



## Article

# IoT and Interpretable Machine Learning Based Framework for Disease Prediction in Pearl Millet

Nidhi Kundu <sup>1</sup>, Geeta Rani <sup>1,\*</sup>, Vijaypal Singh Dhaka <sup>1</sup>, Kalpit Gupta <sup>1</sup>, Siddaiah Chandra Nayak <sup>2</sup>, Sahil Verma <sup>3</sup> , Muhammad Fazal Ijaz <sup>4,\*</sup>  and Marcin Woźniak <sup>5</sup> 

<sup>1</sup> Department of Computer and Communication Engineering, Manipal University Jaipur, Jaipur 303007, India; kundu.nidhi1990@gmail.com (N.K.); vijaypalsingh.dhaka@jaipur.manipal.edu (V.S.D.); kalpit.189201045@muj.manipal.edu (K.G.)

<sup>2</sup> ICAR DOS in Biotechnology, University of Mysore Manasagangotri, Mysore 570005, India; moonnayak@gmail.com

<sup>3</sup> Department of Computer Science and Engineering, Chandigarh University, Mohali 140413, India; sahilverma@ieee.org

<sup>4</sup> Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea

<sup>5</sup> Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland; marcin.wozniak@polsl.pl

\* Correspondence: geetachikara@gmail.com (G.R.); fazal@sejong.ac.kr (M.F.I.)

**Abstract:** Decrease in crop yield and degradation in product quality due to plant diseases such as rust and blast in pearl millet is the cause of concern for farmers and the agriculture industry. The stipulation of expert advice for disease identification is also a challenge for the farmers. The traditional techniques adopted for plant disease detection require more human intervention, are unhandy for farmers, and have a high cost of deployment, operation, and maintenance. Therefore, there is a requirement for automating plant disease detection and classification. Deep learning and IoT-based solutions are proposed in the literature for plant disease detection and classification. However, there is a huge scope to develop low-cost systems by integrating these techniques for data collection, feature visualization, and disease detection. This research aims to develop the ‘Automatic and Intelligent Data Collector and Classifier’ framework by integrating IoT and deep learning. The framework automatically collects the imagery and parametric data from the pearl millet farmland at ICAR, Mysore, India. It automatically sends the collected data to the cloud server and the Raspberry Pi. The ‘Custom-Net’ model designed as a part of this research is deployed on the cloud server. It collaborates with the Raspberry Pi to precisely predict the blast and rust diseases in pearl millet. Moreover, the Grad-CAM is employed to visualize the features extracted by the ‘Custom-Net’. Furthermore, the impact of transfer learning on the ‘Custom-Net’ and state-of-the-art models viz. Inception ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19 is shown in this manuscript. Based on the experimental results, and features visualization by Grad-CAM, it is observed that the ‘Custom-Net’ extracts the relevant features and the transfer learning improves the extraction of relevant features. Additionally, the ‘Custom-Net’ model reports a classification accuracy of 98.78% that is equivalent to state-of-the-art models viz. Inception ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19. Although the classification of ‘Custom-Net’ is comparable to state-of-the-art models, it is effective in reducing the training time by 86.67%. It makes the model more suitable for automating disease detection. This proves that the proposed model is effective in providing a low-cost and handy tool for farmers to improve crop yield and product quality.

**Keywords:** machine learning; interpretable; context-aware; deep learning; IoT



**Citation:** Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woźniak, M. IoT and Interpretable Machine Learning Based Framework for Disease Prediction in Pearl Millet. *Sensors* **2021**, *21*, 5386. <https://doi.org/10.3390/s21165386>

Academic Editor: Sigfredo Fuentes

Received: 10 June 2021

Accepted: 28 July 2021

Published: 9 August 2021

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The traditional systems of farming focus on meeting the dietary requirements of people and domestic animals. Therefore, the farmers used to grow more nutritious cereals

such as millets and sorghum rather than high-yielding grains such as rice and wheat. With the commercialization of agriculture, the farmers have shifted their interest towards high crop yields that can fulfill their dietary and financial requirements. This shift has increased the burden of malnutrition, causing undernourishment and micronutrient deficiencies [1]. Therefore, there is a need to implement the precision system of agriculture that improves the yield and the quality of highly nutritious crops.

The prime minister recognized millets as a treasure of nutrition and commended for a call to start a millet revolution in India. He has declared millets as the 'Nutri Cereals' for production, consumption, and trade points [2]. In addition, the Union Ministry for Human Resource Development (MHRD) [3] has requested states to include Millet in Mid-day meals served in schools. Moreover, a continuous decrease in the yield of common crops such as wheat, rice, groundnut, and maize [4,5] has attracted farmers to grow pearl millet. Pearl millet is resilient to climate issues due to its less water demand of 200 to 600 mm, stability at high temperatures, and drought-prone ability. Therefore, millets with their 'Nutri Cereals' capability can be tapped for food security in the future. To meet this rising demand, the farmers started the use of fertilizers, pesticides, and controlled irrigation. This has increased the global crop yield of pearl millet over the past 50 years [6,7]. As per the Project Coordinator, a review report published by the Directorate of Millets Development, 2020, the pearl millet covers 6.93 million hectares of land. The average production of 8.61 million tons was reported during 2018–2020 [5].

The productivity and quality of pearl millet are adversely affected by plant diseases such as blast and rust [5]. These diseases put a substantial threat to food security and harm the economy of farmers [8,9]. Therefore, it is mandatory to introduce a system for the detection of diseases in crop plants.

Context-aware and interpretable machine learning (ML) and deep learning (DL) have gained remarkable attention in human health monitoring [10–16], crop health monitoring, and yield prediction [17]. These models are effective to give automatic, accurate, and quick systems for plant disease detection and classification. Several models of ML and DL are found effective and precise in disease detection and classification [18]. However, as per the discussion given in [19,20], the convolutional neural network (CNN) outperforms the machine learning (ML) models due to their potential in automatic feature extraction. However, CNN models demand a huge and labeled dataset for training. The collection of the dataset and correct labeling of a disease in a vast dataset is challenging as it requires time and effort from the experts.

The frameworks based on the Internet of Things (IoT) have proven their acceptance in automating data collection, data storage, and processing of collected datasets to make real-time predictions [10,11,21]. The adequate use of drone, camera, and sensors significantly reduce the time and cost of labor.

Furthermore, the advancements in deep transfer learning bring down the requirement for the vast dataset [22]. In this approach, the model is initially trained on an extensive dataset that is not necessarily labelled. At this stage, the model learns about the low-level features such as texture, pixel intensity, marking of boundaries, etc. The weights of this trained model are saved and utilized for further training the model with the dataset comprising labelled samples [22].

The potential of IoT in data collection, data storage, quick processing, and efficacy of interpretable ML, DL, and transfer learning techniques in object detection, classification, visualization, and pattern matching even with the small-sized labelled dataset [19,23,24] motivated the authors to employ the integration of these techniques for developing a framework for detection of disease in pearl millet.

In this manuscript, the authors propose an IoT and interpretable deep transfer learning-based framework, 'Automatic and Intelligent Data Collector and Classifier' (AIDCC), for the detection and classification of diseases viz. rust and blast [25] in pearl millet.

The significant contributions of this manuscript are as follows:

1. Highlighting the need to automate the detection of diseases in the underexplored crop 'pearl millet'.
2. Automatic collection of the real-time datasets by the IoT system fixed at the farmlands of pearl millet.
3. Developing the IoT and deep transfer learning-based framework for detection and classification of diseases in pearl millet.
4. Presenting the comparative analysis of the proposed framework and the systems available in the literature to detect and classify plant diseases.

## 2. Related Works

The extensive study of literature in plant disease detection and classification gives insights into the techniques employed to collect datasets, pre-processing, disease detection, classification, and visualization.

In the traditional approaches, the farmers manually detect diseased and healthy plants [26]. These approaches lack in the tracking of the essential parameters such as soil type, humidity, temperature, amount of macro and micronutrients in the soil, and nutrient requirements of the crop plant at different stages of its growth and maturity. Moreover, the traditional approaches are time-consuming and need a lot of human effort. Furthermore, the farmers need advice from experts for the correct diagnosis of diseases in crop plants.

The applications of IoT, computer vision, ML, DL, and deep transfer learning have streamlined the automation of plant disease detection and classification [27,28]. In this line of research, the authors proposed IoT and ML models for data capturing and disease prediction [29]. They used the drone for capturing the images over a large area in less time. They applied the support vector machine (SVM) for the classification of diseases in rice crops. However, their system did not consider the on-demand capturing of images for real-time monitoring and prediction.

Furthermore, the authors in [30] utilized the potential of IoT to categorize the healthy and diseased leaves. They anchored the sensors for monitoring of soil quality, temperature, and humidity. They used the camera to capture the images of crop plants. The authors established the interface of sensors and a camera with the Raspberry Pi to store and process the captured data for real-time predictions. They employed the K-means algorithm for clustering images followed by masking of pixels to detect whether the leaf is diseased or healthy.

The authors in [31,32] stated that DL techniques are effective in the early detection of crop diseases. They recommended the use of these techniques to overcome the limitations of traditional approaches. The research works presented in [24,27,32–38] and [39–46] introduced deep learning (DL) models for the detection and classification of plant diseases. Furthermore, Mohammed Brahimi claimed the supremacy of deep transfer learning over deep learning [27].

The authors in [39] collected a dataset of 36,258 images from the AI challenger [47]. However, the dataset comprised images of poor visual quality. They employed the ResNet model on the collected dataset and reported an accuracy of 93.96%.

The works presented by the authors in [19,48,49] highlighted the importance of collecting imagery datasets and employing an appropriate DL model on the collected dataset for the detection and classification of plant diseases. They also focused on integrating DL models with the IoT systems comprising sensors, a drone, a camera, etc. They claimed that these integrated systems effectively minimize human efforts and reduce the time required for different agricultural practices. These systems are capable of gathering real-time information from farms and quick processing of the collected datasets to predict plant diseases.

The research works discussed by C. Shorten and T. M. Khoshgoftaar in [50] and P. Cao et al., in [51] clarified that employing the augmentation techniques such as geometric transformations, colour space augmentations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, neural style transfer, and meta-learning

may prove constructive to improve the performance of the DL models. To carry out further research, the authors in [52] conducted the experiments using 124 images downloaded from the Internet. They applied data augmentation techniques such as zoom, rotation, flip, and rescale to increase the size of the dataset to 711 images. In addition, they reported the training accuracy of 95% and the validation accuracy of 89%. The low validation accuracy and impractical implementation using standard memory devices such as mobile phones are the significant limitations of this research. Moreover, they did not consider the parameters such as soil type, temperature, humidity, nutrient requirements, etc. while disease detection. Furthermore, the authors focused only on detecting one disease, ‘downy mildew’ in pearl millet. Therefore, there is considerable scope for improving the performance and working on the most common diseases such as blast and rust.

One more research group in [52] exploited the applications of deep transfer learning. They employed pre-trained VGG-16 [53] to detect downy mildew disease [54] in pearl millet. Based on the experiments, they claimed that deep transfer learning effectively extracts the essential features. The extracted features of the pre-trained network are available for reuse. The pre-trained networks reuse these features and continue learning from the more dataset available for training. This improves the performance of the model. Transfer learning is also important for fine-tuning the model according to the size and type of the dataset. The authors also claimed that transfer learning is helpful to avoid overfitting and to improve the model’s predictive capacity [24].

To work in synergy with the system proposed in [52], the authors in [23,55] integrated IoT and DL techniques for disease detection in crops. However, the system could not prove its practical application due to low accuracy.

The above discussion of the related research works shows that the integration of DL techniques with IoT provides good opportunities for developing the architectures to automate the collection of imagery and parametric data, storage of collected dataset, plant disease detection, generating alerts, and classification of detected diseases. However, these works lack in sensing the parameters that are the root cause for the plant diseases. Moreover, the collection of imagery datasets requires substantial human effort and high cost. Furthermore, the DL models employed for disease detection and classification report low accuracy and take more time to respond. To the best of our knowledge, there is no automatic and intelligent system for identifying and classifying blast and rust diseases in pearl millet. The potential of integration of IoT and deep transfer learning is still underexploited in the field of agriculture.

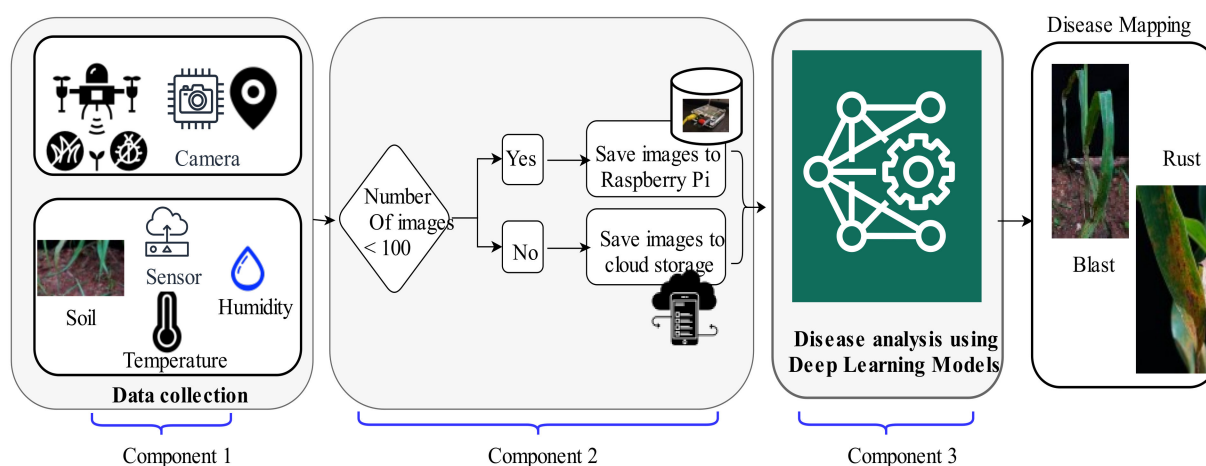
Therefore, there is a huge scope for improving the performance of the existing systems and providing a new architecture for the automatic collection of the dataset, detection, and classification of diseases in pearl millet.

### 3. Materials and Methods

In this section, the authors present the details of the proposed framework, dataset prepared, training mechanism, and evaluation metrics used to evaluate the performance of the models.

#### 3.1. Proposed Framework

In this manuscript, the authors propose the ‘Automatic and Intelligent Data Collector and Classifier’ (AIDCC) for the collection of data, detection, and classification of rust, and blast diseases in pearl millet. The framework is an integration of three components, as demonstrated in Figure 1.



**Figure 1.** The framework of ‘Automatic and Intelligent Data Collector and Classifier’.

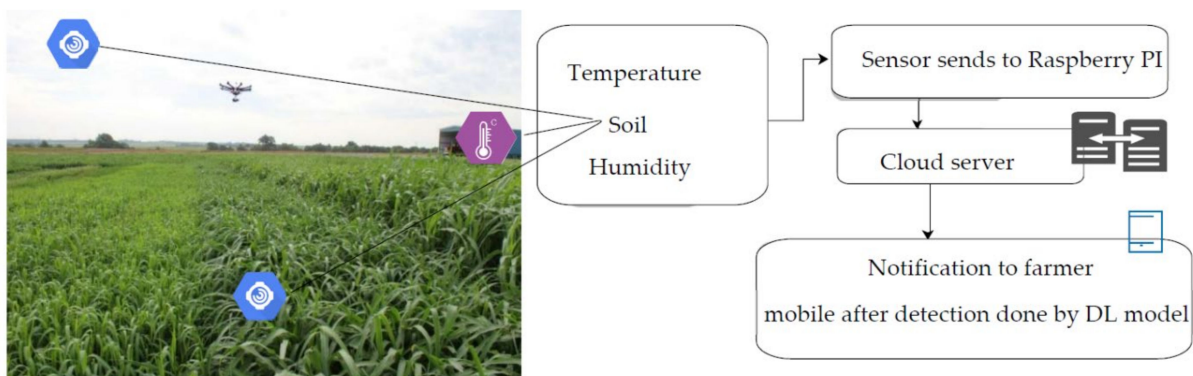
Component 1: It comprises the digital drone, camera, global positioning system (GPS), and sensors. A digital drone is a crewless aerial vehicle used to monitor the farmlands [56]. Here, the drone is equipped with a Panasonic GH3 camera ‘DJI S1000’ that can focus in the range from 25 to 30 m, offers video resolution of  $1920 \times 1080$ , and a CMOS sensor of 16 MP. The camera of the drone clicks the images and transfers them automatically and instantly to the Raspberry Pi and/or cloud storage. It also captures the variation in RGB scaling of plants to spot the major disease areas in the farm. The drone is specified for a flying range of 7 km. The patrolling of the farm using a drone is useful in obtaining the coordinates of the field used by GPS. It saves time and the cost of labor.

In addition to the drone, we used the NIKON D750 digital camera for clicking the pictures. The digital camera is used to capture the desired region such as leaves rather than the complete plant or multiple plants together. Similar to the drone camera, it transfers the captured images directly to the cloud server.

The GPS module embedded with sensors, drones, and Raspberry Pi is helpful to monitor the location of the diseased plants. It is also important to find the region of farmland where fertilizers and/or water are required. The sensors are anchored with the proposed framework to monitor the changes in soil, temperature, and humidity. For example, the variations in temperature and moisture of soil indicate the susceptibility of pearl millet towards the blast and rust diseases [25]. Moreover, the oospores present in the soil are the primary source to infect the underground parts of plants [54]. Therefore, the sensors anchored for continuous monitoring of the soil can detect the presence of oospores in the soil and help in predicting the disease at an early stage. Furthermore, the hyperspectral sensors are fixed with drone cameras for monitoring the environmental and physical conditions.

To identify the suitable sensors for the AIDCC, the authors referred to the system proposed in [57]. They used four sensors viz. GY-30, soil sensor, DHT22, and BMP180 sensors to measure the humidity in the soil, temperature, and light intensity. The authors embedded the DS3231 sensor for transferring the information from the mounted sensors to the processor of Raspberry Pi (RPI). Taking clues from the work proposed by N. Materne and M. Inoue in [58] and A. Thorat et al. in [30], we fitted two DHT11 sensors to measure the temperature and humidity in order to spot the rust and blast diseases at an early stage. These sensors are connected to the Raspberry Pi (RPI) for transmitting the captured information to the cloud server. This information is disseminated as an alert or notification to the farmers on their mobile phones. The role of sensors from information gathering to sending notifications to farmers for a pearl millet farmland is demonstrated in Figure 2.





**Figure 2.** Role of sensors in the pearl millet farmland.

**Component 2:** This component comprises the Raspberry Pi and cloud storage. It receives the parametric and imagery dataset collected by component 1. Raspberry Pi can store up to 100 images due to its limited storage capacity. Therefore, the photos are sent to the cloud server, if their number exceeds 100, as demonstrated in Figure 1.

**Component 3:** In this component, the DL based classifier classifies the data stored at the cloud server and Raspberry Pi into the rust and blast classes. Component 3 works synchronously with the Raspberry Pi for facilitating real-time predictions and notifying the farmers about the diseases or other variations observed in the farmland.

### 3.2. Dataset Preparation

For the dataset collection, the hardware components such as drones, digital cameras, and sensors as shown in component 1 of Figure 1, were fixed at the farmland of the Indian Council of Agricultural Research-All India Coordinated Research Project (ICAR (AICRP-Mysore center). The pearl millet plants infected with blast and rust diseases were grown purposefully to monitor the symptoms and impacts of these diseases. The characteristics of diseased leaves of pearl millet are shown in Table 1.

**Table 1.** Characteristics and symptoms of diseased leaves of pearl millet.

| Name of Disease | Causing Agent                           | Stage of Infection           | Shape of Infected Region     | Colour of Infected Region  |
|-----------------|---|------------------------------|------------------------------|--|
| Downy mildew    | <i>Sclerospora graminicola</i>          | Seedling.                    | Foliar and green ear         | Green and whitish  |
| Blast           | <i>Magnaporthe grisea</i>               | Seedling and tillering stage | Elliptical or diamond-shaped | Pale green to greyish green, later turning yellow to grey with age |
| Rust            | <i>Puccinia substriata var. indica.</i> | Before flowering             | Pistules type small spots    | Reddish-orange   |

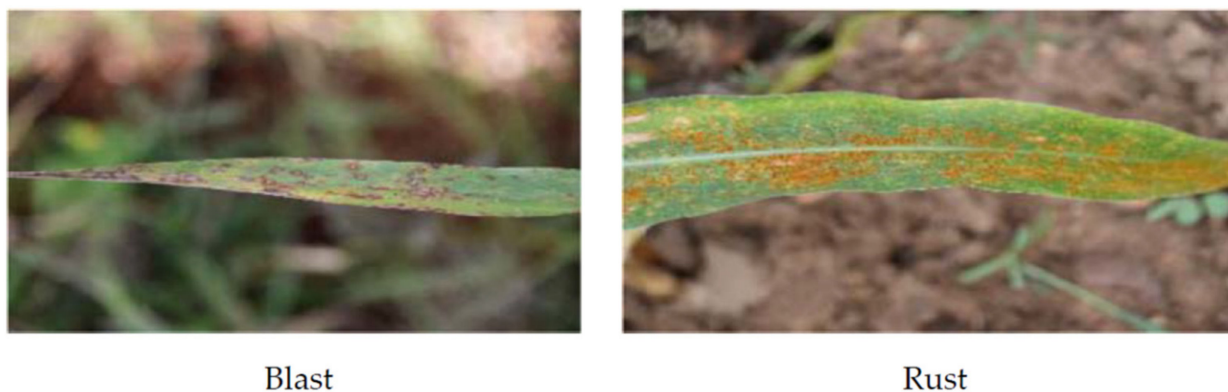
The images of pearl millet plants infected by blast and rust diseases were captured in close observation of the plant pathology expert involved in this research. The authors considered 55- to 60-day-old plants for capturing the images since the blast and rust diseases were easily distinguishable at this age of plants. Moreover, the pathology experts claimed that the degree of severity of rust and blast diseases has reached more than 80% in the plants of this age. In these plants, the pathology experts easily identified rust and blast diseases in pearl millet based on their visible symptoms. For example, the leaves of plants infected with blast turned greyish, and water-soaked lesions appear on the foliage [59]. These lesions vary in size from  $-2$  to 20 mm. The lesions also vary in shape from roundish, elliptical, diamond shaped to elongated. These lesions may enlarge and become necrotic with an increase in the severity of the disease. On the other hand, the leaves of plants infected with rust contain pinhead chlorotic flecks. These flecks turn into reddish-orange as the disease severity increases. Moreover, the round to elliptical pustules appear on

both surfaces of leaves [59]. The observable differences in the patterns of both diseases are important for the precise training of the DL model.

The authors captured 1964 images of leaves of pearl millet infected with blast and 1336 images infected with rust. They divided the prepared dataset into training and testing datasets in the ratio of 70% and 30% of the total dataset, respectively. The number of images in these datasets is shown in Table 2, and the sample images of blast and rust diseases are shown in Figure 3.

**Table 2.** Number of images in training and testing datasets of blast and rust.

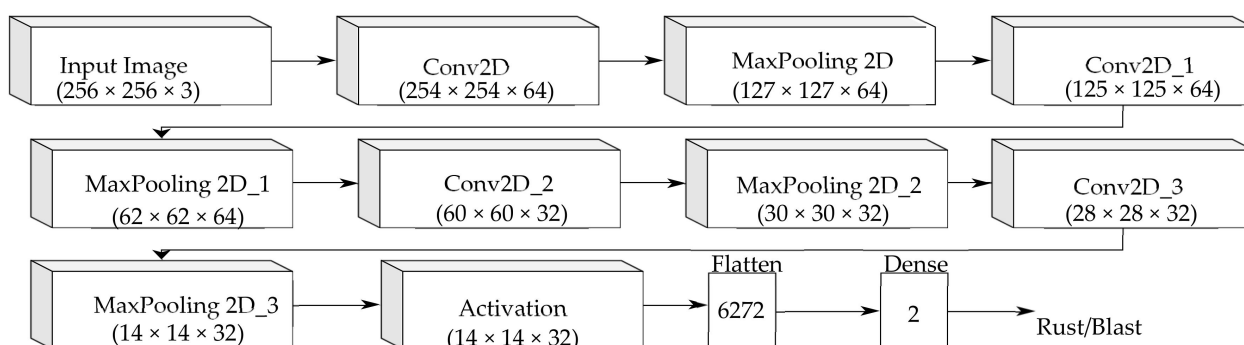
| Name of Disease | Total Number of Images | Number of Images in the Training Dataset | Number of Images in Testing Dataset |
|-----------------|------------------------|--|-------------------------------------|
| Blast           | 1964                   | 1375                                     | 567                                 |
| Rust            | 1336                   | 935                                      | 423                                 |
| Total           | 3300                   | 2310                                     | 990                                 |



**Figure 3.** Sample images of pearl millet infected with blast and rust.

### 3.3. The Architecture of the ‘Custom-Net’ Model

The architecture of the ‘Custom-Net’ model designed to predict the samples infected with blast and rust diseases is shown in Figure 4. It comprises four convolution layers, and a max-pooling layer follows each convolution layer. Furthermore, the last max-pooling layer is followed by the activation, flatten, and dense layer.



**Figure 4.** The architecture of the ‘Custom-Net’ model.

### Training of ‘Custom-Net’ and State-of-the-Art Deep Learning Models

Based on the set of experiments conducted and the experimental results reported in the related works [51,60,61], the authors employed the Adam optimizer to deal with the problems of sparse gradients that may be generated on the noisy dataset. This optimizer

adopts the best properties of AdaGrad and RMSProp optimization algorithms and favors the better training of the model. Moreover, the authors employed the softmax activation function and categorical cross-entropy loss function for precise training of the proposed network. In addition, they set the learning rate of 0.0001 to optimize the learning of the model and obtain its optimum performance. Furthermore, the authors continuously monitored the model's performance and observed that it reports the optimum performance for the batch size of 16 samples.

The authors employed pre-trained and non-pre-trained versions of 'Custom-Net' and state-of-the-art models. The model is named as pre-trained if it is trained on the 'ImageNet dataset' [62], its weights are saved and it is further trained on the dataset collected in this research. The pre-trained model learns the low-level features such as boundary and edge marking from the 'ImageNet dataset'. It further learns the high-level features such as pattern differences in leaves infected with blast and rust diseases, from the dataset used in this manuscript. In contrast, the model is named as non-pre-trained if it is initialized with random weights and directly trained with the dataset collected in this research. The non-pre-trained model learns both the high level as well as low-level features from the dataset used in this manuscript.

Now, to showcase the impact of transfer learning on the shallow neural network, the authors pre-trained the 'Custom-Net' on the publicly available 'ImageNet dataset' comprising more than 14 million images [62] followed by the training on the dataset collected as a part of this research. In addition, they also trained the model only on the collected dataset without using the concept of transfer learning. They compared the results of the pre-trained and non-pre-trained versions of 'Custom-Net' to demonstrate the impact of transfer learning on feature extraction and classification. Moreover, they also plotted the output matrix obtained after each layer of the 'Custom-Net' as shown in Figure 5. This is important to visualize how the 'Custom-Net' extracts the relevant features and ignores the irrelevant features at its different layers. It is evident from Figure 5 that there are no clear boundaries visible at the initial convolution layers. However, the feature map is reduced, and boundaries are more precise at the later convolution layers and their following max-pooling layers. It is apparent from the last matrix shown in Figure 5 that the model learned to identify even the complex patterns hidden in the image.

Moreover, it is clear from the matrices shown in Figure 5 that the model starts learning the pixel-level features at the initial layers. Gradually, it starts discarding the features picked from the background and considers only the relevant features for decision-making once it is trained. They also recorded that each epoch takes 4 s and the model completes its training in 20 epochs. The quick training of the model shows its efficacy in feature extraction.

Now, for comparing the efficacy of the proposed 'Custom-Net' model, the authors employed the pre-trained as well as non-pre-trained versions of the state-of-the-art models viz. VGG-16, VGG-19 [53], ResNet-50 [39], Inception-V3 [42], and Inception ResNet-V2 [41] to predict the samples infected with blast and rust diseases.



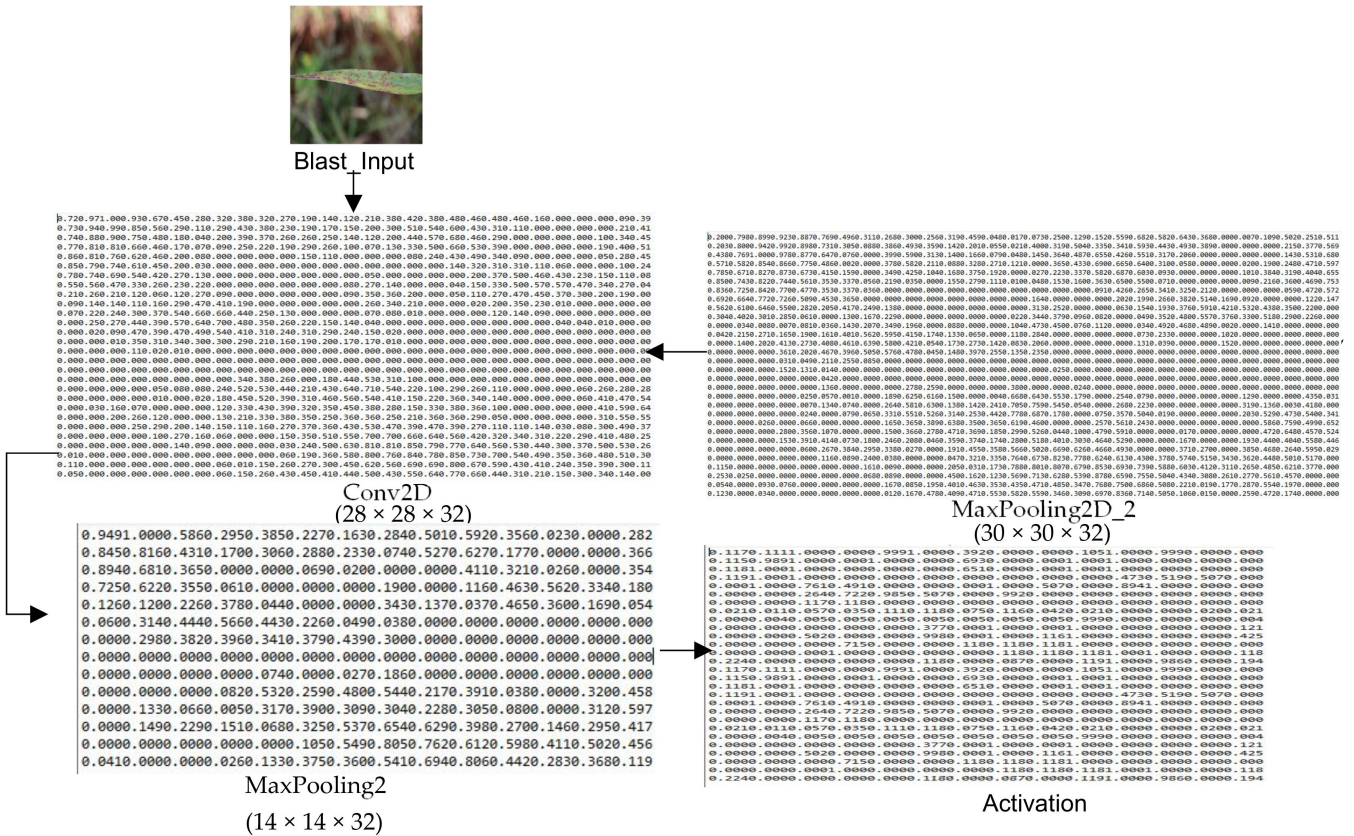


Figure 5. Output matrix of selected layers of ‘Custom-Net’ model.

### 3.4. Evaluation Metrics

To evaluate the performance of the classifiers accompanied by the ‘Automatic and Intelligent Data Collector and Classifier’, the authors used the confusion matrix as presented in [53], average accuracy, precision, recall, and training time. The definitions of these metrics are given below:

**Confusion Matrix:** This represents the number of correctly and incorrectly classified samples into each labelled class. Here, TB denotes the number of correctly classified samples of blast disease, FB denotes the number of incorrectly classified samples of blast disease, TR is the number of correctly classified samples of rust disease, and FR is the number of incorrectly classified samples of rust disease. The sample confusion matrix is shown in Table 3. Based on the labels presented in the confusion matrix, the authors define the evaluation matrices, namely sensitivity, accuracy, precision, recall, F1 score.

1. **Average accuracy:** It is the measure of the degree of correctness of the classification. It can be calculated using the formula given in Equation (1).

$$Accuracy = \frac{TB + TR}{TR + FB + FR + TB} \tag{1}$$

2. **Precision:** It is the measure of classifying the samples of the blast correctly to the blast class. The formula to calculate the precision is given in Equation (2).

$$Precision = \frac{TB}{TB + FB} \tag{2}$$

3. *Recall*: This is the measure of correct identification of samples of the blast class from the total number of samples of that class. The formula to calculate the precision is given in Equation (3).

$$Recall = \frac{TB}{TB + FR} \quad (3)$$

**Table 3.** Sample Confusion matrix.

|                 |       | Actual Label |      |
|-----------------|-------|--------------|------|
|                 |       | Blast        | Rust |
| Predicted Label | Blast | TB           | FB   |
|                 | Rust  | FR           | TR   |

#### 4. Results

In this section, the authors present the results obtained by evaluating the performance of the trained ‘Custom-Net’ model on the test dataset comprising 990 images of blast and rust diseases in pearl millet.

##### 4.1. Confusion Matrix for Classification

The confusion matrix of the pre-trained ‘Custom-Net’ model on the training and testing datasets are shown in Table 4a,b, respectively. Similarly, the confusion matrix of the non-pre-trained ‘Custom-Net’ model on the training and testing datasets are shown in Table 5a,b, respectively. It is clear from Tables 4a and 5a that the pre-trained, as well as non-pre-trained models, do not misclassify any sample from the training dataset. Whereas, it is evident from Table 4b that the pre-trained ‘Custom-Net’ model misclassifies 34 samples from the test dataset containing 567 images of plant leaves infected with blast disease. Furthermore, it is clear from Table 4b that the ‘Custom-Net’ model misclassifies 69 images from the test dataset comprising 423 images of plant leaves infected with rust disease. However, at the same time, it can be observed in Table 5b that the non-pre-trained ‘Custom-Net’ model misclassifies only 4 and 8 samples from the testing dataset comprising images of leaves infected with blast and rust diseases, respectively.

**Table 4.** Confusion matrix of pre-trained ‘Custom-Net’ model.

| (a) Training dataset |       |              |          |
|----------------------|-------|--------------|----------|
|                      |       | Actual Label |          |
|                      |       | Blast        | Rust     |
| Predicted Label      | Blast | 1375 (TB)    | 0(FB)    |
|                      | Rust  | 0(FR)        | 935 (TR) |
| (b) Testing dataset  |       |              |          |
|                      |       | Actual Label |          |
|                      |       | Blast        | Rust     |
| Predicted Label      | Blast | 533 (TB)     | 34(FB)   |
|                      | Rust  | 69(FR)       | 354(TR)  |

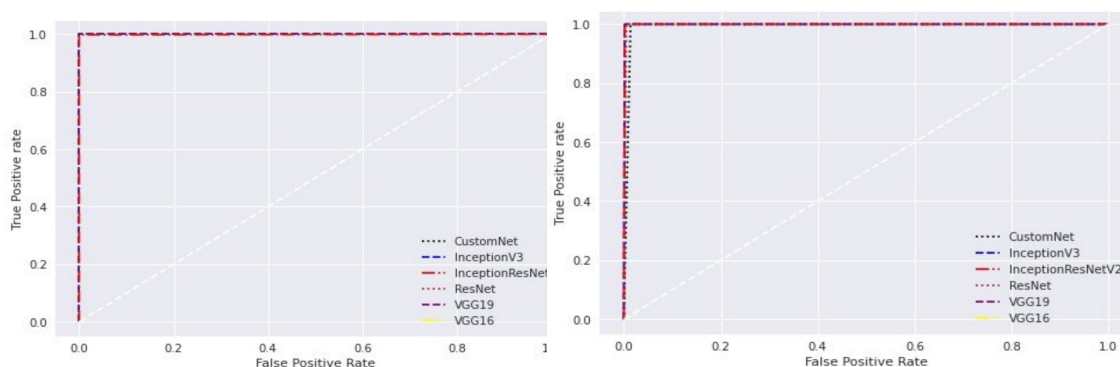
**Table 5.** Confusion matrix of non-pre-trained ‘Custom-Net’ model.

| (a) Training dataset |       |              |      |
|----------------------|-------|--------------|------|
|                      |       | Actual Label |      |
|                      |       | Blast        | Rust |
| Predicted Label      | Blast | 1375         | 0    |
|                      | Rust  | 0            | 935  |

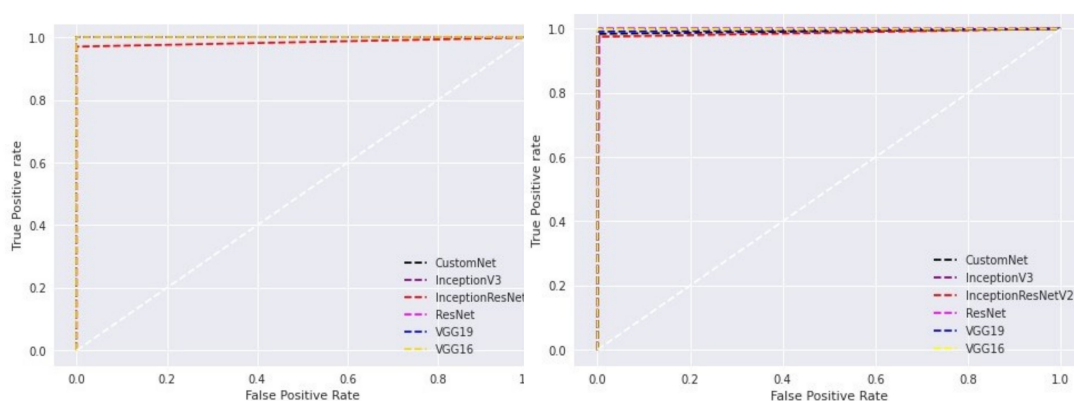
| (b) Testing dataset |       |              |         |
|---------------------|-------|--------------|---------|
|                     |       | Actual Label |         |
|                     |       | Blast        | Rust    |
| Predicted Label     | Blast | 563 (TB)     | 4(FB)   |
|                     | Rust  | 8(FR)        | 415(TR) |

Furthermore, it is claimed in [63] that the area under the curve (AUC) and receiver operating characteristic (ROC) (AUC-ROC) curves are the most effective tools for visualizing the classification performance of a model. In this manuscript, these curves are used to check the capability of the model to distinguish the rust and blast disease classes. The AUC-ROC curves for different classifiers on the training and testing datasets are shown in Figure 6.



(a) AUC-ROC curve for non-pre-trained model on training dataset

(b) AUC-ROC curve for non-pre-trained model on testing dataset



(c) AUC-ROC curve for pre-trained model on training dataset

(d) AUC-ROC curve for pre-trained model on testing dataset

**Figure 6.** AUC-ROC curves of six classifiers on the training and testing datasets.

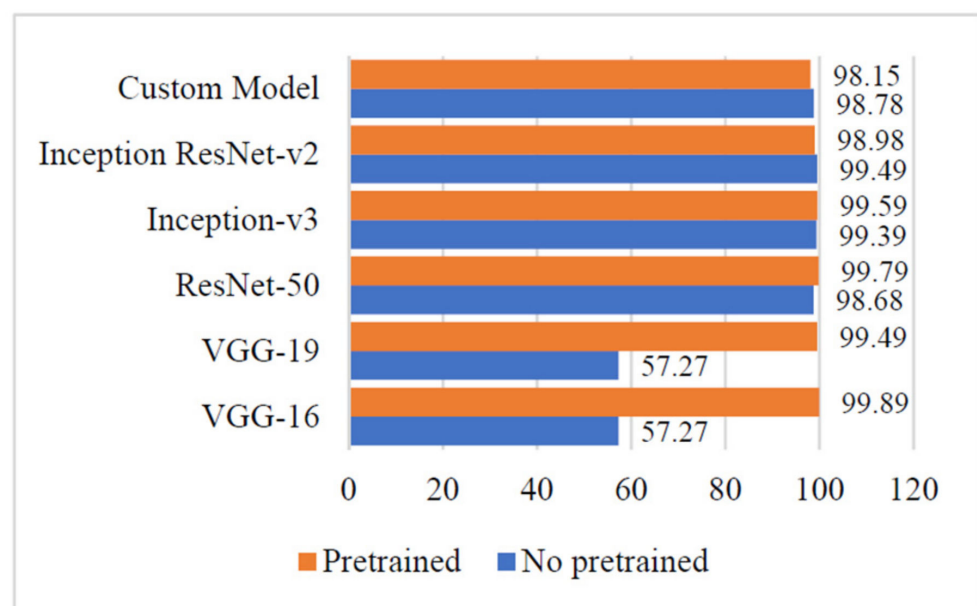
Furthermore, the authors also present the classification performance of the ‘Custom-Net’ model and state-of-the-art DL models, as shown in Table 6. It is evident from the results shown in Table 6 that except VGG-16 and VGG-19 models, the pre-trained and non-pre-trained versions of all the DL models employed in this manuscript report the equivalent values of accuracy, precision, recall, and F1 score.

**Table 6.** Classification performance of different deep learning models.

| Metrics       | Non-Pre-Trained Models |        |           |              |                    |              | Pre-Trained Models |        |           |              |                    |                    |
|---------------|------------------------|--------|-----------|--------------|--------------------|--------------|--------------------|--------|-----------|--------------|--------------------|--------------------|
|               | VGG-16                 | VGG-19 | ResNet-50 | Inception-V3 | Inception ResNetV2 | ‘Custom-Net’ | VGG-16             | VGG-19 | ResNet-50 | Inception-V3 | Inception ResNetV2 | ‘Custom-Net’ Model |
| Accuracy (%)  | 57.27                  | 57.27  | 98.68     | 99.39        | 99.49              | 99.78        | 99.89              | 99.49  | 99.79     | 99.59        | 98.98              | 98.15              |
| Precision (%) | 100                    | 100    | 99.29     | 99.11        | 99.64              | 99.29        | 99.82              | 99.29  | 99.64     | 98.64        | 99.58              | 99.10              |
| Recall (%)    | 57.27                  | 57.27  | 98.42     | 99.82        | 99.47              | 98.59        | 100                | 99.82  | 100       | 99.64        | 99.64              | 98.39              |
| F1 score (%)  | 72.83                  | 72.83  | 98.85     | 99.46        | 99.55              | 98.94        | 99.91              | 99.55  | 99.82     | 99.64        | 99.11              | 98.69              |

#### 4.2. Average Accuracy

Figure 7 shows that the values of average accuracy, reported by the non-pre-trained and pre-trained versions of the ‘Custom-Net’ and state-of-the-art DL models viz. Inception ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19, are comparable except for the non-pre-trained versions of VGG-16 and VGG-19.



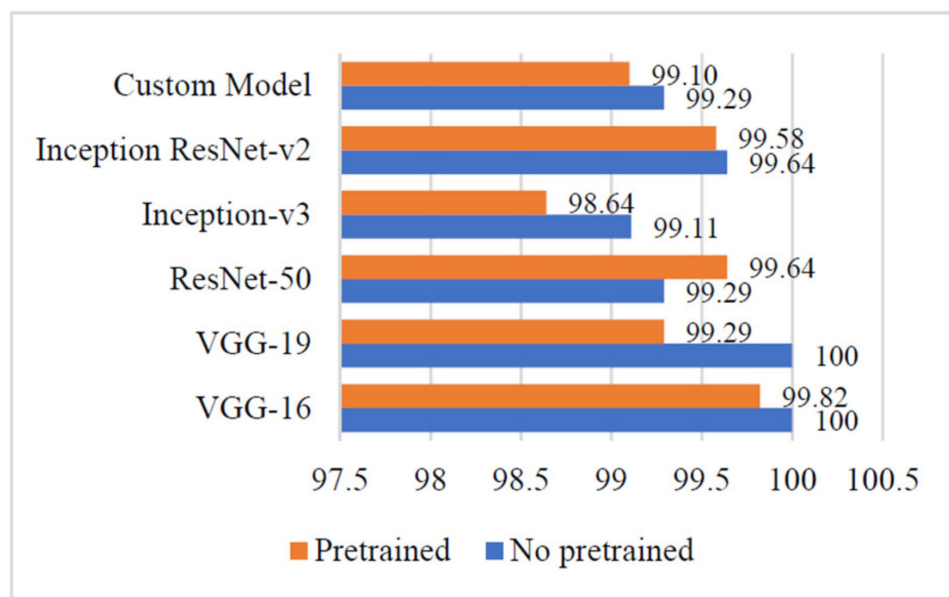
**Figure 7.** Average accuracy of different deep learning models.

The Non-pre-trained version of the ‘Custom-Net’ model reported an average accuracy of 98.78%, whereas its pre-trained version reported an average accuracy of 98.15%.

Similarly, the non-pre-trained versions of Inception ResNet-V2, Inception-V3, and ResNet-50 reported the average accuracies of 99.49%, 99.39%, and 98.68%, respectively. It is also apparent from Figure 7 that the pre-trained versions of ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19 also reported the comparable values of average accuracies of 98.98%, 99.59%, 99.79%, 99.49, and 99.89%, respectively. However, in strong contrast, the non-pre-trained versions VGG-16 and VGG-19 reported a low average accuracy of 57.27%.

#### 4.3. Precision

It is evident from Figure 8 that the ‘Custom-Net’ model reported the highest precision of 99.29%. Moreover, there is a slight difference of 0.19% in the precision of its pre-trained and non-pre-trained versions. It is also clear from Figure 8 that the VGG-16 and VGG-19 models reported the highest precision of 100%. There is a minor variation of 0.18% and 0.71% in the precision of the pre-trained and non-pre-trained versions of VGG-16 and VGG-19, respectively. The other deep learning models viz. Inception ResNet-v2, Inception-v3, and ResNet-50 reported the highest precision of 99.64%, 99.11%, and 99.29%, respectively.



**Figure 8.** Precision of different deep learning models.

#### 4.4. Recall

The results shown in Figure 9 indicate that the ‘Custom-Net’ model reported a recall of 98.59%. A minor variation of 0.20% has been observed in the values of recall of its pre-trained and non-pre-trained versions. Additionally, the results also show that VGG-16 and ResNet-50 reported the highest recall of 100%. Moreover, there is a significant variation of 42.73% and 42.55% in the recall of pre-trained and non-pre-trained versions of VGG-16 and VGG-19, respectively. The Inception ResNet-V2, and Inception-V3 reported the highest values of 99.64% and 99.82%, respectively. Furthermore, there is a minor difference of 0.17% and 0.18% in the recall of the pre-trained and non-pre-trained versions of Inception ResNet-V2 and Inception-V3.

#### 4.5. F1 Score

The experimental results demonstrated in Figure 10 show that the ‘Custom-Net’ model achieved the highest F1 score of 98.94%. It reported a small variation of 0.25% in the F1 score of its pre-trained and non-pre-trained versions. The results shown in Figure 10 also indicate that VGG-16, VGG-19, ResNet-15, Inception-V3, and Inception ResNet-V2, give the highest F1 score values as 99.91%, 99.85%, 99.11%, and 99.64%, respectively. It is also clear from the figure that the VGG-16 and VGG-19 give the highest difference of 27.08% and 26.72%, respectively. There is a minor variation in the F1 score of pre-trained and non-pre-trained versions of ResNet-15, Inception-V3, and Inception ResNet-sV2 models.



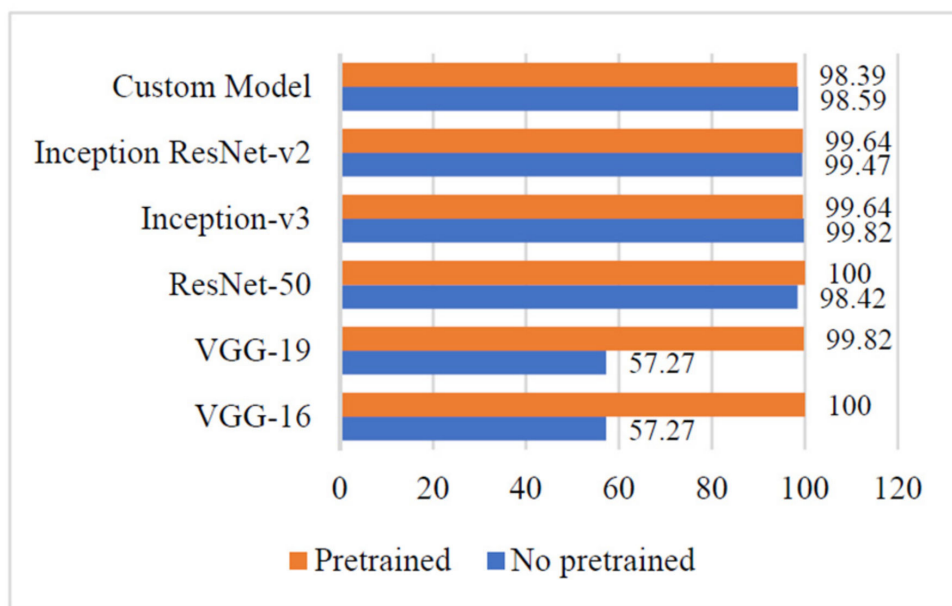


Figure 9. Recall of different deep learning models.

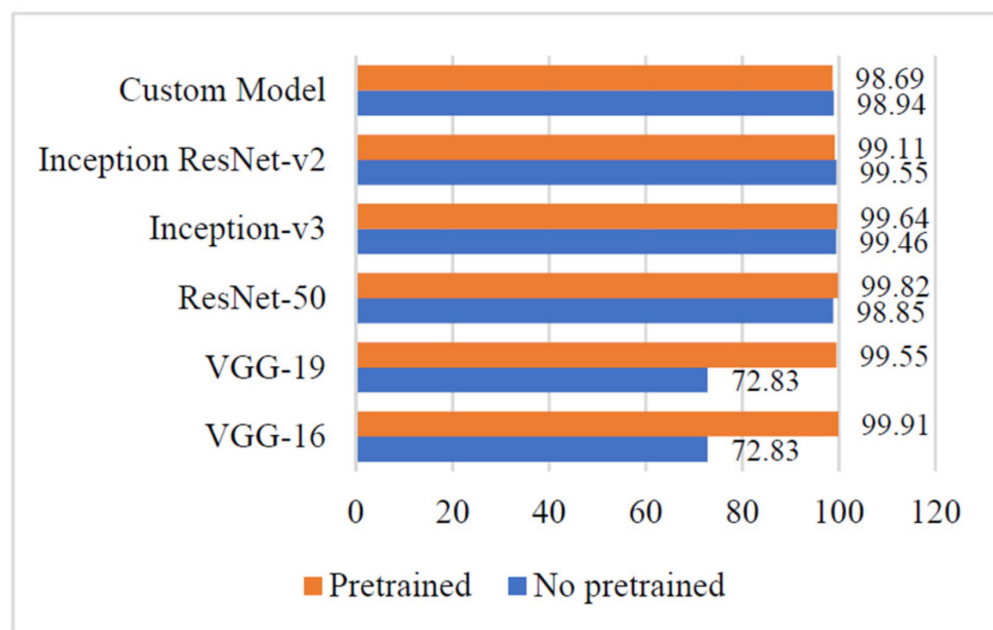


Figure 10. F1 score of different deep learning models.

#### 4.6. Computation Cost

To validate the scope of adopting the proposed ‘Custom-Net’ model and state-of-the-art deep learning models viz. Inception ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19 for classifying diseased leaves, the authors demonstrated the training time and the number of trainable parameters in Figures 11 and 12, respectively. It is noticeable from Figure 12 that the Inception ResNet-V2 model has the maximum number of trainable parameters, whereas the ‘Custom-Net’ model has the minimum number of trainable parameters. Furthermore, it is clear from the training time shown in Figure 11 that the ‘Custom-Net’ model requires a minimum time of only 80 s for training through 20 epochs.

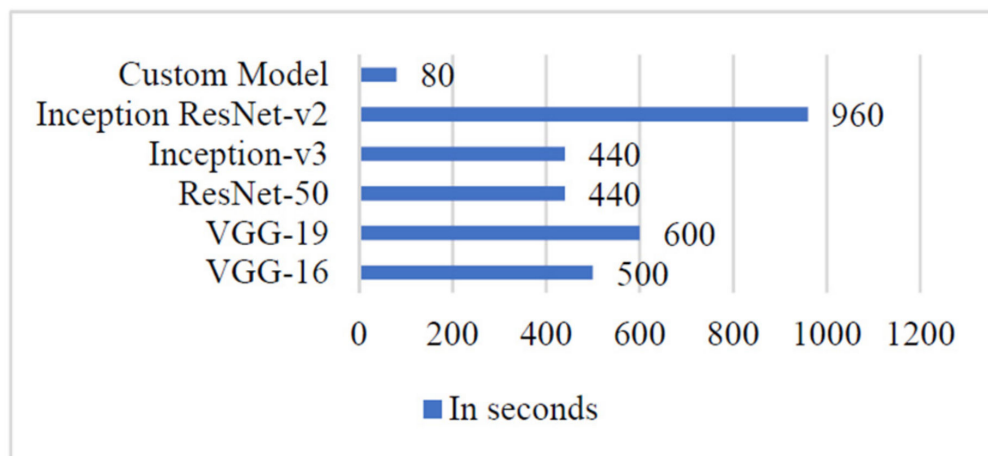


Figure 11. Training time of different deep learning models.

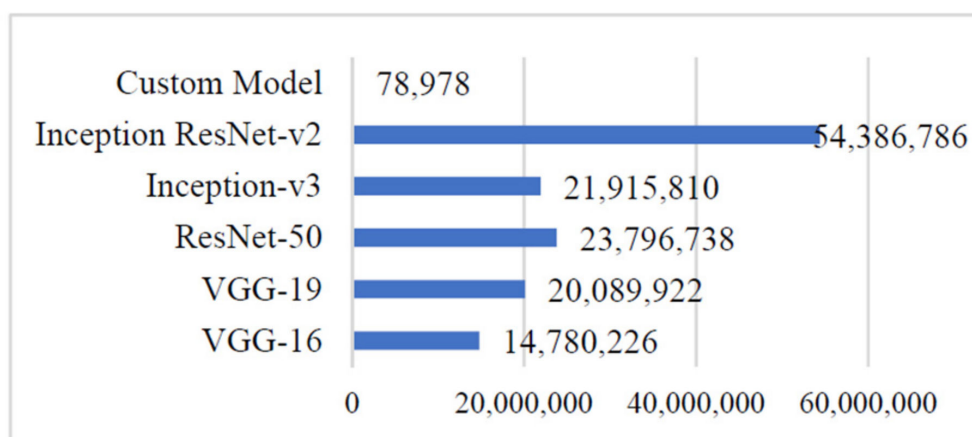


Figure 12. Training parameters of different deep learning models.

#### 4.7. Grad-CAM

Now, the authors plotted the Grad-CAM to visualize the features involved in the classification. The visualization of features involved in classification for the pre-trained and non-pre-trained versions of Inception ResNet-V2, Inception-V3, ResNet-50, VGG-16, and VGG-19 are shown in Figures 13 and 14, respectively.

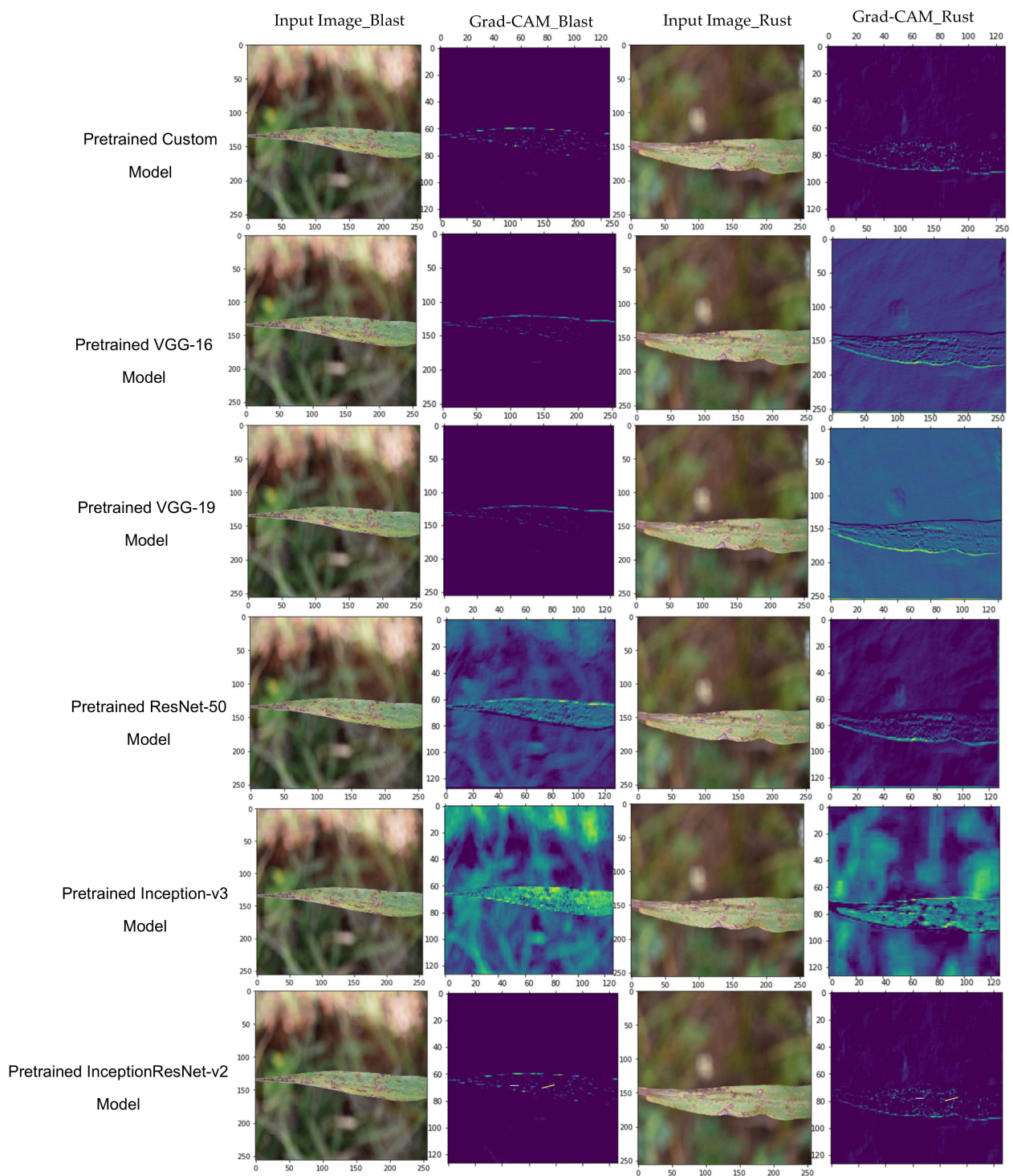


Figure 13. Grad-CAM to visualize the features of pre-trained models.

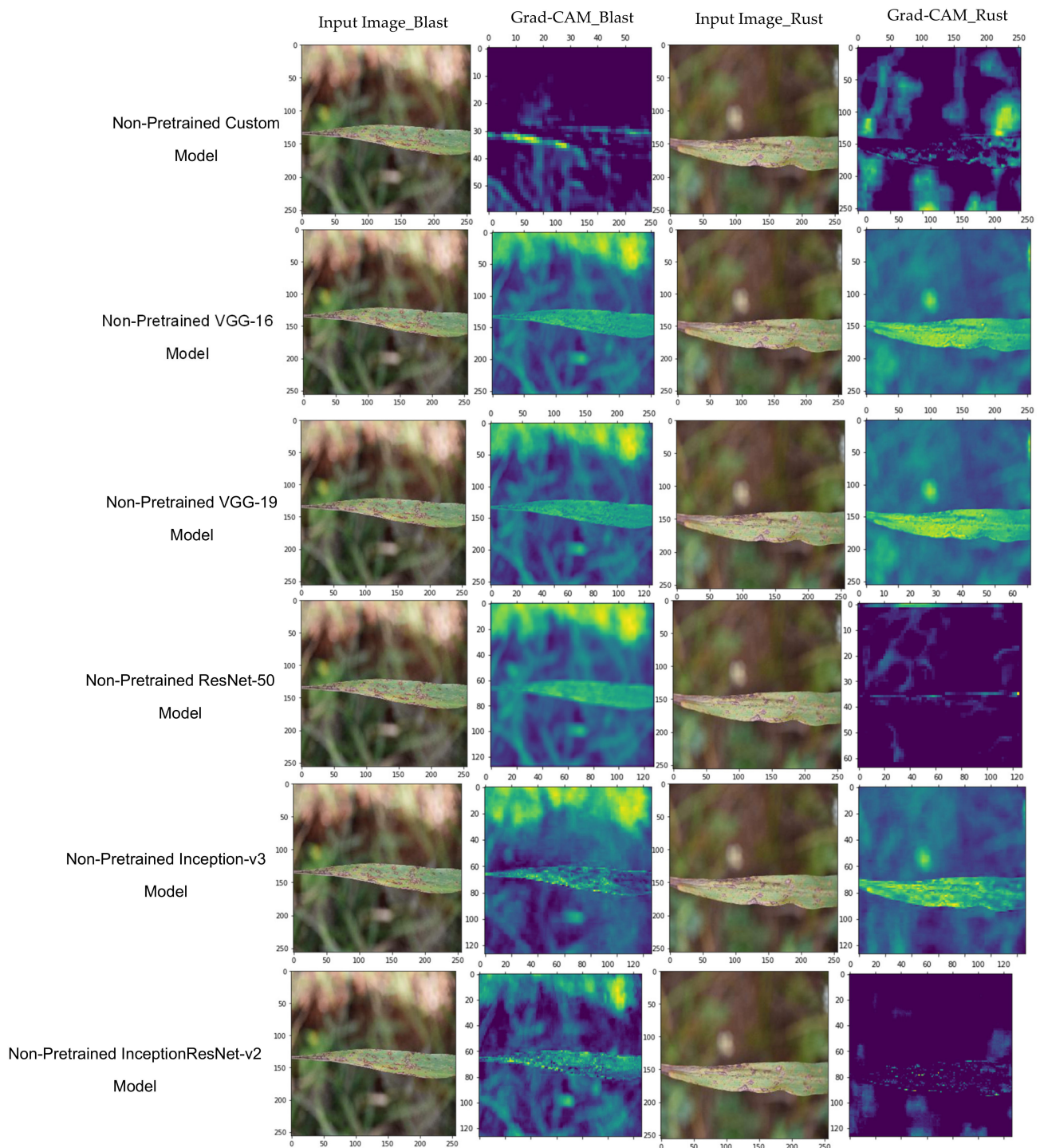


Figure 14. Grad-CAM to visualize the features of non-pre-trained models.

## 5. Discussion

In this section, the authors present the inferences deduced from the experimental results obtained by employing the ‘Custom-Net’, Inception ResNet-v2, Inception-v3, ResNet-50, VGG-16, and VGG-19 models.

It is apparent from Figure 6 that the pre-trained version of VGG-16 gives the highest average accuracy. Whereas, the non-pre-trained versions of VGG-16 and VGG-19 reported the minimum value of average accuracy. The pre-training of these models lead to a significant increase of 42.62% in the average accuracy. This proves that these deep networks



require a vast dataset for training. Therefore, transfer learning becomes vital for the training of these networks if the dataset size is small. By adopting the advantages of transfer learning, these networks learn the low-level and basic features of the dataset, such as boundary recognition and shape identification. Now, the networks use the weights acquired during pre-training and further learn the recognition of high-level features such as sub-boundaries or details about the image segments.

In contrast, with the models viz. Inception ResNet-v2, Inception-v3s, and ResNet-50, a minor impact of transfer learning was reported on the average accuracy. Similarly, a low impact of 0.63% is observed when the 'Custom-Net' model adopted the pre-training and transfer learning.

It is inferred from the above discussion that the shallow neural networks learn the low-level as well high-level features by training on the small dataset size. At the same time, deep networks either require large datasets for training or transfer learning.

Furthermore, the trends of the precision, recall, and F1 measures of the above-stated models are demonstrated in Figures 7–9. It is evident from Figure 7 that the non-pre-trained VGG-16 and VGG-19 models reported the highest precision of 100%. In contrast, the non-pre-trained Inception-V3 model gave the lowest precision of 99.11%. The other non-pre-trained models viz. Inception ResNet-v2, ResNet-50, and 'Custom-Net' reported 0.36%, 0.71, and 0.71% lower precision than the VGG-16 and VGG-19 models. The small variation in the precision of all the non-pre-trained versions of the above-stated models implies that these models are efficient in recognizing the relevant instances of each class from the input test dataset.

The discussion proves that both the pre-trained and non-pre-trained models are efficient in recognizing the relevant instances of each class from the input test dataset. Moreover, the pre-training helps discriminate the relevant and irrelevant features.

A further analysis of the results shown in Figure 8 reveals that the pre-trained VGG-16 and ResNet-50 models reported a 100% recall. The other pre-trained models Inception ResNet-v2, Inception-v3, VGG-19, and 'Custom-Net' reported the 0.61%, 0.36%, 0.18, and 1.61% lower values of recall, respectively.

It is evident from the comparison of non-pre-trained models that the Inception-v3 reported the highest value of 99.82% recall. The other models viz. Inception ResNet-v2, ResNet-50, and 'Custom-Net' also reported the equivalent values of recall, as shown in Figure 8. Moreover, there is a minor difference of 0.17% and 0.18% in the recall of the pre-trained and non-pre-trained versions of Inception ResNet-V2 and Inception-V3. Therefore, the comparable values of recall for all the above-stated models indicate that all the models are efficient in correctly identifying the blast disease from the leaves of pearl millet.

However, the VGG-16 and VGG-19 models gave the lowest values of 57.27% recall. Moreover, there is a significant variation of 42.73% and 42.55% in the recall of pre-trained and non-pre-trained versions of VGG-16 and VGG-19, respectively.

This proves that transfer learning is important for the deeply layered models such as VGG-16 and VGG-19, in order to minimize the number of misclassification of leaves infected by blast disease to the rust class.

Moreover, it is evident from the F1 score shown in Figure 9 that the pre-trained VGG-16 model reported the highest F1 score of 99.91%. Furthermore, the other models viz. Inception ResNet-v2, Inception-v3, VGG-19, and 'Custom-Net' reported the equivalent values of the F1 score.

Simultaneously, it is also observed that the models, viz. 'Custom-Net', ResNet-50, Inception-V3, and Inception ResNet-V2 reported the slight variations of 0.25%, 0.44, 0.18%, and 1.03%, respectively in the F1 score of their pre-trained and non-pre-trained versions. However, the VGG-16 and VGG-19 gave the highest difference of 27.08% and 26.72%, respectively. This proves that transfer learning is important for relevant feature extraction and minimizing the number of misclassifications in VGG-16 and VGG-19 models. However, there is an insignificant impact on the performance of the other above-stated state-of-the-art models. Moreover, the comparable values of the F1 score of all the models reflect that these



models are efficient in correctly identifying the samples of the blast as well as rust diseases from the test dataset.

Furthermore, it is coherent from the Grad-CAM plotted in Figures 10 and 11 that the pre-trained ‘Custom-Net’ model is effective in recognizing the acceptable boundaries from the leaves infected with blast and rust. Therefore, it makes the classification based on the relevant features rather than noise. In contrast, its non-pre-trained version is efficient in identifying all the relevant features. Still, it also picks some features from the background that may increase the number of misclassifications.

Similarly, the pre-trained versions of the above-stated state-of-the-art models also perform as a better feature extraction than their non-pre-trained versions. This proves that transfer learning helps the model in the extraction of more relevant features, recognizing acceptable boundaries, and preventing the involvement of noise in decision-making.

However, the ‘Custom-Net’ model shows comparable values of average accuracy, precision, recall, and F1 score with the state-of-the-art models, but there is a significant decrease in the number of trainable parameters and training time. It is noticeable from Figure 11 that the ‘Custom-Net’ model has the minimum number of trainable parameters. Moreover, it is evident from the training time presented in Figure 12 that the ‘Custom-Net’ model requires a minimum time of 4 s per epoch. It completes its training in merely 80 s through 20 epochs. The analysis of training time shows that it requires 84%, 86.6%, 81.81%, 81.81%, and 91.67% lower training time than VGG-16, VGG-19, ResNet-15, Inception-V3, and Inception ResNet-V2 models, respectively. Its efficacy in achieving the classification accuracy comparable to the state-of-the-art models and low training time proves its usability for real-life systems. Furthermore, it is effective in quick decision-making to classify the blast and rust diseased samples in real-time. Table 7 presents the comparative analysis of the approaches available in literature and the approach proposed in this manuscript.

**Table 7.** Comparison of the proposed approach and the approaches available in literature.

| Reference | Year | Crop          | Diseases  | Number of Images, Source  | Tools Used for Dataset Collection   | Model(s) Applied  | Evaluation Metrics  |
|-----------|------|---------------|---|---|---|---|---|
| Our work  | 2021 | Pearl millet  | Rust, blast   | 3300, ICAR Mysore   | X8-RC Drone camera<br>NIKON D750 Digital camera<br>DHT11 sensor<br>Raspberry Pi | ‘Custom-Net’<br>VGG-16<br>VGG-19<br>ResNet-50<br>Inception-v3<br>Inception<br>ResNet-v2 | Accuracy = 98.78%<br>Precision = 99.29%<br>Recall = 98.59%<br>F1 score = 98.64%<br>Training time = 80 s<br>Number of training parameters = 78,978 |
| [28]      | 2020 | Tomato        | Early blight<br>Late blight<br>Healthy                        | 5923 Plant Village Dataset, Internet images, and leaf images captured from Tansa Farm, Bhiwandi | Sensor  | Support vector machines<br>Random Forest (RF)<br>K-means<br>VGG-16<br>VGG-19            | Clustering accuracy using RF = 99.56%<br>Classification accuracy using VGG-16 = 92.08%  |
| [23]      | 2020 | 59 categories | 49 disease categories, 10 healthy                             | 36,252, AI-challenger   | Video cameras<br>Smartphone   | MDFC-ResNet<br>VGG-19<br>AlexNet<br>ResNet = 50   | Accuracy = 93.96%<br>Precision = 98.22%<br>Recall = 95.40%<br>F1 score = 96.79%   |
| [52]      | 2019 | Pearl millet  | Downy mildew  | 711 Images from the Internet  | No camera<br>No IoT   | VGG-16<br>Transfer learning   | Accuracy = 95%<br>Precision = 94.50%<br>Recall = 90.50%<br>F1 score = 91.75%  |
| [29]      | 2018 | Rice          | Bacterial Blight<br>Sheath Blight<br>Brown Spot<br>Leaf Blast | International Rice Research Institute (IRRI) database   | Drone<br>Camera<br>GPS sensor   | Support vector machine (SVM)  | Only disease boundary detected  |

Moreover, the technique has a biological significance too. The quick and automatic detection of plants infected with rust and blast helps the farmers apply disease control measures, thus preventing the further spread of diseases to the whole farmland.

To further validate the efficacy of the proposed model ‘Custom-Net’, the authors compared its performance with the state-of-the-art models viz. VGG-16, VGG-19, ResNet-15, Inception-V3, and Inception ResNet-V2. The comparison shows that the ‘Custom-Net’ model efficiently extracts relevant features and involves the relevant features in the decision-

making. It achieved the classification performance equivalent to the InceptionResNet-V2. Moreover, it requires a minimum time for training. Therefore, the authors integrated the 'Custom-Net' model in the 'Automatic and Intelligent Data Collector and Classifier'.

In the future, there is a scope of making the predictions based on the parametric dataset collected by the data collector part of the proposed framework. Moreover, there is a need to develop a multi-class classifier to classify the healthy plants infected with Downey mildew, blast, smut, ergot, and rust.

## 6. Conclusions

The framework 'Automatic and Intelligent Data Collector and Classifier' (AIDCC) is proposed in this manuscript for automating the collection of imagery and parametric datasets from the pearl millet farmland, feature visualization, and prediction of blast and rust diseases. The framework is an appropriate integration of IoT and deep learning to analyze imagery and numeric data. The hardware components, such as drone cameras, digital cameras, sensors, etc. are anchored in the pearl millet farmland at ICAR, Mysore, India, to collect data automatically. The 'Custom-Net' model is designed as a part of this research and deployed on the cloud server. This DL model processes the data collected by the data collector and provides real-time prediction for the blast and rust diseases in pearl millet. Moreover, to showcase the impact of transfer learning, the authors pre-trained the proposed model on the online available ImageNet dataset. The pre-trained model is further trained on the dataset of 2310 images of leaves of pearl millet infected with blast and rust. The performance of the pre-trained and non-pre-trained 'Custom-Net' models is evaluated. Based on the visualization of features through Grad-CAM, it is concluded that transfer learning improves the extraction of relevant features and helps the model discard the features picked from the background. At the same time, the slight difference of 0.25% in the F1 score of pre-trained and non-pre-trained 'Custom-Net' models prove that being a shallow network, it is equally efficient in making correct classifications even though the training dataset is small.

Moreover, the authors compared the performance of the pre-trained and non-pre-trained state-of-the-art DL models viz. VGG-19, VGG-16, Inception ResNetV2, Inception V3, and ResNet-50 architectures. Furthermore, the authors implemented these models using transfer learning. They employed the pre-trained models on the ImageNet dataset and further trained them on the dataset collected by the data collector of the framework proposed in this research. However, the pre-trained and fine-tuned VGG-19 model outperformed all the models. It achieved the highest values of 99.39%, 99.82%, 99.11%, and 99.46% for the average accuracy, precision, recall, and F1 score, respectively, on the test dataset comprising 990 images of leaves infected with blast and rust. However, this model requires a training time of 600 s that is 86.67% higher than the 'Custom-Net' model. Moreover, the high number of training parameters of 20,089,922 increases its computation cost. Therefore, the authors deployed the pre-trained and fine-tuned 'Custom-Net' model as a classifier in the framework 'AIDCC'. Therefore, this research provides a low-cost and user-friendly framework for automating the data collection, feature visualization, disease detection, and prediction of blast and rust diseases in pearl millet. As a result, it may prove a significant contribution to the food industry and farmers in order to increase the yield and quality of crop products.

**Author Contributions:** This research specifies below the individual contributions: Conceptualization, N.K., G.R., V.S.D., K.G.; Data curation, N.K., G.R., S.C.N., S.V., Formal analysis, N.K., G.R., S.C.N., S.V., M.F.I., M.W.; Funding acquisition, M.F.I., M.W.; Investigation, N.K., G.R., V.S.D., K.G.; Methodology S.C.N., S.V., M.F.I., M.W.; Project administration, M.F.I., M.W.; Resources, M.F.I., M.W.; Software, N.K., G.R., V.S.D., K.G., S.C.N., S.V.; Supervision, G.R., V.S.D., S.C.N., S.V., M.F.I., M.W.; Validation N.K., G.R., V.S.D., K.G., S.C.N., S.V., M.F.I., M.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors acknowledge contribution to this project from the Rector of the Silesian University of Technology under a proquality grant grant no. 09/020/RGJ21/0007. This work was

supported by the Polish National Agency for Academic Exchange under the Programme PROM International scholarship exchange of PhD candidates and academic staff no. PPI/PRO/2019/1/00051. The authors would like to acknowledge contribution to this research from the National Agency for Academic Exchange of Poland under the Academic International Partnerships program, grant agreement no. PPI/APM/2018/1/00004.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data prepared as a part of this research is available at <https://www.kaggle.com/kalpitu/gupta/blast-and-rust-compressed>. The researchers who wish to use the dataset available at the above link must cite this article.

**Acknowledgments:** The authors acknowledge the contribution to this project from the Rector of Silesian University of Technology under the pro-quality grant for outstanding researchers. The authors are also grateful to ICAR (Indian Council of Agricultural Research), New Delhi, India for providing the financial support to University of Mysore under AICRP-Pearl millet Scheme.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. C. Pearl and M. Improvement, *Pearl Millet News. Project Coordinator ICAR—All India Coordinated Research Project on Pearl Millet*. 2020. Available online: <http://www.aicpmip.res.in/pmnews.html> (accessed on 6 July 2021).
2. Authority, P.B.Y.; Delhi, N.E.W. The Gazette of India. 2018; pp. 1–2. Available online: [https://en.wikipedia.org/wiki/The\\_Gazette\\_of\\_India](https://en.wikipedia.org/wiki/The_Gazette_of_India) (accessed on 6 July 2021).
3. ICRISAT. Millet in Schools by Union Ministry. Available online: <https://www.icrisat.org/indias-millet-makeover-set-to-reach-poor-school-meals/> (accessed on 6 July 2021).
4. Climate Change Impact. Available online: <https://thewire.in/environment/millet-india-food-basket-climate-change> (accessed on 6 July 2021).
5. Jukanti, A.K.; Gowda, C.L.L.; Rai, K.N.; Manga, V.K.; Bhatt, R.K. Crops that feed the world 11. Pearl Millet (*Pennisetum glaucum* L.): An important source of food security, nutrition and health in the arid and semi-arid tropics. *Food Secur.* **2016**, *8*, 307–329. [[CrossRef](#)]
6. Chougule, A.; Jha, V.K.; Mukhopadhyay, D. Using IoT for integrated pest management. In Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA), Prune, India, 22–24 January 2016; pp. 17–22. [[CrossRef](#)]
7. Savary, S.; Bregaglio, S.; Willocquet, L.; Gustafson, D.; Mason-D’Croz, D.; Sparks, A.H.; Castilla, N.; Djurle, A.; Allinne, C.; Sharma, M.; et al. Crop health and its global impacts on the components of food security. *Food Secur.* **2017**, *9*, 311–327. [[CrossRef](#)]
8. Schütz, H.; Jansen, M.; Verhoff, M.A. How to feed the world in 2050. *Arch. Kriminol.* **2011**, *228*, 151–159.
9. Darwin, R. Effects of Greenhouse Gas Emissions on World Agriculture, Food Consumption, and Economic Welfare. *Clim. Chang.* **2004**, *66*, 191–238. [[CrossRef](#)]
10. Park, S.J.; Hong, S.; Kim, D.; Seo, Y.; Hussain, I.; Hur, J.H.; Jin, W. Development of a Real-Time Stroke Detection System for Elderly Drivers Using Quad-Chamber Air Cushion and IoT Devices. *SAE Tech. Pap.* **2018**, *2018*, 1–5. [[CrossRef](#)]
11. Hussain, I.; Park, S.J. HealthSOS: Real-Time Health Monitoring System for Stroke Prognostics. *IEEE Access* **2020**, *8*, 213574–213586. [[CrossRef](#)]
12. Park, S.J.; Hussain, I.; Hong, S.; Kim, D.; Park, H.; Benjamin, H.C.M. Real-time gait monitoring system for consumer stroke prediction service. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020. [[CrossRef](#)]
13. El-Jerjawi, N.S.; Abu-Naser, S.S. Diabetes prediction using artificial neural network. *Int. J. Adv. Sci. Technol.* **2020**, 327–339. [[CrossRef](#)]
14. Oza, M.G.; Rani, G.; Dhaka, V.S. Glaucoma Detection Using Convolutional Neural Networks. In *Handbook of Research on Disease Prediction through Data Analytics and Machine Learning*; IGI Global: Hershey, PA, USA, 2021; pp. 1–7.
15. Rani, G.; Oza, M.G.; Dhaka, V.S.; Pradhan, N.; Verma, S.; Rodrigues, J.J. Applying Deep Learning for Genome Detection of Coronavirus. *Multimed. Syst.* **2021**, 1–12. [[CrossRef](#)]
16. Kundu, N.; Rani, G.; Dhaka, V.S. Machine Learning and IoT based Disease Predictor and Alert Generator System. In Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 11–13 March 2020; pp. 764–769. [[CrossRef](#)]
17. Sinwar, D.; Dhaka, V.S.; Sharma, M.K.; Rani, G. AI-Based Yield Prediction and Smart Irrigation. *Stud. Big Data* **2019**, *2*, 155–180. [[CrossRef](#)]
18. Dyrmann, M.; Karstoft, H.; Midtby, H.S. Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **2016**, *151*, 72–80. [[CrossRef](#)]

19. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [[CrossRef](#)] [[PubMed](#)]
20. Kundu, N.; Rani, G.; Dhaka, V.S. A Comparative Analysis of Deep Learning Models Applied for Disease Classification in Bell Pepper. In Proceedings of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Himachal Pradesh, India, 6–8 November 2020; pp. 243–247.
21. Mohanraj, I.; Ashokumar, K.; Naren, J. Field Monitoring and Automation Using IOT in Agriculture Domain. *Procedia Comput. Sci.* **2016**, *93*, 931–939. [[CrossRef](#)]
22. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
23. Hu, W.-J.; Fan, J.; Du, Y.-X.; Li, B.-S.; Xiong, N.N.; Bekkering, E. MDFC–ResNet: An Agricultural IoT System to Accurately Recognize Crop Diseases. *IEEE Access* **2020**, *8*, 115287–115298. [[CrossRef](#)]
24. Feng, J.Z.B.; Li, G.Z. *DCNN Transfer Learning and Multi-Model Integration for Disease and Weed Identification*; Springer: Singapore, 2019; Volume 2.
25. Nyvall, R.F.; Nyvall, R.F. Diseases of Millet. In *Field Crop Diseases Handbook*; ICAR: Hyderabad, India, 1989; Volume 500030, pp. 265–280.
26. Singh, R.; Singh, G.S. Traditional agriculture: A climate-smart approach for sustainable food production. *Energy Ecol. Environ.* **2017**, *2*, 296–316. [[CrossRef](#)]
27. Brahim, M. *Deep Learning for Plants Diseases*; Springer International Publishing: New York, NY, USA, 2018.
28. Khan, S.; Narvekar, M. Disorder detection of tomato plant(solanum lycopersicum) using IoT and machine learning. *J. Phys.* **2020**, *1432*, 012086. [[CrossRef](#)]
29. Kitpo, N.; Inoue, M. Early rice disease detection and position mapping system using drone and IoT architecture. In Proceedings of the 2018 12th South East Asian Technical University Consortium (SEATUC), Yogyakarta, Indonesia, 12–13 March 2018. [[CrossRef](#)]
30. Thorat, A.; Kumari, S.; Valakunde, N.D. An IoT based smart solution for leaf disease detection. In Proceedings of the 2017 International Conference on Big Data, IoT and Data Science (BIGDATA), Pune, India, 20–22 December 2017. [[CrossRef](#)]
31. Chapaneri, R.; Desai, M.; Goyal, A.; Ghose, S.; Das, S. Plant Disease Detection: A Comprehensive Survey. In Proceedings of the 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, 3–4 April 2020.
32. Abdullahi, H.S.; Sheriff, R.; Mahieddine, F. Convolution neural network in precision agriculture for plant image recognition and classification. In Proceedings of the 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Sao Carlos, Brazil, 16–18 August 2017.
33. Singh, U.P.; Chouhan, S.S.; Jain, S.; Jain, S. Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease. *IEEE Access* **2019**, *7*, 43721–43729. [[CrossRef](#)]
34. Nagasubramanian, K.; Jones, S.; Singh, A.K.; Sarkar, S.; Singh, A.; Ganapathysubramanian, B. Plant disease identification using explainable 3D deep learning on hyperspectral images. *Plant Methods* **2019**, *15*, 1–10. [[CrossRef](#)]
35. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep Learning for Image-Based Cassava Disease Detection. *Front. Plant Sci.* **2017**, *8*, 1852. [[CrossRef](#)]
36. Barbedo, J.G.A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **2019**, *180*, 96–107. [[CrossRef](#)]
37. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 59069–59080. [[CrossRef](#)]
38. Wu, Q.; Zhang, K.; Meng, J. Identification of Soybean Leaf Diseases via Deep Learning. *J. Inst. Eng. (India) Ser. A* **2019**, *100*, 659–666. [[CrossRef](#)]
39. He, L.; Zhang, X.; Ren, S.; Sun, J. Deep Residual learning for image recognition. *arXiv* **2020**, arXiv:1512.03385v1.
40. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
41. Längkvist, M.; Karlsson, L.; Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognit. Lett.* **2014**, *42*, 11–24. [[CrossRef](#)]
42. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826. [[CrossRef](#)]
43. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [[CrossRef](#)]
44. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
45. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
46. Review: Inception-v3—1st Runner Up (Image Classification) in ILSVRC. 2015. Available online: <https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c> (accessed on 12 September 2020).

47. AI Challenger Crop Disease Detection. 2018. Available online: <https://pan.baidu.com/s/1TH9qL7Wded2Qiz03wHTDLw#list/path=%2F> (accessed on 10 December 2020).
48. Ferentinos, K. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [[CrossRef](#)]
49. Shah, J.P.; Prajapati, H.B.; Dabhi, V.K. A survey on detection and classification of rice plant diseases. In Proceedings of the 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), Piscataway, NJ, USA, 10–11 March 2016.
50. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [[CrossRef](#)]
51. Cao, P.; Li, X.; Mao, K.; Lu, F.; Ning, G.; Fang, L.; Pan, Q. A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation. *Biomed. Signal Process. Control* **2020**, *56*, 101675. [[CrossRef](#)]
52. Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Deep neural networks with transfer learning in millet crop images. *Comput. Ind.* **2019**, *108*, 115–120. [[CrossRef](#)]
53. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
54. Shetty, H.S.; Raj, S.N.; Kini, K.R.; Bishnoi, H.R.; Sharma, R.; Rajpurohit, B.S.; Yadav, O.P. Downy Mildew of Pearl Millet and its Management. *Indian Counc. Agric. Res. Mandor Jodhpur* **2016**, *342304*, 55.
55. Garg, D.; Alam, M. Deep learning and IoT for agricultural applications. In *Internet of Things (IoT)*; Springer: Cham, Switzerland, 2020.
56. Mogili, U.R.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [[CrossRef](#)]
57. Chen, C.-J.; Huang, Y.-Y.; Li, Y.-S.; Chang, C.-Y.; Huang, Y.-M. An AIoT Based Smart Agricultural System for Pests Detection. *IEEE Access* **2020**, *8*, 180750–180761. [[CrossRef](#)]
58. Materne, N.; Inoue, M. IoT Monitoring System for Early Detection of Agricultural Pests and Diseases. In Proceedings of the 2018 12th South East Asian Technical University Consortium (SEATUC), Piscataway, NJ, USA, 12–13 March 2018.
59. Thakur, V.R.R. P Screening techniques for pearl millet. *Flexo Tech.* **2008**, *96*, 13–14.
60. Amara, J.; Bouaziz, B.; Algergawy, A. A deep learning-based approach for banana leaf diseases classification. *Lect. Notes Inform.* **2017**, *266*, 79–88.
61. Sharma, P.; Berwal, Y.P.S.; Ghai, W. Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Inf. Process. Agric.* **2020**, *7*, 566–574. [[CrossRef](#)]
62. ImageNet Dataset. Available online: [Image-net.org](http://Image-net.org) (accessed on 2 October 2020).
63. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]