Performance Evaluation of VGG models in Detection of Wheat Rust

Rajwinder Singh[1], Rahul Rana[2] and Sunil Kr. Singh[3]
[1,2&3]Department of Computer Science and Engineering,
Chandigarh College of Engineering and Technology (Degree Wing), Sector-26, Chandigarh
Affiliated to Panjab University, Chandigarh, India
E-Mail: rajwindersingh12310@gmail.com, rahulrana9944@gmail.com, sksingh@ccet.ac.in

Abstract - The agricultural sector is the backbone of Indian economy and social development but due to lack of awareness towards crop management, a large number of crops get wasted each year. Automated Systems are required for this purpose. This paper tries to highlight the efficiency of two existing models of deep learning, VGG16 and VGG19 for proper detection of wheat rust disease in the infected wheat crop. These two models use convolutional neural networks for image classification and which can be used to design an intelligent system which can easily detect wheat rust in crop images. This paper basically presents the comparative analysis of the accuracy and efficiency along with usability to select the best model for systems that can be used for crop safety.
Keywords: Deep Learning, Wheat Rust, Pre-Trained Models, Performance Evaluation, Crop Disease Detection

I. INTRODUCTION

In India agriculture contributes significant figure to the gross domestic product (GDP) and also it is the source of the largest provider of livelihood, mostly in the vast rural areas. About 70% of the Indian population depends upon agricultural sector. Although, there is a wide variety of crops that are cultivated in India wheat holds the majority of share amongst them. The wheat production was around 100 million tonnes for 32 million hectare area of land.

Mostly the production of the wheat is dependent on the capricious weather, however, recently Stripe rust caused by Puccinia striiformis f. sp. tritici (Pst) and leaf rust caused by P. recondita f. sp. tritici(Prt) are two of the many wheat diseases which holds importance in airborne diseases of wheat worldwide.[1,2]. These two are important factors which are responsible for affecting food security in India, and also for jeopardizing the potential GDP growth every year, this mostly happens in areas, where there are large occurrence and high epidemic frequency of these two diseases. Thus we gravely need the timely detection and monitoring of these diseases in order to impede the spread of these diseases. Present scenario of wheat rust detection and monitoring is purely dependent on the on-field physical inspection by the professionals, which is indeed a very costly and time-consuming process and mostly the availability of such professionals is only confined to suburban parts, because of which such wheat diseases in rural areas go unchecked most of the time, which subsequently causes tremendous loss of cost, time and efforts. Hence it is the call of the hour to develop such systems which can automatically detect and tell about the wheat rust just by feeding the image of a crop to be tested.

The main indicators of field diagnosis are the symptoms produced by many diseases; in recent years many safe practices have been made for proper detection of these symptoms but automatic disease diagnosis and control is the topmost concern of scientists worldwide. A dilemma has been created by plant diseases as a significant amount of reduction in both of quantity and quality can be caused by these diseases [3].

With rapid advancement in the development of image classification techniques, we can easily classify the image by using CNNs. CNNs consist of multiple hidden layers, along with the general input and the output layer. The hidden layers of a CNN consist of pooling layers, fully connected layers, convolution layers, and normalization layers. In the proposed model, data augmentation is done for pre-processing, to improve the accuracy of the training the model on these images. In this paper, we have tried to incorporate the state of the art pre-trained CNN model available to accomplish the task of wheat images as healthy or diseased.

Section I contains the introduction, Section II is dedicated for the related works and Section III is about the literature survey, Section IV will explain the experimentation part, we have dedicated the Section V for result and analysis and Section VI contains the conclusion and future scope.

II. RELATED WORK

Application of Convolutional Neural Networks (CNN) for the purpose of image classification is discussed by Lele in his paper [4]. Supervised and Unsupervised Image classification is discussed by him along with the working of the layers of CNN. Krizhevsky *et al.,* [5] trained and used a large deep neural network to classify the 1.3 million high-resolution images from the Imagenet training set into 1000 different classes. Kim [6] has employed the use of convolutional neural networks for the purpose of sentence classification. Ji *et al.,* [7] has developed a 3D convolutional neural network which is capable of recognising spatial as well as temporal dimensions by using 3D convolutions to detect human activities in surveillance systems. They have obtained an average accuracy of 90.2% by employing the

use of 3D CNN. Ciresan *et al.,* [8] implemented variants of convolutional neural networks and published results on benchmark object classification (NORB, CIFAR10) and handwritten digit recognition (MNIST) with error rates of 2.53%, 19.51%, 0.35%, respectively. Two new approximations to standard CNN were proposed by Rastegari *et al.,* [9], namely binary-weight-networks and XNOR-networks. These approximations are basically aimed towards improving the time and space complexities where 32X memory can be saved using the binary values for filters and 58X faster convolutional operations.

Till date, various conventional methods are used for detection of infections in crops due to various viruses and pathogens. Chaudhary *et al.,* [10] has reviewed various techniques namely DNA-RNA based methods, enzyme and antibody-based detection method, etc., which require well-built infrastructure and controlled environment to detect infections caused by various pathogens mainly polymerase chain reaction (PCR), etc. Later on, they have to review a loop-mediated isothermal amplification (LAMP) technique for the same purpose. Abdalla *et al.,* [11] used electron microscopy examination to observe severe viral-like symptoms in potatoes. Dutta, *et al.,* [12] in their case study have shown an approach where forewarning of various areas affected by yellow rust can be done using satellite data, various factors like temperature and humidity conditions at a place were taken into account to infer the results from the satellite images.

Sladojevic [13] used deep convolutional neural networks to classify leaf images and develop a model for recognition of plant diseases, they were able to achieve an average accuracy of 96.3% for leaves dataset. Revathi and Hemalatha [14] have used particle swarm optimisation (PSO) [15] for feature selection which is then used for injured leaf spot in the cotton plant. Badugu *et al.,* [16] worked with polluted leaves and calculated texture features of it. They have presented an analytical study by presenting the histograms and discussed various parameters such as mean, skewness, etc. They have claimed that automatic detection of polluted leaves can be done using these statistical measures and histograms.

III. LITERATURE SURVEY

A. Convolutional Neural Network

The convolutional neural networks or the CNN consist of a combination of various layers, which include the convolutional layer, the max-pooling layer and the fully connected layer. A short description of the layers of CNN is given below:

1. Convolutional Layer: Convolution is the process of filtering through the image for a specific pattern. Tweaking the weights and biases impact the effectiveness of the filtering process. Neurons perform the 'convolution operation' for the process.

Same weight and bias parameters are shared by the neurons in a given filter. Therefore, anywhere on the filter, a given neuron possesses the same weights and biases and is connected to the same number of input neurons. Hence it permits the filter to identify the same pattern in different sections of the image. Entire image scan is ensured by arranging these neurons in a grid-like structure.

2. Max-Pooling Layer: The simple patterns discovered by the convolutional layer are built up by the Rectified Linear Unit (ReLu) and pooling layer. The convolutional layer has each node connected to a node that fires like in other networks. ReLu is the activation unit. Vanishing gradient is a potential issue because convolutional neural networks are trained using back-propagation algorithm. ReLu helps to keep the gradient constant at each layer of the network. Therefore, ConvNets are saved from the risk of not being properly trained because of a harmful slowdown in pivotal initial layers. The dimensionality reduction is done by the pooling layer. During Max Pooling, the largest element from the rectified feature map within a spatial neighborhood is considered. Average Pooling or Sum Pooling can also be used. In practice, Max Pooling surpasses the other two and hence, it is incorporated in the proposed model.

3. Fully-Connected Layer: In order to equip the network with the ability to classify data samples efficiently, a fully connected layer is attached to the end of the network.

In this paper, we compare two pre-trained models present in keras [17] namely VGG16 and VGG19. VGG stands for Visual Geometric Group.

B. VGG Network Architecture

In 2014, Simonyan and Zisserman introduced the VGG network architecture [18]. The simplicity is what characterizes this network; it uses only 3×3 convolutional layers stacked on top of each other in increasing depth. max pooling handles the reducing volume size. Softmax classifier follows the two fully-connected layers, each having 4,096 nodes. Training VGG16 and VGG19 seemed challenging for Simonyan and Zisserman (specifically regarding convergence on the deeper networks), so they first trained smaller versions of VGG with fewer weight layers first in order to make training easier. The converged smaller networks were then used as initializations for the larger and deeper networks — this process is known as pre-training. Pre-training is a very time consuming, tedious task, which requires an entire network to be trained before it can serve as an initialization for a deeper network. Weights are downloaded automatically when instantiating a model. Of these models, we have used VGG16 and VGG19 pre-trained model. The subset of the ImageNet database was used to train this model, which is used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [19]. More than a million images can be trained and can be classified into 1000 object categories by VGG-19. As a result, the model has learned rich feature representations for a plethora of

images. 19 layers with learnable weights constitute the VGG-19: 16 convolutional layers and 3 fully connected layers. In this network, filters of size 3-by-3 are used in all convolutional layers.

IV. EXPERIMENTATION

The experimentation phase consists of various steps from data acquisition to data pre-processing to data augmentation and then training followed by validation and testing step and in the end final result generation and analysis. The complete flow of control is depicted in Fig. 1.
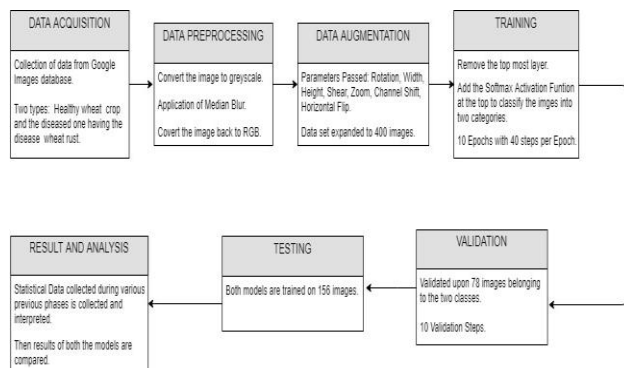

Fig. 1 Workflow

The above steps are explained below:

A. Data Acquisition: The first step was the data acquisition part. The dataset for the purpose was created by us, taking Google images [20] of two types the healthy wheat crop and the diseased one having the disease of wheat rust and also from Imagenet [21]. We have taken 200 images for the training purpose, 26 images for validation, 39 images for the purpose of testing. Fig. 2 shows the images imported along with the class labels. Where the class label [1,0] signifies that the particular image is healthy and the class label [0,1] means the image is that of the diseased crop.

Fig. 2 Imported images with class labels

Fig. 3 shows a healthy wheat crop and Fig. 4 shows wheat having the disease of wheat rust

Fig. 3 Image of Healthy Wheat Crop [21]

Fig. 4 Image of Infected Wheat Crop with Wheat crust [21]

B. Data Pre-Processing: In data pre-processing step we took the images collected and to remove the noise and to smoothen it we have applied the median blur algorithm and before feeding the images to median blur, it was first transformed to the grey scale. Fig. 5 depicts the original image in Fig. 6 shows the image transformed to grey scale and Fig. 7 shows the smoothened image.

Fig. 5 Original Coloured Image [20]

Fig. 6 Converted Grey Scale Image

Fig. 7 Smoothened Image

C. Data Augmentation: The smoothened images are converted back to the colour images. As the data set was very small therefore data augmentation techniques were applied to the data. Fig. 8 shows one of the images having wheat rust and Fig. 9 shows the augmented image set.
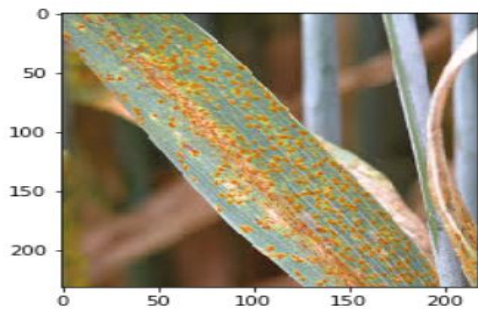
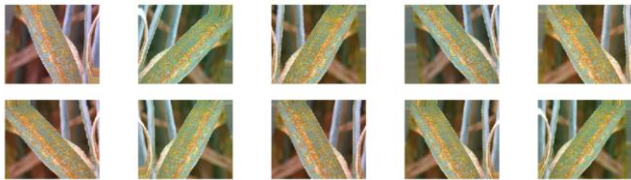Fig. 8 Leaf with wheat rust before augmentation

Fig. 9 Augmented Images

The parameters passed during the data augmentation are rotation_range=10, width_shift_range=0.1, height_shift_range=0.1, shear_range=0.15, zoom_range=0.1, channel_shift_range=10., horizontal_flip=True

And as a result the new data set generated on which we have applied the models. This dataset consists of 400 training images, 78 images for validation and 156 images for the purpose of testing.

1. Model Training: The next phase is the training of the models. Models are trained one by one on 400 images of healthy and diseased wheat. When we have imported the VGG16 and VGG19 models the pre-trained weights are automatically downloaded and these models are meant to classify the images in 1000 different categories, but here we only want to classify it into 2 categories, therefore we have popped the topmost layer and added a dense layer with softmax activation function with only two neurons to classify it into two categories of healthy and diseased. The number of epochs that have been running is 10 and steps per epoch are 40.

2. Model Validation: To avoid the problem of over fitting of the data, the trained model has been validated upon 78 images belonging to the two classes. The number of validation steps is 10.

3. Testing: Finally after the validation both the models are tested on a set of 156 images and the results are compared in the results and analysis part.

V. RESULT AND ANALYSIS

Both the models VGG16 and VGG19 are present in the keras library. When the models are imported, the pre-trained weights are automatically downloaded. The top layer of these models has been replaced by a dense layer having a softmax activation function which categorises the images into two classes. Finally, the models were compiled using

the Adam optimizer having the learning rate of 0.0001. The VGG16 model was trained with an average accuracy of 93.825% with an average loss of 62.207%. The average validation accuracy of the same model came out to be 82.993% with an average loss of 63.953%. Fig. 10 shows the result of VGG16 over 10 epochs.

```
Epoch 1/10
 - 504s - loss: 0.6754 - acc: 0.7825 - val_loss: 0.6608 - val_acc: 0.8367
Epoch 2/10
 - 550s - loss: 0.6523 - acc: 0.8825 - val_loss: 0.6504 - val_acc: 0.8367
Epoch 3/10
 - 465s - loss: 0.6404 - acc: 0.9075 - val_loss: 0.6468 - val_acc: 0.8469
Epoch 4/10
 - 460s - loss: 0.6289 - acc: 0.9500 - val_loss: 0.6436 - val_acc: 0.8542
Epoch 5/10
 - 490s - loss: 0.6197 - acc: 0.9600 - val_loss: 0.6352 - val_acc: 0.8571
Epoch 6/10
 - 512s - loss: 0.6122 - acc: 0.9750 - val_loss: 0.6356 - val_acc: 0.8163
Epoch 7/10
 - 493s - loss: 0.6059 - acc: 0.9775 - val_loss: 0.6355 - val_acc: 0.8061
Epoch 8/10
 - 535s - loss: 0.6000 - acc: 0.9775 - val_loss: 0.6303 - val_acc: 0.8229
Epoch 9/10
 - 516s - loss: 0.5953 - acc: 0.9825 - val_loss: 0.6244 - val_acc: 0.8367
Epoch 10/10
 - 517s - loss: 0.5906 - acc: 0.9875 - val_loss: 0.6327 - val_acc: 0.7857
<keras.callbacks.History at 0x1d479194630>
```

Fig. 10 Epoch Summary for VGG16

Fig. 11 shows the accuracy graph for training, Fig. 12 shows the loss graph for training, Fig. 13 shows the validation accuracy and Fig. 14 shows the validation loss. All of these graphs are for the VGG16 model and are plotted against number of epochs.
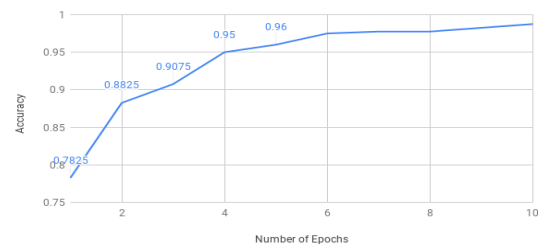

Fig. 11 Graph of Accuracy vs. No of epochs for VGG16

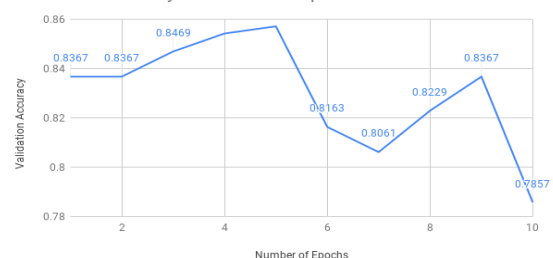Fig. 12 Graph of loss vs. No of epochs for VGG16

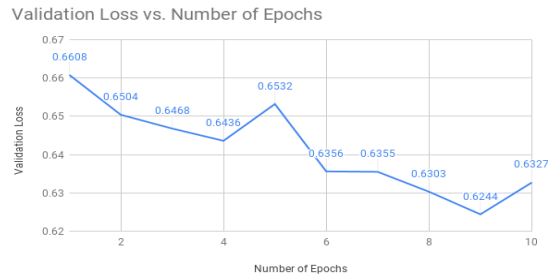Fig. 13 Graph of validation accuracy vs. No of epochs for VGG16

Fig. 14 Graph of validation loss vs. No of epochs for VGG16

The VGG19 model was trained with an average accuracy of 93.675% with an average loss of 62.301%. The average validation accuracy of the same model came out to be 75.504% with an average loss of 65.388%. Fig. 15 shows the result of VGG19 over 10 epochs.

```
Epoch 1/10
  - 577s - loss: 0.6743 - acc: 0.7775 - val_loss: 0.6724 - val_acc: 0.7041
Epoch 2/10
  - 591s - loss: 0.6528 - acc: 0.8950 - val_loss: 0.6679 - val_acc: 0.7041
Epoch 3/10
  - 588s - loss: 0.6394 - acc: 0.9175 - val_loss: 0.6600 - val_acc: 0.7449
Epoch 4/10
  - 563s - loss: 0.6280 - acc: 0.9325 - val_loss: 0.6597 - val_acc: 0.7396
Epoch 5/10
  - 566s - loss: 0.6199 - acc: 0.9500 - val_loss: 0.6465 - val_acc: 0.8163
Epoch 6/10
  - 562s - loss: 0.6136 - acc: 0.9675 - val_loss: 0.6491 - val_acc: 0.7653
Epoch 7/10
  - 584s - loss: 0.6083 - acc: 0.9700 - val_loss: 0.6475 - val_acc: 0.7857
Epoch 8/10
  - 589s - loss: 0.6024 - acc: 0.9850 - val_loss: 0.6488 - val_acc: 0.7292
Epoch 9/10
  - 577s - loss: 0.5978 - acc: 0.9850 - val_loss: 0.6409 - val_acc: 0.7755
Epoch 10/10
  - 567s - loss: 0.5936 - acc: 0.9875 - val_loss: 0.6460 - val_acc: 0.7857
```

Fig. 15 Epoch Summary for VGG19

Fig. 16 shows the accuracy graph for training, Fig. 17 shows the loss graph for training, Fig. 18 shows the validation accuracy and Fig. 19 shows the validation loss. All of these graphs are for the VGG19 model and are plotted against number of epochs.
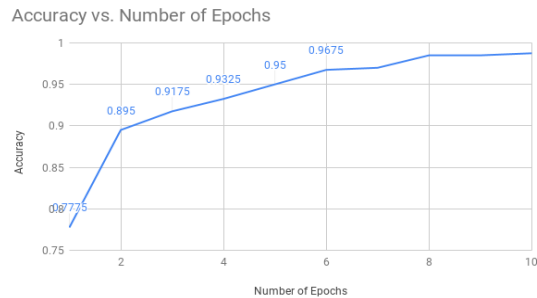
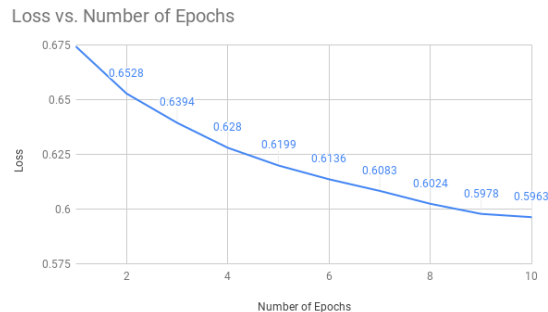Fig. 16 Graph of Accuracy vs. No of epochs for VGG19
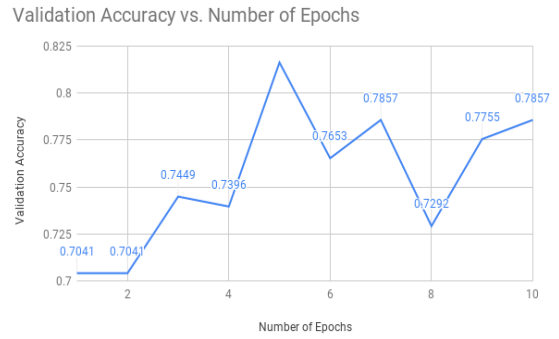
Fig. 17 Graph of loss vs. No of epochs for VGG19

Fig. 18 Graph of validation accuracy vs. No of epochs for VGG19
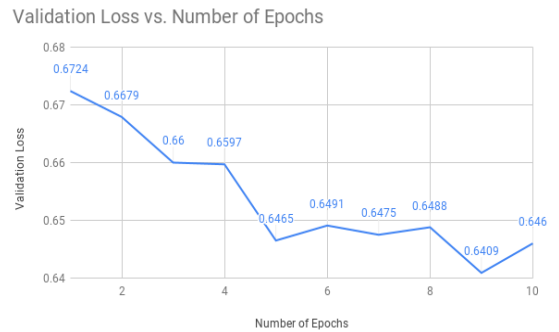
Fig. 19 Graph of validation loss vs. No of epochs for VGG19

Fig. 20 shows the confusion matrix plot for VGG16 with false positive 4 and false negative 17 and Fig. 21 shows the confusion matrix plot for VGG19 with false positive 2 and false negative 23.

Confusion matrix, without normalization
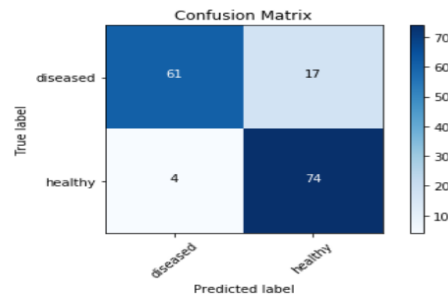[[61 17]
 [4 74]]

Fig. 20 Confusion Matrix for VGG16

Confusion matrix, without normalization
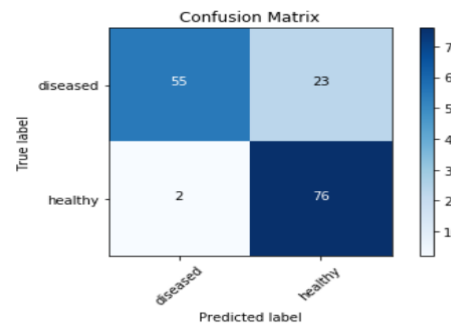[[55 23]
 [2 76]]

Fig. 21 Confusion Matrix for VGG19

From these two figures, it can be seen that the VGG16 model predicted correctly 61 times diseased crops when the image was actually that of diseased and 74 times predicted healthy for healthy images. In total it predicted 21 times wrongly the diseased as healthy or vice versa both cases included, thus leading to an 86.538% testing accuracy. Whereas the VGG19 model on the same data set gave the testing accuracy of 83.974%, wherein correctly predicting diseased as diseased 55 times and healthy as healthy 76 times. This model failed 25 times to correctly predict the class of test image. The table is given which contains the comparison between VGG16 and VGG19.

TABLE I COMPARISON BETWEEN VGG16 AND VGG19

	VGG16	VGG19
Weight Layers	16	19
Total Parameters	13,42,62,546	13,95,72,248
Trainable Parameters	2,002	2,008
Non-Trainable Parameters	13,42,60,544	13,95,70,240
Average Training Accuracy	93.825%	93.675%
Average Training Loss	62.207%	62.301%
Average Validation Accuracy	82.993%	75.504%
Average Validation Loss	63.953%	65.388%
Testing Accuracy	86.538%	83.974%

VI. CONCLUSION AND FUTURE SCOPE

Through this paper, a comparison can be drawn between the models of existing deep neural networks in terms of their efficiency to detect wheat rust in crop images. It has been found that the VGG16 performs much better than the VGG19 for this task with both average training and average testing accuracies more for VGG16 than VGG19. These findings can be used further in future for the purpose of making choice between using the two CNN models present in the Keras library. Also, these findings can be used extensively for the purpose of designing automated systems which can directly detect the wheat rust in infected crops, this purpose can be achieved by making an application which when fed with the crop image can classify it into a healthy or infected crop. This is an efficient method to prevent a huge amount of crop wastage due to these kinds of crop diseases. Also, the accuracies of the existing models can be further improved by using the techniques like dropout in hidden layers, random weight initialisation and parameter tuning.

REFERENCES

[1] Z. Q. Li, and S. M. Zeng, "Wheat rust in China", *Beijing: China Agriculture Press*, 2002.

[2] J. Huerta-Espino, R. P. Singh, S. Germain, B. D. Mccallum, R. F. Park, W. Q. Chen, S. C. Bhardwaj, and H. Goyeau, "Global status of wheat leaf rust caused by Puccinia triticina", *Euphytica*, Vol. 179, No. 1, pp. 143-160, 2011.

[3] S. Weizheng, W. Yachun, C. Zhanliang, and W. Hongda, "Grading method of leaf spot disease based on image processing", 2008 *International Conference on Computer Science and Software engineering*, Vol. 6, pp.491-494, 2008.

[4] N. S. Lele, "Image Classification Using Convolutional Neural Network", *International Journal of Scientific Research in Computer Science and Engineering*, Vol. 6, No. 3, 2018.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[6] Y. Kim, "Convolutional neural networks for sentence classification", *arXiv preprint arXiv:1408.5882*, No. 2, 2014.

[7] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 35, No. 1, pp.221-231, 2013.

[8] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22, No. 1, pp. 1237, 2011.

[9] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks", *European Conference on Computer Vision*, pp.525-542, 2016.

[10] S. Chaudhary, and P. G. Bagali, "Applications of LAMP Based Diagnostic Kits in Crop Management", *Current Trends in Biomedical Engineering & Biosciences*, Vol. 7, No. 2, 2017.

[11] O. A. Abdalla, A. I. Eraky, S. A. Mohamed, and F. G. Fahmy, "Molecular identification of viruses responsible for severe symptoms on potato (Solanum sp.) growing in Assiut Governorate (Upper Egypt)", *International Journal of Virology Studies and Research*, Vol. 4, No. 3, pp.29-33, 2016.

[12] S. Dutta, S. K. Singh, and M. Khullar, "A case study on forewarning of yellow rust affected areas on wheat crop using satellite data", *Journal of the Indian Society of Remote Sensing*, Vol. 42, No. 2, pp.335-342, 2014.

[13] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification", *Computational intelligence and neuroscience*, 2016.

[14] P. Revathi, and M. Hemalatha, "Identification of cotton diseases based on cross information gain_deep forward neural network classifier with PSO feature selection", *International Journal of Engineering and Technology*, Vol. 5, No. 6, pp.4637-4642, 2014.

[15] C. Zhou, H. B. Gao, L. Gao, and W. G. Zhang, "Particle swarm optimization (PSO) algorithm", *Application Research of Computers*, Vol. 12, pp.7–11, 2003.

[16] S. K. Badugu, R. K. Kontham, V. K. Vakulabharanam, and B. Prajna, "Calculation of Texture Features for Polluted Leaves", *International Journal of Scientific Research in Computer Science and Engineering*, Vol. 6, No. 1, pp.11-21, 2018.

[17] Keras.io. (2018). *Keras Documentation*. [online] Available at: https://keras.io/ [Accessed 7 Feb. 2018].

[18] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, No. 4, 2014.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large-scale visual recognition challenge", *International Journal of Computer Vision*, Vol. 115, No. 3, pp.211-252, 2015.

[20] Google.co.in. (2018). *wheatrust - Google Search*. [online] Available at: https://www.google.co.in/search?q=wheatrust&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUotrx4Z7dAhXCdysKHYUxDRgQ_AUIDCgD&biw=1536&bih=762&dpr=1.25 [Accessed 21 Mar. 2018].

[21] Image-net.org. (2018). *ImageNet Tree View*. [online] Available at: http://www.image-net.org/synset?wnid=n13065089# [Accessed 20 Mar. 2018].