funds (NASF), ICAR and National Agricultural Higher Education Project (NAHEP), ICAR.

Data Availability:

The dataset used and analysed in this study is publicly available in https://github.com/spMohanty/PlantVillage-Dataset

References:

1. United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019: Highlights (ST/ESA/SER.A/423)
2. Kaur H, Kumar S, Hooda KS, et al (2020) Leaf stripping: an alternative strategy to manage banded leaf and sheath blight of maize. Indian Phytopathology 73(2):203–211. https://doi.org/10.1007/s42360-020-00208-z
3. FAOSTAT (2021) Statistical Database of the Food and Agriculture of the United Nations. In: FAO. http://www.fao.org. Accessed 4 Jun 2021
4. Annual Maize Progress Report Kharif 2020. ICAR-IIMR, PAU Campus, Ludhiana-141004
5. Oerke EC, Dehne HW (2004) Safeguarding production - Losses in major crops and the role of crop protection. Crop Protection 23(4):275–285. https://doi.org/10.1016/j.cropro.2003.10.001
6. Donatelli M, Magarey RD, Bregaglio S, et al (2017) Modelling the impacts of pests and diseases on agricultural systems. Agricultural Systems 155:213-224. https://doi.org/10.1016/j.agsy.2017.01.019
7. LeCun Y, Bengio Y, Hinton G (2015) Deep Learning. Nature 521(7553):436–444
8. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: A survey. Computers and Electronics in Agriculture 147:70–90
9. Haque MA, Marwaha S, Arora A, et al (2021) Image-based identification of maydis leaf blight disease of maize (Zea mays) using deep learning. Indian Journal of Agricultural Sciences 91(9):1362–1367
10. Hughes DP, Salathé M (2016) An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060
11. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Frontiers in Plant Science 7:1–20. https://doi.org/10.3389/fpls.2016.01419
12. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp:25
13. Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. PP 1-9.
14. DeChant C, Wiesner-Hanks T, Chen S, et al (2017) Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology 107(11):1426–1432. https://doi.org/10.1094/PHYTO-11-16-0417-R
15. Zhang X, Qiao Y, Meng F, et al (2018) Identification of maize leaf diseases using improved deep convolutional neural networks. IEEE Access 6:30370-30377. https://doi.org/10.1109/ACCESS.2018.2844405
16. Sibiya M, Sumbwanyambe M (2019) A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. AgriEngineering. 1(1):119-131. https://doi.org/10.3390/agriengineering1010009
17. Priyadharshini RA, Arivazhagan S, Arun M, Mirnalini A (2019) Maize leaf disease classification using deep convolutional neural networks. Neural Computing and Applications 31(12):8887-8895. https://doi.org/10.1007/s00521-019-04228-3
18. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: Proceedings of the 86th IEEE pp:2278-2324 https://doi.org/10.1109/5.726791
19. Lv M, Zhou G, He M, et al (2020) Maize Leaf Disease Identification Based on Feature Enhancement and DMS-Robust Alexnet. IEEE Access 8:57952-57966. https://doi.org/10.1109/ACCESS.2020.2982443
20. Waheed A, Goyal M, Gupta D, et al (2020) An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. Computers and Electronics in Agriculture 175:105456. https://doi.org/10.1016/J.COMPAG.2020.105456

21. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition pp:4700-4708

22. Chen J, Wang W, Zhang D, et al (2021) Attention embedded lightweight network for maize disease recognition. Plant Pathology 70(3):630–642. https://doi.org/10.1111/ppa.13322

23. Ward JMJ, Stromberg EL, Nowell DC, Nutter FW (1999) Gray leaf spot: A disease of global importance in maize production. Plant Disease *83*(10):884-895. https://doi.org/10.1094/PDIS.1999.83.10.884

24. Aggarwal SK, Gogoi R, Rakshit S (2021) Major Diseases of maize and their management. IIMR Technical Bulletin 2021-04. ICAR-IIMR, Ludhiana

25. Hooda KS, Khokhar MK, Shekhar M, et al (2017) Turcicum leaf blight—sustainable management of a re-emerging maize disease. Journal of Plant Diseases and Protection 124(2):101-113

26. Géron A (2017) Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc

27. Scotti F (2005) Automatic morphological analysis for acute leukemia identification in peripheral blood microscope images. In: Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications pp:96-101

28. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press

29. Gu J, Wang Z, Kuen J, et al (2018) Recent advances in convolutional neural networks. Pattern Recognition 77:354–377. https://doi.org/10.1016/J.PATCOG.2017.10.013

30. Zhang A, Lipton ZC, Li M, Smola AJ (2021) Dive Into Deep Learning. arXiv preprint arXiv:2106.11342

31. Haque MA, Marwaha S, Deb CK, et al (2022) Deep learning-based approach for identification of diseases of maize crop. Scientific Reports 12(1):6334. https://doi.org/10.1038/S41598-022-10140-Z

32. Atila Ü, Uçar M, Akyol K, Uçar E (2021) Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics 61:101182. https://doi.org/10.1016/j.ecoinf.2020.101182

33. Deng J, Dong W, Socher R, et al (2009) ImageNet: A Large-Scale Hierarchical Image Database. In: conference on computer vision and pattern recognition, IEEE pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

34. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations. arXiv preprint arXiv:1409.1556

35. Szegedy C, Vanhoucke V, Ioffe S, et al (2016) Rethinking the Inception Architecture for Computer Vision. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp:2818-2826

36. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp: 770-778

37. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition pp:1251-1258

38. Sandler M, Howard A, Zhu M, et al (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp:4510-4520

39. Zoph B, Vasudevan V, Shlens J, Le Q v. (2018) Learning Transferable Architectures for Scalable Image Recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp:8697-8710

40. Tan M, Le Q v. (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. In: 36th International Conference on Machine Learning pp:6105-6114

41. Abadi M, Agarwal A, Barham P, et al (2016) TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv preprint arXiv:1603.04467

42. Selvaraju RR, Cogswell M, Das A, et al (2017) Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: Proceedings of International Conference on Computer Vision pp: 618-626