International Journal of Advanced Technology and Engineering Exploration, Vol 11(115)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100083

Research Article

Comparative analysis of potato blight diseases BARI-72 and BARI-73 using a simplified convolutional neural network method

Md. Ashikur Rahman Khan[1*] Jesmin Akther[2] and Fardowsi Rahman[3]
Department of Information and Communication Engineering (ICE), Noakhali Science and Technology University (NSTU), Noakhali 3814, Bangladesh[1]
Department of Computer Science and Engineering (CSE), European University of Bangladesh (EUB), Dhaka, Bangladesh[2]
Department of Computer Science and Engineering (CSE), Primeasia University (PaU), Dhaka, Bangladesh[3]

Received: 23-January-2024; Revised: 25-June-2024; Accepted: 26-June-2024

Abstract

Crop diseases significantly threaten global food security, impacting agricultural productivity and economic stability. These diseases are caused by various pathogens, including fungi, bacteria, viruses, and nematodes, which can infect various plant parts, including leaves, stems, roots, and fruits. Classifying the several crop diseases is the requirement for the prevention of distinct disease problems. However, it is challenging to detect exact crop diseases that cause slight differences among the diseases of the same crop. Meanwhile, multi-layer convolutional neural networks, while effective in daily computer vision tasks, come with drawbacks such as significant computational memory requirements and extended training times. The simplified convolutional neural network (SCNN) model comprises three hidden layers with increasing order in each convolution kernels 16, 16, and 32, reducing the time and space complexity. This study incorporates normalization, dropout, and regulation techniques to accelerate training merging and enhance accuracy. Then, the performance metrics are found, and distinct algorithms are compared to measure the effectiveness of the top-performing model. The investigational comparisons among the projected SCNN model and others revealed that the planned SCNN model offers the uppermost accuracy. Furthermore, the SCNN outcome is applied to actual crop image datasets, achieving a classification accuracy of 95.69%. Above all, the planned SCNN model demonstrates promising results in potato blight disease classification, offering high accuracy while mitigating the computational memory requirements and training time. These findings suggest its potential applicability in real-world agricultural scenarios for efficient crop disease detection and prevention.

Keywords

Neural networks, SCNN, Potato blight disease, Crop diseases, VGG-16, ResNet-18.

1.Introduction

Potatoes are a staple food for millions globally, particularly in regions where they are a primary source of calories and nutrients. Diseases affecting on potato leaf diseases represent a multifaceted challenge with implications ranging from economic stability and food security to environmental sustainability [1]. Addressing these challenges requires a holistic approach that integrates scientific research, technological innovation, and sustainable agricultural practices to ensure the resilience and productivity of potato crops in the face of disease pressure.

The United Nations Food and Agriculture Organization (FAO) statistically reported that potato cultivation spans 157 countries globally, covering a collective lodging zone of 19.46 million hectares and yielding approximately 370 million tons annually [1]. Yet, crop diseases hinder crop production, particularly in emerging nations like Bangladesh. Despite momentous advancements in farming procedures and technology, the risks of crop diseases continue, as well as disheartening efforts to increase yields and ensure food security [2]. The potato industry in the region has experienced substantial growth, reaching a considerable measurement of over 27,811.6 tons between 2013 and 2014, establishing itself as the fourth most extensive crop worldwide. Nevertheless, it confronts significant trials from disorders such as potato tuber moth, brown rot, and vital blight-infected

[*] Author for correspondence

fungicidal, leading to losses ranging from 25% to 57%. Over three years, the agronomy of potatoes has witnessed a decline of 70,000 hectares, with cultivation areas dropping from 4.64 lakh hectares in the fiscal year 2021-22 to 4.55 lakh hectares in 2022-23 and further to 3.93 lakh hectares in 2023-24[3].

Figure 1 visually illustrates the impact of early-indicated blight leaves and late blight on potato plants in this context. The black spots caused by early affected blight are shown in *Figure 1(a)*. Late blight and damage to leaves are depicted in *Figure 1(b)*. Similarly, the Plant-village potato early infected leaf is shown in *Figure 1(c)*, and severe lesion damage on late blight is shown in *Figure(d)*. The same dataset of other crops of tomatoes with early irregular spots is shown in *Figure (e)*, and the severely affected lesion spot is shown in *Figure (f)*.

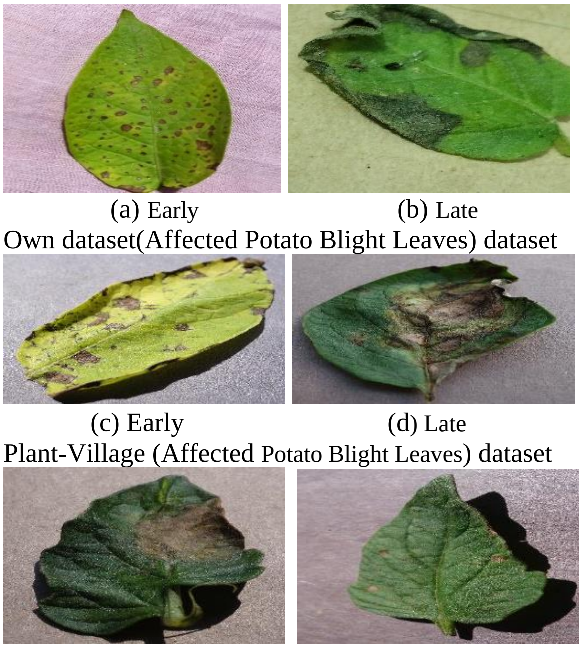

(a) Early (b) Late
Own dataset(Affected Potato Blight Leaves) dataset
(c) Early (d) Late
Plant-Village (Affected Potato Blight Leaves) dataset
(e) Early (f) Late
Plant-village (Affected Tomato Blight Leaves) dataset

Figure 1 Example pictures of Plant-Village dataset and own dataset

Late blight infection in potatoes is caused by the pathogen Phytophthora infestans, a shocking illness on potato foliage and tubers, leading to rapid defoliation and decay [4]. Also, early blight, characterized by small dark spots on mature leaves that range from 3-8 inches and have contoured shapes, is one of these challenges. BARI Alu-73 and BARI Alu-72 (potato) are widely used varieties in Bangladesh and are susceptible to these diseases. In light of these challenges, there is a need for innovative approaches to disease surveillance, prevention, and management in potato cultivation. By addressing the complex interplay of socio-economic, environmental, and technological factors influencing crop health, stakeholders can work towards building a more sustainable and resilient potato industry in Bangladesh and beyond.

Creating a model to detect potato blight is crucial in mitigating its impact on yield levels. Automating this detection process eliminates the need for manual feature extraction, enabling early disease monitoring and prevention. Such a strategy holds significant practical value in improving potato yields, cutting production costs, and boosting overall revenue. The pressing need to address the challenges posed by crop diseases in potato cultivation, particularly in Bangladesh, has motivated the exploration of techniques that can significantly reduce the costs of finding damaged plants early on. Convolutional neural networks (CNNs) have garnered significant interest within the agricultural domain due to their prevalent application in various areas. These include tasks, for example, plant-based detection [5], fruit variety identification [6], disease data diagnosis [7], weed image detections [8], pest data recognition [9], and so on. The models based on CNN are appreciated for their capacity to extract pertinent features from datasets automatically. While deep learning (DL) pre-trained models like visual geometry group networks (VGGNet) [10], AlexNet [11], GoogleNet [12], ResNet [13], and DenseNet [14] have been established for vegetable disease credentials, their great quantity of parameters and computational charge [15] position challenges for implementation on small processor devices such as android, iOS tablet, etc. with limited resources and dataset. Thus, there is a growing need for less timing detection, lower number parameters, and low-power consumption dl applications in agriculture.

A modified version of the CNN architecture was proposed, featuring three hidden layers, to improve the efficiency of feature extraction while reducing computational complexity in diagnosing plant diseases. Under varying conditions, it evaluates the model's effectiveness across diverse datasets, including public repositories and newly gathered data. The proposed architecture abstracts improved features using consecutive three convolutions with batch normalization of 3×3 for each filter size. The vanishing gradient issue was addressed using standard small sequence Convolution, which minimizes the computational complexity and parameter size without impacting performance. The model was practiced using three crop datasets, including its own set, and the robustness of the model performances was evaluated graphically and using metrics. The datasets for tomato and potato plants early consisted of ailments and late blight disease obtained from public storage (Plant-Village repository) dataset used; in this study, images were captured under consistent backgrounds and laboratory setup conditions while photos were taken. A notable role in image background noise and image quality affects model accuracy. It provides a beneficial role, as single channel image conversion from RGB in the image pre-process section is applied. The survey plant dataset was captured in real-time field conditions, and images cover augmentation. This work evaluates the proposed model's performance on three separate datasets against other cutting-edge models. The upshot determines that our recommended simplified convolutional neural network (SCNN) model performs better than other these types of models.

The paper's primary contribution is the model's efficacy, evaluated by confusion matrices predicted and actual values from true (positive, negative) and false (positive, negative) and other matrices among accuracy, recall, precision, and F1-score. This study focuses on pre-trained models, which contain more extensive layers but generate better results. Our top-performing method consistently delivers an accuracy of over 95.69% for all data gathered.

This paper is organized as follows: Section 2 discusses the related work. Section 3 describes the materials and methods. Section 4 presents the experimental findings, while Section 5 offers discussions. Finally, Section 6 provides a detailed conclusion and explores the future prospects of this research.

2.Related works

This section summarizes recent research on documenting crop diseases using deep neural networks (DNNs). DNNs are employed to accurately identify and classify various crop diseases, aiding farmers in timely intervention and management practices.

DL spreads have exhibited encouraging results in spotting and categorizing lesions in digital images, according to Siddiqua et al. (2022) and Wang et al. (2022). These replicas can autonomously grasp image structures, pinpointing even the subtlest disease indicators that conventional image processing techniques might overlook. However, the latest dl requires ample labeled training data and substantial computational resources, posing potential constraints for specific applications [16, 17].

Using both learning from scratch and transfer learning models, Mohanty et al. [8] identified 26 illnesses affecting 14 plant species using AlexNet and GoogleNet. They reached their maximum accuracy with GoogleNet, which was 99.34%. Similarly, these techniques are extensively employed for detecting diseases like greenery blotch, powdery mildew, and rust as indications of abiotic stresses such as drought and nutrient deficiency [18]. Nevertheless, they face constraints in accurately discerning subtle disease symptoms and detecting diseases in their early stages. In addition, they need help in handling intricate and high-resolution images.

Genaev et al. [19] used a you only look once version 4 (YOLOv4)-tiny algorithm to sorting fruit flies by gender in images. The model could identify with an F1 value of 0.838, favorably related to the judgment of skilled crop inspectors. To find 58 plant leaf illnesses, Ferentinos [20] employed five distinct pre-trained methods outcomes on dl which encompass 60M parameters AlexNet, classifier AlexNetOWTBn, VGG and the classifier GoogleNet, the authors stated a prediction accuracy of more than 99%. A deep (9 layers) CNN twisted by Geetharamani and Pandian [21] used AlexNet and GoogLeNet and successfully identified plant diseases 96.46% of the time. Here, the work, parameters, and layer numbers are computationally complex.

Chambon et al. [22] focused on classifying mineral deficiencies in rice, utilizing digital image processing and neural networks with texture and color features. With an accuracy of 88.56%, their approach demonstrates versatility by effectively categorizing blast and brown spot diseases. It highlights the efficacy of their method in addressing multiple disease types, underscoring its potential for broad application in agricultural settings. These studies underscore the potential of digital image processing and learning in diagnosing and managing rice crop health, aiding agricultural efforts.

Ahad et al. reported that the CNN architectures for classifying rice diseases achieves a 98% accuracy in detecting and pinpointing leaf disease locations in rice plants [23]. Another research on CNN achieved an accuracy of 96.09% for potato leaves infected by early and late blight [24] and healthy leaves, and the model can be utilized for Android mobile devices [25].

Additionally, studying rice plants using a VGGNet construction, Chen et al. [26] performed the arrangement of rice plant diseases. Chen et al. [27] also projected a CNN model, namely the MobileNet Beta edition, by covering a pre-build trained MobileNetV2 model to detect plant diseases. Ahmad et al. [28] employed the most extensive dataset-trained neural network architectures, including VGGNet versions, ResNet, and InceptionV3, with InceptionV3 achieving 99.60% accuracy in laboratory settings and 93.70% in field images for tomato leaf recognition. Krishnaswamy and Purushothaman [29] utilized red green blue (RGB) photos, a pre-build VGG16 model, and multi-class support vector machine (SVM) to identify eggplant diseases, achieving a supreme accuracy of 99.4%. Kumar and Domnic [30] enhanced various pre-trained neural network models using DenseNet, a determined testing accuracy of 99.75%. Rangarajan and Raja [31] employed six pre-trained dl architectures to identify ten illnesses across four plant types, with VGG16 obtaining an accuracy value of 90% on the tested images. Acharya et al. [32] utilized InceptionV3 to identify crop illness, noting higher accuracy; they far ahead employed InceptionV3 to identify paddy crop diseases, achieving accuracies of 95.54%. Sethy et al. [33] applied deep features and SVM for diagnosing rice leaf diseases, with the ResNet50 model achieving an f1-score of 98.38%. Shah et al. [34] used dl ResNet50 model with promising accuracy for f1 score 99.75% for rice leaves. Deep residual neural networks outperformed simple CNN in identifying cassava disease, according to Oyewola et al. [35]. Picon et al. [36] work multiple residual neural networks with 50 layers, batch normalization, and rectified linear unit (ReLU) activation to identify wheat diseases, achieving 96% accuracy on real-time field photos. Li et al. [37] diagnosed tomato leaf diseases using a modified CNN model, namely multi-scale local and global feature representation with convolutional neural network (MFRCNN), with accuracy for laboratory and field datasets of 99.01% and 98.75%, respectively. Hu et al. [38] enhanced accuracy to 92.5% by using a modified Cifar10 fast CNN model with depth-wise separable convolution for tea leaf disease identification. To identify various corn and rice diseases, Chen et al. [39] joined the pre-build networks (VGGNet) with an inception layer (INC-VGGN), having average testing accuracies of 92% for rice diseases and 80.38% for corn diseases. On a public dataset at Kaggle Plant-Village dataset, Atila et al. [40] utilized EfficientNet architecture, surpassing traditional CNN models like AlexNet and VGG with an accuracy value of 99.97% with an EfficientNet-B4 networks model. To detect illnesses in corn and apples, Hassan et al. [41] deployed a shallow CNN based on the first few layers of the VGG16 model, achieving an accuracy of 94%. Zeng et al. [42] suggest a self-attention CNN model to imprison meaningful structures from plant disease spots, facilitating the identification of crop diseases. The instances above illustrate the potential of employing RGB images at the leaf level within the computer vision basis for identifying crop diseases. Yet, effectively training models for image recognition demands a specific volume of annotated images captured under diverse conditions, times, and locations to ensure dependable predictions. Annotating images necessitates significant effort according to time and expertise; hence, the need for annotated images poses a constraint on the technique [43]. Moreover, challenges emerge from the worth of the pictures utilized to train the model; subpar resolution or insufficient annotation may diminish accuracy.

Many prior studies encountered challenges with data and methodology, resulting in subpar outcomes. Our research addressed these issues by integrating techniques such as fine-tuning dropout, adjusting batch sizes, and exploring different architectural approaches with trained features, labels, and neurons. Including three hidden layers helped streamline feature extraction complexity while processing data by cropping pictures, which bolstered dataset diversity and simplified complexity. These methodological refinements allowed us to overcome previous hurdles and improve the robustness and effectiveness of our approach.

3.Methodology

The primary focus of SCNN is to advance a programmed and precise network model for predicting potato variety (**BARI Alu-73 and BARI Alu-72**) blight ailments. The experiments took place on Google Colab, utilizing the Keras v3.0.5 and TensorFlow v2.15 libraries, recognized as one of the premier Python v3.7.17 libraries for dl, and facilitated the implementation of dl techniques. Throughout the study, models including original, transfer learning pre-trained model, and using the NVIDIA GeForce RTX 3050 Ti graphics processing unit (GPU) available on Google Colab where GPU consists of 12GB RAM and 360 GB in the cloud storage collaborative services were trained. The SCNN methodology consists of several essential steps, including assembly of survey data, manual and software processing data, pre-processing data, dl techniques, training and testing, performance analysis, and comparative analysis. These steps are essential in achieving the research objective. This section discusses utilizing and implementing various prior-trained network learning-based models for potato variety disease prediction. The entire research process is described in a step-by-step manner. The analysis of the blight disease forecast is summarized in *Figure 2*. Pictures of leaves are gathered, analyzed, and then saved in the researcher's cloud. Before the model is trained, the data is then pre-processed, standardized, and shuffled. After this, the trained model undergoes evaluation and normalization, with the normalized data employed to validate the outcomes. The ultimate selection of the disease prediction model occurs upon achieving satisfactory consequences compared to other models in the training, testing, and validation phases, ensuring no overfitting issues.

3.1Compilation and preparation of data

The initial phase of model development focuses on data collection and analysis. A dataset comprising approximately 4000 images was gathered. The data selected location was explicitly from the Noakhali Science and Technology University Garden, which is positioned in Noakhali, Bangladesh, at 22.7916° N and 91.1028° E. The data acquisition process involved physically placing individual potatoes and capturing images using diverse smart devices, including Samsung Galaxy A7, iPhone version 6 and 10, and Nokia 6, encompassing both platforms between android and iOS. To tackle the difficulty of distinguishing between late and early blight infection during the initial stages of potato disease, the Potato Leaves Virus Dataset was developed in 2022, comprising images of both diseased conditions. The leaves were selected by Ferdousi Begum, an agriculture expert who has been harvesting vegetables since an early age. Jesmin Akther and an expert farmer also cooperated to distinguish and label early and late blight-infected leaves in the information and communication Engineering (ICE) laboratory. In the early blight, leaves exhibited small dark green spots, which were circular to irregular in shape, and leaves became yellow in color [11]. Conversely, late blight leaves retained moisture at the edges or tips, resulting in lesions of up to 3/8 diameter, exhibiting dark brown to black colors and undergoing rapid destruction [24]. These images were taken under the same weather conditions, including consistent shooting distance, lighting, and sunny and gloomy environmental background, ensuring clear visibility of the leaves and their disease spots against the backdrop. Then, manually grouped leaves into training, testing, and validation datasets. Again, each group divided into two, namely early blight and late blight, then began to capture all the leaves. Photos with poor clearness or evident inscriptions were removed by hand to maintain high picture resolution. Additionally, images featuring many pictures were cropped to retain the dataset's emphasis on individual leaves and the disease's physical appearance. The resulting dataset incorporates surveys conducted for potato blight disease, as shown in *Figure 3*.

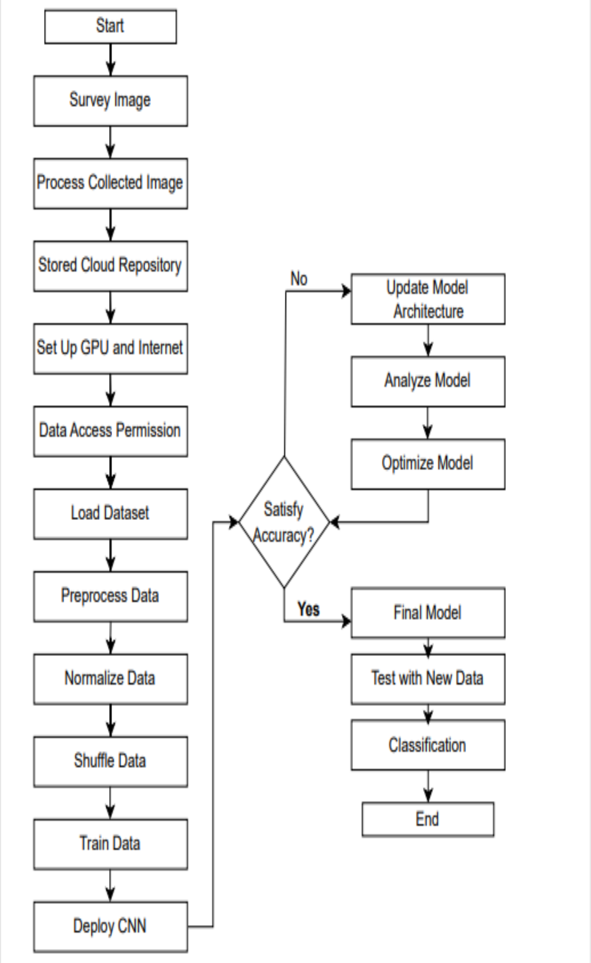


Figure 2 Workflow diagram of the potato blight disease prediction model

Figure 3 Field survey with the manual process of potato leaves

Each device's images are in joint photographic group (JPG) format and belong to the RGB color space. The CNN applied in this training operates with the exact specifications, including dimension, size, feature, and color [14].

Two tools are utilized for processing the raw images: an online web application called bulk-resizephotoes.com and the Microsoft Windows 10 default system application. These tools adjust the images to a standardized size compatible with each model, including our SCNN, size of 230×230, and save them in JPG format without introducing any noise. Due to hardware limitations, this study resizes input image resolutions for all models. Consequently, set the input size for all models to 230 × 230 to ensure uniform evaluation conditions across all models.

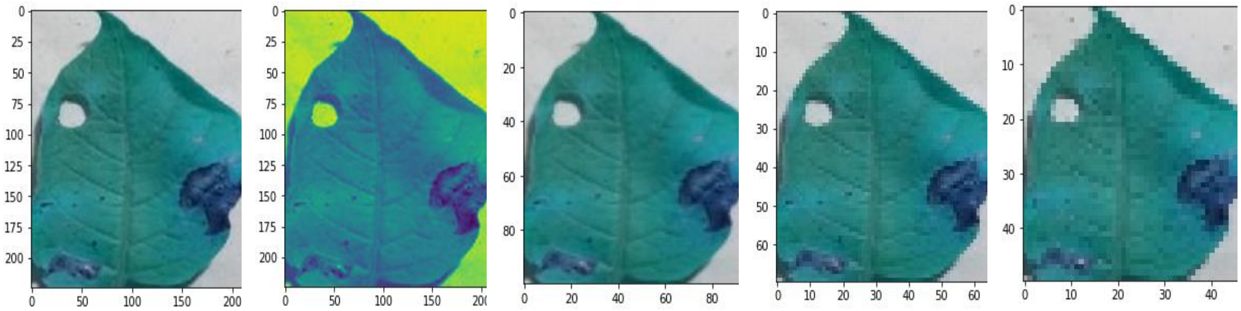
The photos are stored online in cloud storage for data pre-processing and avoiding physical memory. After experimentation, this is determined that Google Collaboratory could support a maximum data size for training the model. GPU support simultaneously saves run time. Two classes of samples are created: one for training and one for testing. Two directories called "Early Blight" and "Late Blight" are formed within these classes. The model is then used to assess the testing results. One thousand early and 1000 late blight photographs make up the training class. Further details on digital image and signal processing can be found by Hong et al. [44].

In the initial stage, the dataset images are plotted and examined for their size, ensuring they are 230× 230 resolution for smooth compilation of data in a short time. Furthermore, make sure that the features of the images are apparent to the naked eye. Following *Figure 4*, it is clear that 100×100 is not clear visually at a time when the original resolution impacts run-time. This procedure is conducted through Python programming in Google Colab to reduce size and maintain data quality. 70×70 size has a better impact than tested other dimensions and original input resolution. Thus, the dataset is converted into an image array of size 70×70 and imported in a monochromatic(single-channel) format utilizing libraries like OpenCV and Matplotlib. It has also been verified that there are no duplicate images within the dataset.

The data is indexed and treated as a directory list of categorical values to commence the training process. Since the data includes all early blight images followed by all late blight images, The data is rearranged using the random library. Kinsley [45] depicts the training data as an array and mixes data by importing a shuffle library to blend the trained data. Imagine a convolutional layer split into ‘p’ groups, resulting in an output of pxn containing channels. At first, we reconfigure the output's channel dimension (px) to a predetermined setup, then transpose it before flattening it, preparing it to be the input for the following layer. The specified index value indicates the validity outcome as either 0 or 1.

Preceding the CNN deployment, the cloud archives the features and labels for each image to train the SCNN model. A Python function is formulated for reshaping the image array; the conversion between grayscale to image array visually plots the image feature to decide the minimum smaller size image array that undermines complexity, where Gray-Data = image array * (-1, 70, 70, 1) as seen in *Figure 4*. The final digit, 1, signifies the grayscale nature of the images to speed up processes which were 3 in input images. Following the reshaping of features, features and labels are normalized per image by dividing each pixel value by 255 by Ramya et al. [46]. A check is performed on the resized image array by comparing the output with 50× 50 and 70× 70 sizes. The resizing of the image array is visually displayed in *Figure 4.*

Following the above *Figure 4(a)* the normal RGB leaves with 230 height and weight, to speed up process convert this *Figure 4(b)* RGB to Gray colour having 230 height and weight, to achieve visual affects with minimum size convert into three different dimension such as in *Figure 4(c), (d)* and *(e)*. Considering both sizes and visualization among these images selected *Figure 4(d)* as the final resize and reshaping data.



(a) is 230×230 RGB is 3 channel

(b) is 230×230 with a color map of Gray

(c) is 100×100 with a color map of Gray

(d) is 70×70 with a color map of Gray

(e) is 50×50 with a color map of Gray

Figure 4 Image channel modification

Achieving optimal performance for the classification algorithm requires data normalization. The minimum(min) and maximum (max) normalization method is utilized, where the min cost of each feature is converted to 0, the max price is converted to 1, and all other values are scaled to a decimal between 0 and 1. Image quality and consistent background affect model beneficial accuracy. We must also not normalize our dataset to avoid losing the information contained within highly valued features that will affect final performance metrics undesirably. Therefore, the data normalization method is expressed as surveys as shown in Equation 1:

$$Xnormalized \ = \frac{x.-\min(x.)}{\max(x.)-\min(x.)} \qquad (1)$$

Xnormalized is represented as the newly normalized cost.

3.2Training, testing and validation

The data is separated among classes, namely training, validation, and testing categories, as outlined in *Table 1*. The training set is essential for acquainting the model with the data's inherent features, encompassing diverse inputs to enable training datasets across various scenarios and ensure adaptability to predict future samples. Simultaneously, the validation set evaluates the model's performance during training, aiding in refining its configurations. After every epoch, the model is trained with the training dataset and evaluated using the validation dataset, preventing overfitting and enhancing its capacity to take a broad view of unused samples. Finally, the test set evaluates the model's post-training performance, providing a final assessment of its accuracy and precision.

Table 1 Percentage of data

Dataset	Portion
Training	70%
Testing	15%
Validation	15%

The research employs various dl techniques, namely Inceptionv1, ResNet18, and VGG16 pre-trained models. The models are trained using 70% of the dataset [24], while data validity is essential for assessing performance without overfitting problems and fine-tuning the model [11]. Dropout regularization and reducing architecture complexity in limited layers increase the model's performance.

3.3Train data

Early blight photos and late blight images make up both classes of training data at this level. The training speed for the early blight class is 1.86 images per second (IT/S), while for the late blight class, the training speed is 2.66 IT/S using the GPU [23]. The GPU capably operated the situation memory for a fast-tracked picture group, warehoused in a frame buffer designed for display output. In this circumstance, the data training was performed on a computer. The configuration is an Intel Core i7 processor 6th generation, 8GB DDR3 RAM, 64-bit Windows 10 operating system, and the device display is 15 inches. In short, the GPU accelerates the image creation process by modifying the memory to facilitate output to the laptop display device, as depicted in *Figure 5*.

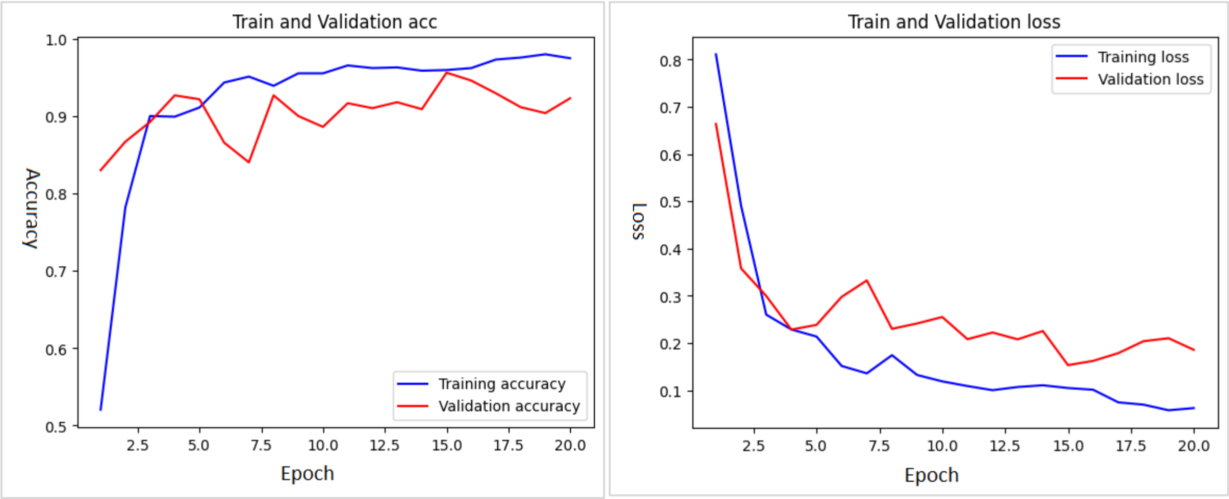
```
100%|██████████|1000/1000 [00:16<00:00,
59.56it/s]
100%|██████████| 1000/1000 [00:17<00:00,
57.51it/s]
```

Figure 5 Train Data time per iteration of each class

The training process involves 59.56 iterations for each class, resulting in a total of 2,000 samples being trained. The Shuffle library is brought in to mix the trained data. The deployment of the CNN utilizes the list of training data. Randomizing the data after training is a critical step for achieving better clarity and accuracy.

3.4Deploy CNN

This approach utilizes the sequential model from Chico and Sallow [47], along with the Adam optimizer for training. The model includes two convolutional layers added with a ReLU activation function each time, taking RGB images as input with a shape of (64, 64, 3). For downsampling, max-pooling layers with a dimension of 2 by 2 are added. Other pooling techniques, min pool and average pool, are fewer fast processes in the work. The feature maps are flattened, and a wholly connected node layer with an activation, specifically ReLU, is introduced. An output layer adding an activation function, SoftMax, is integrated to produce class probabilities. Elements defining the architecture, such as the input shape, number of filters, pooling sizes, and dense layer units, are adjusted based on our specific dataset. After only three epochs, an achieved validation accuracy of 78% is noted. Nevertheless, it is necessary to rerun the CNN deployment, train on 2100 samples, and validate on 900 samples for enhanced validation accuracy. A 30% piece of the data is set aside for validation, while the remaining 70% is used for accuracy testing. It is significant to highlight that additional study is necessary; hence, the model shown here is not the final model. The training values and testing cost outcomes, including accuracy matrices and loss line process, are illustrated in *Figure 6.*



(a) Accuracy Process (b) Loss Process

Figure 6 Accuracy value graph with loss

Figure 6(a) shows validation accuracy is almost accurate with training accuracy, which means training accuracy is above 80% after three epochs. *Figure 6(b)* horizontal axis for epoch and vertical for loss in percent unit. Training loss is blue and red for validation loss, which shows 75%. The more minimalized the loss, the more the data gets validated. Training loss is 40%, and validation loss is nearly 50% after three epochs.

Since the initial accuracy of the model did not meet our expectations, we conducted a thorough analysis to identify areas for improvement and made optimizations to the model.

3.5Deep learning (DL) Techniques

Here, this research work gives the blight disease prediction model sample utilized in the suggested system in the DL techniques section, as depicted in *Figure 2*. Starting the process with internet and GPU maintenance gives authorization for personal cloud storage. The dataset is then loaded into Google Colab for further processing. The dataset is subjected to pre-processing operations like normalization (Decrease the prominence of one characteristic compared to the rest), shuffles, and training. Data is subsequently provided into the DL methods used in this technique. Three pre-trained DL classification models are used for comparative analysis.

A. VGG-16

The University of Oxford's VGG developed the multiple CNN design known as VGG16 Simonyan and Zisserman [14], which consists of 16 layers. It has become well-known for its efficiency and simplicity in picture categorization jobs. The model's input layer accepts photos with the coordinates (224, 224, 3). It is completed in several convolutional stages, combined with convolutional layers and then max pooling. VGG-16 uses ReLU activation, 3×3 size filters per 1 stride, and 13 conv layers are convolved with five pooling(max-pooling) layers. The layers with pooling reduce spatial dimensions while preserving local spatial information. The final fully connected layers accomplish high-level feature extraction and class predictions. VGG-16's strength lies in its deep architecture and consistent design, utilizing stacked smaller filters to capture intricate features. However, its large number of parameters makes it computationally expensive. Nevertheless, VGG-16 has been widely involved and assists as per a benchmark for processor vision tasks, showcasing exceptional performance in image classification and object recognition. The model shows satisfactory outcomes for the new dataset where the complexity lies in time and the number of parameters.

B. ResNet-18

The deep neural network (DNN) ResNet-18 is a shallow modification of the ResNet network architecture introduced by He et al. [48]. It consists of 18 layers and has an impressive presentation in classification(image) tasks. The model architecture includes an input layer that accepts images of shape (224, 224, 3). Four residual blocks are placed after the initial convolutional block, each incorporating two convolutional layers, batch normalization with function ReLU activation. Each residual block's top layer down samples the spatial dimensions. The average pooling layer then receives the feature maps obtained from the final residual block, reducing their spatial dimensions. The output layer, which generates the class predictions, is liable to the class number by a fully networks connected layer. ResNet-18 characteristic is the use of skip or residual connections. These connections enable more accessible training of deep networks by allowing the gradients to flow directly, addressing the problem of vanishing gradients. Despite its relatively shallow architecture, ResNet-18 has confirmed robust performance in various image classification benchmarks. Its simplicity and effectiveness make it a popular choice for tasks where a balance between model depth and computational resources is desired. When we applied the novel dataset, it showed some dataset training faults as more and more layers were added. To mitigate the issues, we added a stack of convolutional layers with a MaxPool layer in the SCNN model; as a result, the computational cost of input became less.

C. Inception v1 or GoogLeNet

Inception v1 is primarily a deep 27-layer CNN worked by Oyewola et al. [35], Lee and Dernoncourt [49]. Convolution, maxpool, two sequential blocks for inception (3a), and inception (3b), followed by another max-pooling, inception(4a) block, another inception(4b), again inception(4c), five sequential blocks for inception(4d), inception(4e). Additionally, there's another max pool, two sequential blocks for inception(5a) block, and inception(5b). The average-pooling layer, dropout (40 percent), linear, and softmax layers are the key features of the model. It was much smaller than the then-prevalent models like VGG and AlexNet. Inception Layer [49] is consisted of all of those layers (11 Layer of Convolutional, 33 Layer of Convolutional, and 55 Convolutional Layer), with their production filter groups cut into a single yield vector that serves as the input proceeding to the subsequent stage. The mentioned layers have two noteworthy additions in the initial inception layer: (1) 11 Convolutional layers primarily used for dimensionality reduction before introducing another layer. (2) A parallel Max Pooling layer broadens the selections available within the inception layer. The error rate in Inceptionv3 was reduced to just 4.2%. Each epoch now takes approximately a quarter of the time compared to VGG16 [11]. In short, Inception networks were proven efficient regarding computational resources and parameter count. Yet, their limitation arises from their restricted flexibility to novel use cases and applied new datasets.

D.Proposed simplistic convolutional neural network (SCNN) approach for identification of potato (bari alu-72,73) leaves blight affects

Figure 7 demonstrates the structure of SCNN for this investigation. It has three consequent convolutional and three pooling layers. The input image set is converted into a trainable three-dimensional (3D) filter set by convolving each convolutional layer, which is important. Pooling layers are employed to lessen the feature map dimension and remove redundant information, which means downsampling.

The network architecture integrates three layers for multiple purposes, including reducing parameter complexity and preventing overfitting. This is one of the most important comparable impacts for SCNN [11,48,49]. Prior to finalizing the model, a few trials and errors confirmed the 3×3 filter size with 16 filters, increasing order dropout regulation0.25, 0.25, 0.25, and 0.5 accurate and fitting features. To add to it, each conv layer stride value 1 following pooling layer two can curb the tallness and thickness of the output from the input images, and padding in conv layer value set 'same' and pooling layer 'valid' adjust the dimensionality of the model data effectively. When inputting data into a trained model, the last stage entails utilizing another fitting technique, like the model.fit() method from the Keras library in Python. Before shuffling the data, this function takes in parameters such as data features and labels, both formatted as NumPy arrays via the NumPy library. A batch size of sixteen samples and a validation split float value of thirty percent is specified, denoting the portion of the training data designated for validation. For the validity check, we separated some images into the validation folder mentioned in section 3.1, which confirms the model's actual validation accuracy. Furthermore, the SCNN model is trained for forty epochs. *Table 2* lists the precise settings for each CNN layer.

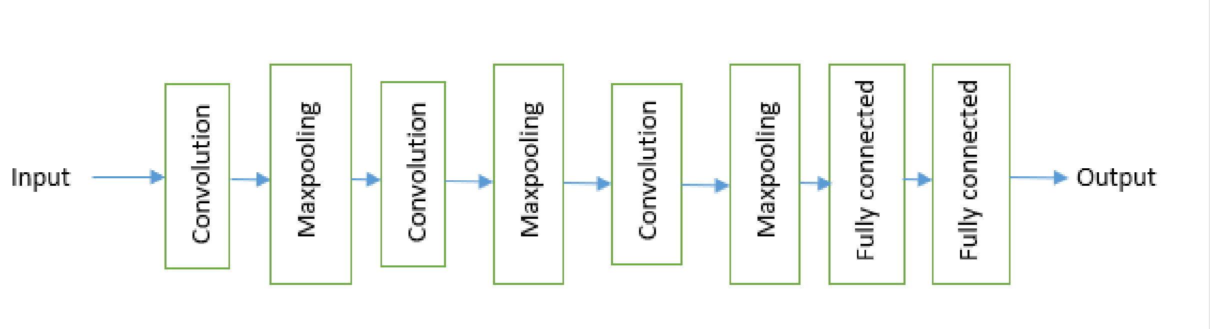


Figure 7 Design of simplistic CNN (SCNN)

Table 2 Parameters and layers that are used in simplistic CNN

Layers	Conv._1	Pool. _1	Conv._2	Pool. _2	Conv._3	Pool. _3	Dense. _1
Filter sizes	3×3	2×2	3×3	2×2	3×3	2×2	
No of filters	16		16		32		
Dropout	0.25		0.25		0.25		0.5
Stride	1	2	1	2	1	2	
Padding	**same**	**valid**	**same**	**valid**	**same**	**valid**	

These layers are designed to adaptively learn discriminative and optimal features. The convolutional layers utilize filters to process localized parts of the input, enabling them to generate strong responses to specific facial features while suppressing others, thereby extracting important local structures.

After each convolutional layer, batch normalization and the ReLU activation function are implemented, which improves performance and speeds up the process. Following this filtering process, down-sampling is performed in the pooling (Maxpooling) layer, enhancing the network's robustness against positional variations. Dropout is then applied after the pooling layers to curb the risk of overfitting the model. Collectively, these layers contribute to the network's efficiency in learning meaningful representations from the input data. In our proposed model, the required parameters amount to 107,601 (as indicated in *Table 3*), whereas a standard CNN model, Too et al. [50], utilizes 1,086,273 parameters (as shown in *Table 4*), indicating a significantly higher parameter count. Our proposed model employs 9.9% fewer hyper-parameters associated with the small CNN architecture.

Table 3 Total parameter required in SCNN

Layer names	Output Map	Params #
Conv2d_1 (conv2d.)	(None. 68, 68, 16.)	160
Activation. (activation)	(None. 68, 68, 16.)	0
Dropout (dropout)	(None. 68, 68, 16.)	0
Max_pooling_2d_1 (max pooling)	(None. 34, 34, 16.)	0
Conv2d_2 (Conv2D.)	(None. 32, 32, 16.)	2320
Activation_1 (Activation)	(None, 32, 32, 16.)	0

Layer names	Output Map	Params #
Dropout_1 (dropout)	(None, 32, 32, 16.)	0
. Max_pooling_2d_2. (max pooling)	(None, 16, 16, 16.)	0
Conv2d_3 (Conv2D)	(None, 14, 14, 32.)	4640
. Activation_2. (Activation)	(None, 14, 14, 32.)	0
Dropout_2(Dropout)	(None, 14, 14, 32.)	0
Max_pooling_2d_3 (max pooling)	(None, 7, 7, 32)	0
Flatten_1 (Flatten)	(None, 12544.)	0
Dense_2(Dense)	(None, 64.)	100416
Dropout_3 (Dropout.)	(None, 64.)	0
Dense_3(Dense.)	(None, 1.)	65
Activation_3 (Activation)	(None, 1.)	0
Total params:		**1,07,601**

Table 4 The parameter required in the original CNN

Layer (Names)	Output Map	Param#
Conv_2d (conv2d)	(None,68,68,64.)	2560
Activation (Activation)	(None.68,68,64.)	0
Max_pooling_2d. (Maxpooling_2d)	(None.34,34,64.)	0
Conv_2d (conv2d.)	(None.32,32,64.)	590080
Activation (Activation)	(None.32,32,64.)	0
Max_pooling_2d (Maxpooling_2d.)	(None.16,16,64.)	0
Flatten (Flatten.)	(None, 65536.)	0
Dense (Dense.)	(None, 64.)	4194368
Dense (Dense.)	(None, 1.)	65
Activation (Activation.)	(None, 1.)	0
Total params:		**1,086,273**

3.6Performance metrics

We estimated our outcomes using various measures, such as accuracy, confusion matrix, precision, recall, and F1-score matrices as shown in Equations 2 to 5.

$$Accuracy \ = \frac{Tp + Tn}{Tp{+}Fp{+}\ Tn{+}\ Fn} \quad (2)$$

$$Precision = \frac{Tp}{Tp{+}Fp} \quad (3)$$

$$Recall \ = \frac{Tp}{Tp{+}Fn} \quad (4)$$

$$F1 - score \ = \frac{2 \times Precision{+}Recall}{Precision{+}Recall} \quad (5)$$

Where Tp, Tn, Fp, and Fn represent True' positive, True' negative, False' positive, and False' negative, respectively.

4.Results

In the paper, emphasize the calculation of SCNN model using different datasets of potato early and late blight [8], as well as other plant disease datasets. The Plant-Village Kaggle dataset [8], a publicly available dataset for crop diseases, is used explicitly in this study. select the potato and tomato blight disease classes from the Plant-Village Kaggle dataset, with 1000 images used for training purposes in each class. It is noteworthy that the data in the Plant-Village dataset exhibit a consistent background and relatively unbroken intensity.

Additionally, introduce our dataset, which includes images captured in multiple backgrounds with leaves present in a single image. Then convert them into grayscale, which results in a faster run. To ensure a fair evaluation, both datasets are arbitrarily Segregated into training and testing sets, adhering to a 70:30 distribution with the help of the validation split method. For consistency, the image sizes are changed to 230×230 pixels. *Figure 1* showcases sample photos from several datasets, while *Table 5* gives comprehensive details on each dataset, including the disease's common and scientific names, its type, and total photographs in each class.

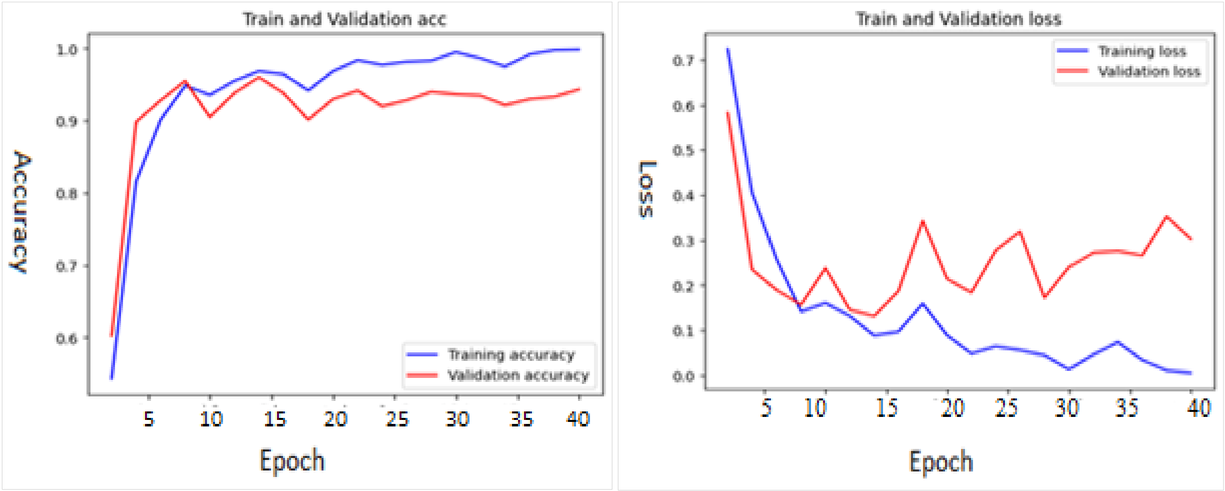
4.1Experimental results

The outcomes of the model are implemented on various datasets, as shown in *Table 6*. On the Kaggle Plant-Village image set (tomato), the model trained for 40 epochs and recorded the training loss peaked at 0.2616, and concurrently, the training accuracy reached 0.8971. The epoch accuracy and validation dataset loss were 0.2429 and 0.9233%, respectively. The highest training loss observed in the Kaggle dataset (potato) was 0.1064, with a trained image accuracy value of 0.9350. Accuracy values with validity loss were 0.1197 and 0.9450%, respectively. On our dataset, the model produced accuracy scores of 95.17% for validation and 99.39% for training, with a validation value loss of 0.2039. These consequences demonstrate the robustness of our dataset, which contains images captured in real-time conditions with intricate backgrounds.

Furthermore, after splitting the dataset into a 70%(train) and 30%(test) ratio for training cost and validation value sets, the build model trained for 40 epochs and evaluated its performance. The accuracy value and cost of loss trends for the SCNN's training and validation on the Plant Village, rice, and cassava datasets, respectively, are shown in *Figure 8* to *10.*

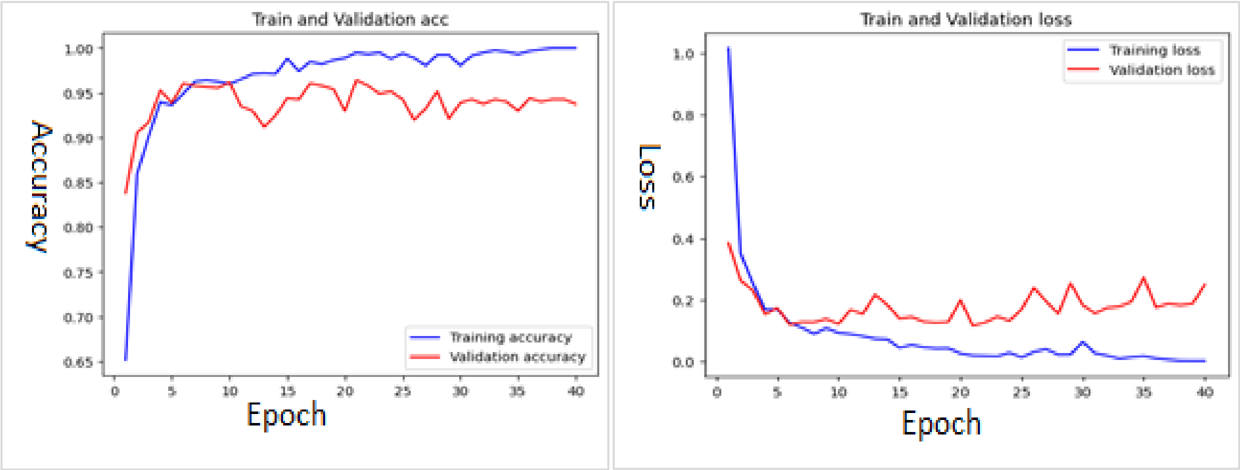
Table 5 Data descriptions for the own surveyed dataset (potato) and Plant-Village's potato crops blight, tomato blight images

Dataset	Plant name	Disease name	Disease Scientific name	No of Images
Plant-Village	Potato	Early Blight	*Alternaria solani*	1000
Plant-Village	Potato	Late Blight	Phytophthora infestans	1000
Plant-Village	Tomato	Early Blight	*Alternaria solan.*	1000
Plant-Village	Tomato	Late Blight	Phytophthora infestans	1909
Own	Potato	Early Blight	*Alternaria solani*	2000
Plant-Village	Potato	Late Blight	Phytophthora infestans	2000



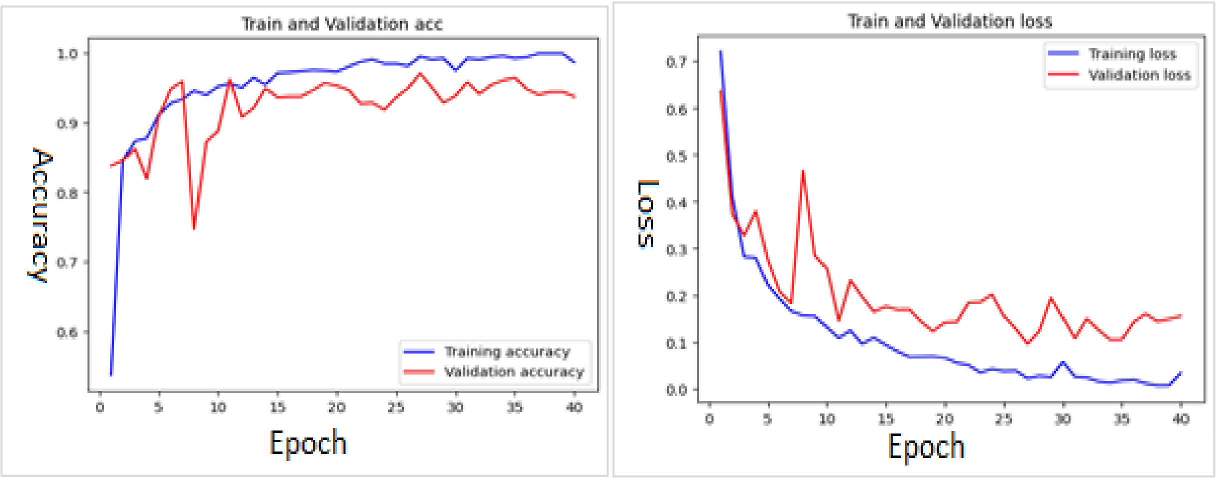
(a)Metrics for trained and validated accuracy (b)Metrics for trained and validated loss

Figure 8 Kaggle Plant-Village (Images of Tomato early and late blight leaves) dataset, (a) Metrics for trained and validated accuracy, (b) Metrics for trained and validated loss



(a) (b)

Figure 9 Kaggle Plant-Village (Images of Potato leaves early and late blight) dataset, (a) Metrics for trained and validated accuracy, (b) Metrics for trained and validated loss



(a) (b)

Figure 10 Own surveyed (Potato leaves early and late blight) dataset, (a) Metrics for trained and validated accuracy, (b) Metrics for trained and validated loss

SCNN build-model indicates its effectiveness despite having fewer parameters, unlike VGG16, ResNet18, and InceptionV1. Additionally, the model contains fewer hidden layers and overfitting issues when considering promising results. Various pre-trained models, including these three, are available for many datasets, but for small datasets like two types of classes, the SCNN model is preparable. The notable work contained in the model is hidden in layer three, with legalization corresponding to .25-.25.5. So, the input image features are not lost, the dimensions of the images are accurate, and fewer argument passes ensure better quality images with better accuracy in a short time comparable to another CNN model underlying pre-trained. The model's generalizability across dissimilar plant diseases beyond potato blight disease included in the work is satisfactory as the early blight that is regularly spotted on the leaves can easily be detected in this model, so the late blight large-size legion can also be detected. The Kaggle dataset confirms this successful detection of the model. Finally, a significant reduction in calculation charge by curbing the sum of parameters by approximately 9.9% [41] compared to standard CNN architectures. The performance of the model is depicted in *Figures 8 to 10.*

Table 6 Summary of the DL methods implementation used

Datasets		**20 epochs**				**40 epochs**				
		Training loss	**Training accuracy**	**Validation Loss**	**Validation accuracy**	**Training loss**	**Training accuracy**	**Validation loss**	**Validation accuracy**	**Test accuracy**
Kaggle Plant -Village	**Tomato blight (Early)**	0.063 0.0633	0.9786 0.9786	0.1993 0.1993	0.8918 0.8918	0.2616 0.2616	0.8971 0.8971	0.2429 0.2429	0.9233 0.9233	0.9220 0.9220
	Tomato blight (Late)									
	Potato blight (Early)	0.0623	0.9775 0.9775	0.1483 0.1483	0.9229 0.9229	0.1064 0.1064	0.9350 0.9350	0.197 0.197	0.9450 0.9450	0.9450 0.9450

	Potato blight (Late)									
Own	Potato blight (Early)	0.0173 0.0173	0.9949 0.9949	0.1339 0.1339	0.9469 0.9469	0.0100 0.0100	0.9939 0.9939	0.2039 0.2039	0.9517 0.9517	0.9517 0.9517
	Potato blight (Late)									

The work evaluated the build model's presentation with the confusion matrix's table attribute. *Table 7* offers standards of test value, accuracy, precision, recall, and F1 score metrics measured from three plant datasets. *Table 7* shows that our own dataset achieved the highest testing accuracy compared to the Plant-Village image set.

Additionally, we separated the dataset into two classes, namely training, and testing, with the test class representing the remaining portion of images. On the basis of this data division, we evaluated how well the model performed using several leave images of the train and test classes. *Table 8* displays how SCNN is presented on each fold. It can be observed from *Table 8* that there is minimal deviation in performance across each dataset. The accuracy in the Plant-Village tomato and potato datasets ranged from 91.83% to 92.01% and 94.17% to 94.33%, respectively. For our dataset, the accuracy ranged from 94.89% to 95.12%. These findings show that the suggested CNN technique performs well over a range of data splits and demonstrates remarkable stability in disease detection.

Table 7 Performance indicator for the proposed model when applied to test pictures

Datasets		**Accuracy**	**Recall**	**Precision**	**F1-score**
Plant - Village	**Tomato blight**	92.33	92.34	92.33	92.37
	Potato blight	94.50%	94.09	93.27	94.18
Own	**Potato blight**	95.69	95.63	95.03	94.91

Using k-fold (4-fold) cross-validation, we incorporated it into our approach to assessing the stability of the SCNN model by analyzing how well it performed on the illness datasets. The datasets were split into five equal portions, five utilized as test sets and the remaining as training sets. This gave us the opportunity to evaluate the model's effectiveness using various combinations of training and testing photos. The performance outcomes for individual folds of the SCNN are outlined in *Table 8.*

In *Table 8*, it is evident that the performance of each pair of datasets exhibits only minor variations.032 The accuracy ranged from 91.79% to 92.27% for the Plant-Village tomato blight dataset, from 94.02% to 94.42% for the Plant-Village potato blight dataset, and from 94.83% to 95.09% for our own dataset. These findings show that our suggested CNN technique demonstrates strong illness identification stability, as the model consistently outperforms other training and testing data combinations.

Table 8 The proposed SCNN's k-fold cross-validation-based output

Number of Fold	**Plant-Village dataset**		**Own dataset**
	Tomato blight	**Potato blight**	**Potato blight**
	Accuracy		
1	0.9183	0.9417	0.9489
2	0.9189	0.9429	0.9495
3	0.9190	0.9437	0.9509

Number of Fold	Plant-Village dataset		Own dataset
	Tomato blight	**Potato blight**	**Potato blight**
4	0.9201	0.9433	0.9512
Average	**0.9193**	**0.9429**	**0.9501**

4.2Evaluation of performance against a trained network

On the survey datasets, evaluated a number of largest neural networks trained models namely VGG16, InceptionV1, and ResNet18. Then, contrasted their performance with that of SCNN model in relations of data training duration with accuracy. *Table 9* displays the findings of this performance comparison.

The percentage of correctly identified classes in the test image set is what accuracy means. It is indistinct from the routine results that our suggested SCNN, which only uses 1,07,601 parameters, performs admirably on our prepared datasets.

Table 9 Performance evaluation using a trained network

Models	Vgg16	ResNet18	Inceptionv1	Proposed SCNN	Dataset	Epoch
Parameter	138,423,208	11,511,784	6,600,000	1,985,665	Own	40
Accuracy	94.65%	94.98%	93.27%	95.17%		
Training time (epoch/second)	104	162	268	59.56		

Table 9 also provides the training time required per epoch for each model on surveyed datasets. It is evident that SCNN model requires significantly less training time compared to the various pre-existing networks. This is applied to the lower number of layers utilized in our SCNN model. The existing trained dl models have lower performance accuracies because they rely on earlier trained weights obtained from the ImageNet data. As a result, these models did not achieve optimal results when applied to our specific datasets. In divergence, our proposed model leverages various convolutional layers with different filter sizes (16, 16, 32) to extract better features, reducing computational complexity and improving model speed. Additionally, dropout is used to mitigate overfitting, while batch normalization enhances the SCNN performance. Thus, compared to pre-trained models, the training time needed for our suggested model is much less. *Table 10* presents a comparison of the precision of the fed dataset in the model with earlier methodologies.

Table 10 Comparison of accuracy with relevant background studies

List of authors	Approaches	Used imagesets	Best accuracy
Chambon et al. (2021) [22]	train a DNN with a mini-batch stochastic gradient descent (SGD)	Self-data	88.56%
Picon et al. in (2019) [36]	deep residual network (DRN)	Imagenet-ILSVRC15.	84%
Hu et al. (2020). [38]	modified Cifar10 fast CNN	Self-data	92.5%
Chen et al. (2020) [39]	Inception-Visual Geometry Group Network (INC-VGGN)	Fujian Institute of Subtropical Botany, Xiamen, China.	92%
Hassan et al. (2021) [41]	Shallow CNN.	Corn and Plant-Village	94%
Ma et al. in (2018) [51]	Deep Convolutional neural networks	Self-data	93.4%
Li et al. (2020) [52]	Shallow CNN	Plant-Village	94%
Barbedo in (2018) [53]	GoogLeNet model	Digipathos.	80.75%
Gandhi et al. in [54]	CNN	Plant-Village	92%
Our work	SCNN	Own	95.17%

5.Discussion

This study proposes a novel SCNN model consisting of three hidden layers, demonstrating effective disease classification in plants. The Plant-Village Kaggle dataset and own dataset were used to construct a CNN model to perceive plant illnesses, including early and late blight on potatoes. The model's performance has also been evaluated through a number of test cases, and the findings for the detection of blight infections have been satisfactory. The model's efficacy across various datasets and training epochs is shown via evaluation measures, which include accuracy, training data loss, and validation data loss. Significant results highlight the model's reliable performance and excellent accuracy in identifying potato blight pathogens. The implications point to practical uses for early illness identification and control in agriculture. Recognizing one's limitations entails recognizing issues with generalizability and other plant diseases or variations in environmental conditions not captured in the datasets used. The reliance on image-based data may raise trials in scenarios where full ground truth labelling is unavailable or where disease symptoms manifest differently. Additional validation and optimization are advised, and comparative analysis demonstrates the model's advantages over current techniques. The study offers insightful information about enhancing crop management techniques and reducing agricultural losses via CNN-based disease detection. Above all, an Android application based on this research is currently being developed, which enables several agricultural specialties like weed and pest identification. There is also an interest in investigating how well the suggested model performs when applied to multiple datasets of plant diseases that contain a variety of photos from various geographical locations.

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion

DL, combined with the internet, encompasses methods within a prevailing domain for accurate disease identification among real-time android or other platforms. In this study, we put forward a unique SCNN model consisting of three hidden layers, demonstrating effective disease classification in plants. According to experimental results, our proposed approach achieves improved accuracy in disease identification. Using two separate plant datasets from Plant-Village, we assess how robust the model is. For the pathogens that cause potato and tomato blight, the suggested model achieves testing accuracies of 94.50% and 92.33%, respectively. Furthermore, we have created a dataset exclusively for developing the suggested model and achieved a 95.17% accuracy rate using this dataset.

Acknowledgment

We thank the entities at the NSTU Laboratory of the ICE Section for their continuous support in successfully completing the task. During the leaves collection, the severity of leaves was initially prepared by, Ferdousi Begum, an expert in crop harvesting. Subsequently, we checked for unlabelled leaves, incorrect labels, and inconsistencies in labeling style. Detailed labelling rules, crucial for determining the pre-processing stage of potato leaves, were established at the ICE lab. Lastly, we sincerely thank the Research Cell at NSTU for their valuable monetary care.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The data considered in this study were gathered from the Noakhali Science and Technology University Garden, Noakhali, Bangladesh. The data are not freely accessible. Nevertheless, the data may be on condition that by the first author upon realistic appeal.

Author's contribution statement

Md. Ashikur Rahman Khan: conceptualization at a time investigation, data curation, suggested draft writing, and reviewing and editing. **Jesmin Akther:** The tasks encompassed gathering data, idealization leading to composing the initial draft manuscript, training the process data, and analyzing and interpreting outcomes. **Fardowsi Rahman:** Involved in problem analysis, designing the model, preparing the draft manuscript, and participating in the review and editing of the writing.

References

[1] Sun DX, Shi MF, Wang Y, Chen XP, Liu YH, Zhang JL, et al. Effects of partial substitution of chemical fertilizers with organic fertilizers on potato agronomic traits, yield and quality. Journal of Gansu Agricultural University. 2023:20230113-1117.

[2] Herrera JP, Rabezara JY, Ravelomanantsoa NA, Metz M, France C, Owens A, et al. Food insecurity related to agricultural practices and household characteristics in rural communities of northeast Madagascar. Food Security. 2021; 13(6):1393-405.

[3] https://www.thedailystar.net/backpage/potato-production-in-bangladesh-glut-badly-hurts-growers-1748572 Accessed 10 June 2023.

[4] https://en.somoynews.tv/news/2024-01-25/potatoes-cost-200-more-than-production-cost. Accessed 29 April 2024.

[5] Feng J, Hou B, Yu C, Yang H, Wang C, Shi X, et al. Research and validation of potato late blight detection

method based on deep learning. Agronomy. 2023; 13(6):1-22.

[6] Shoaib M, Shah B, Ei-sappagh S, Ali A, Ullah A, Alenezi F, et al. An advanced deep learning models-based plant disease detection: a review of recent research. Frontiers in Plant Science. 2023; 14:1158933.

[7] Farjon G, Krikeb O, Hillel AB, Alchanatis V. Detection and counting of flowers on apple trees for better chemical thinning decisions. Precision Agriculture. 2020; 21:503-21.

[8] Mohanty SP, Hughes DP, Salathé M. Using deep learning for image-based plant disease detection. Frontiers in Plant Science. 2016; 7:1419.

[9] Trong VH, Gwang-hyun Y, Vu DT, Jin-young K. Late fusion of multimodal deep neural networks for weeds classification. Computers and Electronics in Agriculture. 2020; 175:105506.

[10] Ren F, Liu W, Wu G. Feature reuse residual networks for insect pest recognition. IEEE Access. 2019; 7:122758-68.

[11] Akther J, Harun-or-roshid M, Nayan AA, Kibria MG. Transfer learning on VGG16 for the classification of potato leaves infected by blight diseases. In emerging technology in computing, communication and electronics 2021(pp. 1-5). IEEE.

[12] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012.

[13] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In proceedings of the conference on computer vision and pattern recognition 2015 (pp. 1-9). IEEE.

[14] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations. 2015:1-14.

[15] Suarez BMJ, Gomez AL, Diaz JE. Supervised learning-based image classification for the detection of late blight in potato crops. Applied Sciences. 2022; 12(18):1-17.

[16] Siddiqua A, Kabir MA, Ferdous T, Ali IB, Weston LA. Evaluating plant disease detection mobile applications: quality and limitations. Agronomy. 2022; 12(8):1-25.

[17] Wang B. Identification of crop diseases and insect pests based on deep learning. Scientific Programming. 2022; 2022(1):9179998.

[18] Genaev MA, Skolotneva ES, Gultyaeva EI, Orlova EA, Bechtold NP, Afonnikov DA. Image-based wheat fungi diseases identification by deep learning. Plants. 2021; 10(8):1-21.

[19] Genaev MA, Komyshev EG, Shishkina OD, Adonyeva NV, Karpova EK, Gruntenko NE, et al. Classification of fruit flies by gender in images using smartphones and the YOLOv4-tiny neural network. Mathematics. 2022; 10(3):1-19.

[20] Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. 2018; 145:311-8.

[21] Geetharamani G, Pandian A. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering. 2019; 76:323-38.

[22] Chambon S, Guillaumin J, Montero L, Nikulin Y, Wambergue P, Fillard P. When high-performing models behave poorly in practice: periodic sampling can help. In the tenth international conference on learning representations 2021 (pp. 1-18).

[23] Ahad MT, Li Y, Song B, Bhuiyan T. Comparison of CNN-based deep learning architectures for rice diseases classification. Artificial Intelligence in Agriculture. 2023; 9:22-35.

[24] Akther J, Nayan AA, Harun-Or-Roshid M. Potato leaves blight disease recognition and categorization using deep learning. Engineering Journal. 2023; 27(9):27-38.

[25] Akther J, Harun-or-roshid M, Mahbub-or-rashid M, Soheli SJ. Bangladesh travel guide (BTG) an android mobile application to utilize free time in a better way. Journal of Advance Research in Mobile Computing. 2021; 3(1):1-10.

[26] Chen J, Zhang D, Zeb A, Nanehkaran YA. Identification of rice plant diseases using lightweight attention networks. Expert Systems with Applications. 2021; 169:114514.

[27] Chen J, Zhang D, Nanehkaran YA. Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimedia Tools and Applications. 2020; 79:31497-515.

[28] Ahmad I, Hamid M, Yousaf S, Shah ST, Ahmad MO. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity. 2020; 2020(1):8812019.

[29] Krishnaswamy RA, Purushothaman R. Disease classification in eggplant using pre-trained VGG16 and MSVM. Scientific Reports. 2020; 10(1):2322.

[30] Kumar JP, Domnic S. Image based leaf segmentation and counting in rosette plants. Information Processing in Agriculture. 2019; 6(2):233-46.

[31] Rangarajan AK, Raja P. Automated disease classification in (selected) agricultural crops using transfer learning. Automatika: a Magazine for Automation, Measurement, Electronics, Computing and Communications. 2020; 61(2):260-72.

[32] Acharya A, Muvvala A, Gawali S, Dhopavkar R, Kadam R, Harsola A. Plant disease detection for paddy crop using ensemble of CNNs. In international conference for innovation in technology 2020 (pp. 1-6). IEEE.

[33] Sethy PK, Barpanda NK, Rath AK, Behera SK. Deep feature based rice leaf disease identification using support vector machine. Computers and Electronics in Agriculture. 2020; 175:105527.

[34] Shah SR, Qadri S, Bibi H, Shah SM, Sharif MI, Marinello F. Comparing inception V3, VGG 16, VGG 19, CNN, and ResNet 50: a case study on early detection of a rice disease. Agronomy. 2023; 13(6):1-13.

[35] Oyewola DO, Dada EG, Misra S, Damaševičius R. Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. PeerJ Computer Science. 2021; 7: e352.

[36] Picon A, Alvarez-gila A, Seitz M, Ortiz-barredo A, Echazarra J, Johannes A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. Computers and Electronics in Agriculture. 2019; 161:280-90.

[37] Li P, Zhong N, Dong W, Zhang M, Yang D. Identifiction of tomato leaf diseases using convolutional neural network with multi-scale and feature reuse. International Journal of Agricultural and Biological Engineering. 2024; 16(6):226-35.

[38] Hu G, Yang X, Zhang Y, Wan M. Identification of tea leaf diseases by using an improved deep convolutional neural network. Sustainable Computing: Informatics and Systems. 2019; 24:100353.

[39] Chen J, Chen J, Zhang D, Sun Y, Nanehkaran YA. Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture. 2020; 173:105393.

[40] Atila Ü, Uçar M, Akyol K, Uçar E. Plant leaf disease classification using efficientnet deep learning model. Ecological Informatics. 2021; 61:101182.

[41] Hassan SM, Jasinski M, Leonowicz Z, Jasinska E, Maji AK. Plant disease identification using shallow convolutional neural network. Agronomy. 2021; 11(12):1-20.

[42] Zeng W, Li M. Crop leaf disease recognition based on self-attention convolutional neural network. Computers and Electronics in Agriculture. 2020; 172:105341.

[43] Oztel I, Yolcu OG, Sahin VH. Deep learning-based skin diseases classification using smartphones. Advanced Intelligent Systems. 2023; 5(12):2300211.

[44] Hong NM, Thanh NC. Distance-based mean filter for image denoising. In proceedings of the 4th international conference on machine learning and soft computing 2020 (pp. 98-102). ACM.

[45] https://pythonprogramming.net/loading-custom-data-deep-learning-python-tensorflow-keras/. Accessed 27 February 2022.

[46] Ramya R, Anandh A, Muthulakshmi K, Venkatesh S. Gender recognition from facial images using multichannel deep learning framework. In machine learning for biometrics 2022 (pp. 105-28). Academic Press.

[47] Chicho BT, Sallow AB. A comprehensive survey of deep learning models based on keras framework. Journal of Soft Computing and Data Mining. 2021; 2(2):49-62.

[48] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In proceedings of the conference on computer vision and pattern recognition 2016 (pp. 770-8). IEEE.

[49] Lee JY, Dernoncourt F. Sequential short-text classification with recurrent and convolutional neural networks. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2016 (pp. 515-20). Association for Computational Linguistics.

[50] Too EC, Yujian L, Njuki S, Yingchun L. A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture. 2019; 161:272-9.

[51] Ma J, Du K, Zheng F, Zhang L, Gong Z, Sun Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Computers and Electronics in Agriculture. 2018; 154:18-24.

[52] Li Y, Nie J, Chao X. Do we really need deep CNN for plant diseases identification? Computers and Electronics in Agriculture. 2020; 178:105803.

[53] Barbedo JG. Factors influencing the use of deep learning for plant disease recognition. Biosystems Engineering. 2018; 172:84-91.

[54] Gandhi R, Nimbalkar S, Yelamanchili N, Ponkshe S. Plant disease detection using CNNs and GANs as an augmentative approach. In international conference on innovative research and development 2018 (pp. 1-5). IEEE.

Md. Ashikur Rahman Khan is a Professor at the Department of ICE, NSTU. He has been working in NSTU since 2006. Formerly, he has worked as an Engineer in Inspection and Testing division of Rural Electrification Board, Bangladesh for the period 2001-2006. He has achieved Ph.D degree in 2012 from University Malaysia Pahang, Malaysia. Dr. Khan is working as reviewer and editorial board member of different journals. His research interests include Advance Machining, Artificial Intelligent, Neural Network, Machine Learning, and Renewable Energy.
Email: ashik@nstu.edu.bd

Jesmin Akther is a Lecturer at the department of CSE, EUB since December 2020. To add to it, she was a lecturer at Uttara Institute of Business & Technology (UIBT), Dhaka, Bangladesh. She has completed her graduation and post-graduation degree from the Department of ICE, NSTU, Bangladesh. She achieved the 3rd position in the University Merit List at the Postgraduate level. She has successfully done hands on activities such as problem solving, programming contests both onsite and offsite (2013 to 2017), android development (four apps are on google play store), taken seminars and arranged workshops for Intra department of NSTU. Her research area includes: Deep and Machine Learning, and Android Component-Based Applications.
Email: jesminnipu1@gmail.com

Fardowsi Rahman is a Lecturer at the Department of CSE, PaU, Dhaka, Bangladesh. She has graduated and post-graduated from the Department of ICE, NSTU, Bangladesh. She is the University Rank Holder (Merit Position: 1st) at both undergraduate and post-graduate levels. Her research fields are Data Mining, Artificial Intelligence, Machine Learning, Neural Networks, Cloud Computing, Cyber Security, and Big Data.
Email: fardowsirahman09@gmail.com

Appendix I

S.No.	Abbreviation	Description
1	3D	Three-Dimensional
2	CNN	Convolutional Neural Network
3	DL	Deep Learning
4	DRN	Deep Residual Network
5	ICE	Information and Communication Engineering
6	INC-VGGN	Inception-Visual Geometry Group Network
7	JPG	Joint Photographic Group
8	MFRCNN	Multi-scale Local and Global Feature Representation with Convolutional Neural Network
9	ReLU	Rectified Linear Unit
10	RGB	Red Green Blue
11	SCNN	Simplified Convolutional Neural Network
12	SVM	Support Vector Machine
13	VGG Net	Visual Geometry Group Networks
14	YOLO v4	You Only Look Once Version 4