Ambient Science, 2022: Vol. 09(3); 34-39
DOI:10.21276/ambi.2022.09.3.ta02

ambient
SCIENCE
Vol. 9(3):34-39
Year 2022

Published by: National Cave Research and Protection Organization, India

TECHNOSCIENCE ARTICLE

Plant Disease Detection using Transfer Learning in Precision Agriculture

S.V.Jansi Rani

Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamilnadu, India

Study Area: Chennai, India
Coordinates: 13°04′57″N 80°16′30″E

Keywords: Classification, Computer Vision, Deep Learning

Abstract

Many plants exhibit apparent disease symptoms and thus the method used so far is that a plant pathologist detects the disease through microscopic and optimal observation. The problem with the diagnosis process is that it is slow to perform, and the level of accuracy depends on the pathologist's capability, here this provides an area for computer-aided diagnostic systems in precision agriculture. Thus, deep learning offers a robust tool with highly accurate results. But however, there are several drawbacks like overfitting, and low performance when tested on datasets from new environments. All the above disadvantages can be overcome by capturing images from different angles, in different weather conditions, and at different backgrounds which replicates practical situations. The augmentation techniques like rotation and zooming are used to generate more data. Finally, the model is trained and developed by transfer learning, using pre-trained models like InceptionV3, ResNet and Inception Resnetv2. From the experimentation, it is found that Resnet performed well than other models.

Introduction:

IChange from the normal state of the plant affects its important functions, and it is caused by microbes and insects called Plant Diseases. The use of various pesticides, fungicides, and chemicals has damaged plant health and also the ecosystem. This ultimately makes a decline in the production of crops in precision agriculture. Not only these but also global warming, ozone layer depletion which causes the penetration of UV rays into the earth's atmosphere and climatic change have drastically affected plant cultivation. The diseases in plants have negative effects on agriculture, and if it was not found on time, there will be a high risk of food insecurity (Fina *et al*., 2013).

According to UNESCO, plant diseases have increased along with globalization in trade, and climate change. The plant disease epidemics are increasing and causing huge losses in food crops, threatening farmers across the globe and the food security of people (Strange & Scott 2005). Approximately 50% of plant cultivated is affected by diseases (Harvey *et al*., 2014).

Visually observable patterns are difficult to decipher the diseases, leading many farmers to make inaccurate assumptions. Thus, mechanisms for the treatment of these diseases taken may be ineffective and sometimes harmful. There were situations where the wrong detection of diseases due to inadequate knowledge of the professionals caused crop damage. Farmers try to implement common disease prevention techniques, as they are not able to reach experts for advice (Chen *et al*., 2012). Identifying diseases made by farmers might go wrong and gathering the Domain expert's ideas are costly. This is the underlying motivation of this work, which enables farmers to identify diseases on their own without other's assistance, which brings into the picture the automated computer-aided image-based disease detection and classification technique that will serve as a replacement for the above difficulties. Deep learning is now an active research area in artificial intelligence and has been successful in different fields of science. Thus, advancements in this field provide an opportunity to extend the research to the field of agriculture(Mohanty et al., 2016) Some Deep learning architectures are: GoogleNet Inception V3 (Szegedy *et al*., 2015), VGG net (Simonyan & Zisserman, 2015), DenseNets (Huang *et al*., 2016), Alexnet (Krizhevsky et al., 2012), Microsoft ResNet (He *et al*., 2016), Inception V4 (Szegedy *et al*., 2016). Some issue with these networks is the vanishing gradient and degradation problem which causes a reduction in accuracy. However, many techniques have

**Corresponding Author: svjansi@ssn.edu.in*

Ambient Science, 2022: Vol. 09(3); 34-39
DOI:10.21276/ambi.2022.09.3.ta02

been introduced to resolve the issues. Here, the proposed method aims to compare different pre-trained models, such as Inceptionv3, Inception ResNetv2, and ResNet50 and finds the best pre-trained model. Then some parameters are tuned to get better accuracy.

Existing Systems

Chen *et al.* (2020) developed a model to identify plant disease using deep learning. Enhanced VGGNet with Inception module was combined to generate a new model named INC- VGGN network, and the high-dimensional feature extraction and classification are done using the external final. They performed a comparison between their models with existing methods.

Argueso *et al.* (2020) used Siamese networks along with triplet loss that was trained about plant leaves and their disease types on small datasets. Their model gave better results than many classical techniques. Ferentinos *et al.* (2018) used Deep learning CNN models for plant disease detection. The dataset was open comprising 87,848 images of plant diseases and 58 classes of plants. Arsenovic *et al.* (2019) used augmentation techniques such as traditional augmentation methods and generative adversarial networks for balancing the data. Extensive experimentation is done to train the model in a controlled environment and also in real-life situations. A two-stage CNN architecture was developed which produced 93.67% of accuracy.

Sladojevic *et al.* (2016) came up with a new idea for developing a plant disease detection model, by using deep CNN. The Berkley Vision and Learning Centre developed a deep learning model called Caffe which recognized 13 plant species. The experimental results obtained precision between 91% and 98% and an average of 96.3%. Tm et al. (2018) performed a slight variation of the CNN model called LeNet on tomato diseases. The model was chosen as the simplest model that uses minimum computing resources to obtain higher accuracy results compared to other models. This model has obtained an average accuracy of 94-95% even under unfavourable conditions.

Tiwari *et al.* (2020) worked by developing a model that chose pre-trained models like VGG19 for fine-tuning, and different classifiers for classification results were analysed. Bhagat *et al.* (2020) in their paper performed an analysis of plant-based disease detection systems developed till today. In this paper, a basic Convolution network was trained on the bell pepper plant dataset, different a variety of simulations using neurons and different layers were used in that basic CNN. Literature seems that each work has its own limitations such as some dealing with specific species and some training on a very small dataset. The proposed model aims at finding the best pre-trained model using the standard plant village dataset. The best pre-trained model found is further fine-tuned with varying hyperparameters to get the best result.

Proposed methodology

The plant disease dataset consisting of 14 plants and 38 classes was divided and a new dataset with 5 plant species namely, Tomato, Grape, Potato, Pepper Bell, and Maize with 23 classes. The new dataset consisting of 5 plants is divided into train and test data. The input to the model is the dataset consisting of 5 plant species and the output is the diseases identified as class labels. In the pre-processing phase, GAN and Traditional augmentation are performed to balance the dataset. This processed data is then fed into different types of pre-trained CNN models namely InceptionV3, InceptionResNetV2 and ResNet50 as shown in Fig.-1.

As per the results using the performance metrics like accuracy, precision, recall and f1 score, the efficient architecture is chosen as the classification model. Finally, the testing data is fed into the chosen classification model. Thus, the output of the model, i.e., the class labels is obtained.

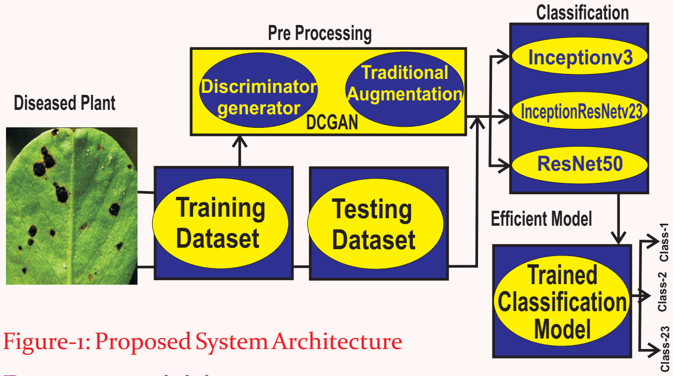

Figure-1: Proposed System Architecture

Dataset acquisition

The New Plant Disease Dataset consists of 14 plant species with 38 classes was divided and a new dataset with 5 plants was generated. The new dataset containing 42,486 images of 5 different plant leaves namely, Tomato, Potato, Grape, Pepper bell, and Maize was chosen as the training and testing datasets. The dataset includes 23 different classes, and each class is labelled as a healthy or disease label.

Generative Adversarial Networks: it contains two CNN architectures namely discriminator and Generator, where both are trained with the dataset in parallel. The generated images classified as real images are stored in the respective classes of the image (Radford *et al.*, 2015).

Generator: this concept is from the noise, the features convolve till it generates a proper image. The layers here are:

- Dense Layer: it is the layer which uses noise.
- Conv2DTranspose: this layer convolves backwards, that is, to upscale and downscale the image at the same time. It performs similar to upSampling2D and then Conv2D.
- LeakyReLU: it performs better than the ReLU functions, as it prevents the vanishing gradient problems.
- Reshape: this layer converts a one-dimensional vector to a three-dimensional array which is named transformation. The generator consists of one input layer, three hidden layers

and one output layer. The output layer uses Tanh activation function (Hitawala *et al*., 2018).

Discriminator: this network is a normal convolutional neural network which contains one input layer, three hidden layers, and one output layer. The input to the model is the image and the output is the binary value. In convolutional networks, dropout layers are only used on the last layer. It uses the Sigmoid activation function in the last layer, as only a binary output will be obtained. The Adam optimizer is used, as it performs better than all other optimizers. Finally, the learning rate is set as 0.0002 and the beta as 0.5. It is trained first with a sample dataset and left to classify the generated images.

Pre-Trained Models

Inceptionv3: it is a CNN model mainly used for image analysis and detection. It was developed by Googlenet. It is the third version of Google's Inception CNN model, introduced first at the ImageNet Recognition Challenge (Szegedy *et al*., 2016) The Input size is299x299x3 and the number of layers is 48. It consists of 3 layers namely: Convolution Layer, Subsampling layer, and Fully Connected layer.

In the convolution layer, each kernel is passed over the input dataset, and a feature map is extracted as a result in each convolution layer by adding bias. In the sub-sampling layer, down-sampling of the feature map is done using average pooling or max pooling. In the fully connected layer, there are 1 or more hidden layers and the final feature maps are passed to the next layer through the Activation function. The weights are updated and are also back-propagated for updating. It stacks up to 11 inception modules, which consist of pooling layers, convolution filters and the Relu Activation Function. Three fully connected layers of size (1024,512,3) are attached to the final concatenation layer. The activation function used here is Softmax (Szegedy et al., 2015).

ResNet50: a residual neural network (ResNet) is a kind that's built on constructs that are similar to pyramidal cells in the cerebral cortex, performed by utilizing skip connections to skip over layers (He et al., 2016). The input size is 224 x 224 x 3 and the number of layers is 50. There are certain issues with CNN models namely, vanishing gradient problems and degradation problems. The resnet architecture brings solutions to the above problems and increases accuracy.

It consists of convolution layers, sub-sampling layers and fully connected layers. The activation function used here is Softmax. The convolution layers have kernels of different sizes. The final sub-sampling layer uses Global Average pooling and the final fully connected layer uses Softmax for multiclass classification. The vanishing gradient problem is solved by carrying out the result of previous results to the next layer which is named a a shortcut or skip connection. It consists of two types of blocks namely, Identity block and Convolution block. The identity block is the unique block of the ResNets. As the name suggests, it has the same dimensions in input activation and output activation. The Convolution block is where the dimensions of input and output activation do not match (Ji *et al.*, 2019) The difference in dimensions are matched by adding a CONV2D layer in the shortcut path.

InceptionResNetv2: it is a Convolutional Neural Network model that combines the architecture of Inceptionv3 and Resnet with some variations added to it.

The input size:299x299x3, Layers:168, Activation Function: Softmax Activation function. It stacks up inception modules and resnet modules together as an Inception-Resnet blocks. There are 3 inception-resnet blocks and dropout and max pooling layers (Szegedy *et al*., 2017). In this model, each inception block is connected to a filter expansion layer (1x1 convolution without activation function) to scale up the dimension of filters before the addition that matches the depth of the input (Nguyen *et al*., 2018). In the case of Inception-ResNet, batch-normalization is used only on top of the inception-resnet blocks. x

Experimental Results

The implementation was built using a python programming language. The experiment is carried out in the NVlink GPU Server, Nvidia V100 32GB. The implementation was set up in three stages namely, the data acquisition stage, pre-processing stage and classification stage. In the pre-processing stage, the number of images in each class was increased by generating new images using the Deep Convolutional Generative Adversarial Network, which consists of a discriminator and a generator. The discriminator was used to check the originality of the fake images, while the generator is used to produce fake images.

In the last phase, classification was done using pre-trained models such as Inceptionv3, InceptionResNetv2, and ResNet50. We used all these models to train upto 25 epochs of batch size 128 with a learning rate of 0.01. The models were trained for various optimizers and categorical cross-entropy loss functions for multi-class classification.

Performance Metrics

Accuracy: it is the ratio of correctly predicted observations to the total observations.

1. Precision: it is the ratio of true positives to all positives.
   $$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$
2. Recall: Recall is the ratio of actual positives to all positive observations.
   $$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

F1 score = it is the mean of Precision and Recall. Thus, it takes all observations including false positive and false

Ambient Science, 2022: Vol. 09(3); 34-39
DOI:10.21276/ambi.2022.09.3.ta02

negative into consideration.

$$F1\ score = \frac{Precision + Recall}{2}$$

Here,
True Positives = these are the correctly predicted positive values where the actual and predicted labels are "True".
True Negatives = these are the correctly predicted negative values where the actual and predicted labels are "False".
False Positives = these are wrongly predicted positive values where the actual label is "False" but the predicted label is "True".
False Negatives = these are wrongly predicted negative values where the actual label is "True" but the predicted label is "False".

Results

Initially, the three models namely, Inceptionv3, ResNet50 and InceptionResNetv2 were trained on a dataset which does not contain DCGAN-generated images. The experiment is executed for 5 to 25 epoch times and the results are tabulated.

Table 1. Performance value for Resnet 50 without GAN results

Epoch	Accuracy	Precision	Recall	F1 Score
5	98.17	0.8341	0.7994	0.7803
10	97.837	0.7923	0.7243	0.7228
25	98.71	0.8642	0.8491	0.8342

Table 2. Performance values for InceptionResNetv2 without GAN results

Epoch	Accuracy	Precision	Recall	F1 Score
5	68.17	0.8374	0.6728	0.6728
10	92.42	0.9423	0.9158	0.9167
25	94.987	0.9554	0.9444	0.9475

Table 3. Performance values for Inceptionv3 without GAN results

Epoch	Accuracy	Precision	Recall	F1 Score
5	81.65	0.8517	0.8161	0.8075
10	90.09	0.9078	0.8953	0.8877
25	80.81	0.8602	0.7837	0.7837

From Table 1-3, the results of ResNet50, InceptionResNetv2 and Inceptionv3 are tabulated for 1 to 25 epochs and the values at 5,10 and 25 are recorded. It is inferred that the InceptionResNetv2 performed better than the other two models. So, now all three pre-trained models were trained on the dataset that contained DCGAN-generated images.

The experiment is executed by varying the epoch to 25 along with the balanced dataset generated through GAN. From the above tables, it is found that InceptionResNetv2 has not performed better than ResNet50 and Inceptionv3. So, comparing all these above tables, it is found that ResNet50 performed better than the other two models in terms of accuracy with a larger difference. Thus, further fine-tuning was done on the ResNet50 pre-trained model.

The accuracy of different pre-trained models trained on the new dataset was compared. It is very well observed that an accuracy above 90% was achieved by almost every model in less than 3 epochs. It was found that Resnet 50's accuracy was much better than the other two models. The precision, recall and f1 score of different pre-trained models trained on the new dataset were compared, from which it was found that both Inceptionv3 and ResNet50 both have only slight differences in all cases. Resnet50 outperforms other models in all the performance metrics and is hence found to be more suitable and efficient than others.

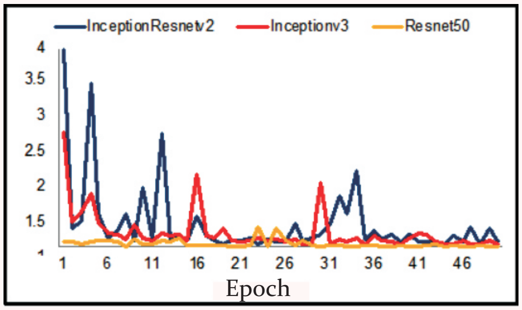

Figure-2: Loss for different models

The experiment is executed by varying the epoch to 25 along with the balanced dataset generated through GAN. From the given tables, it is found that InceptionResNetv2 has not performed better than ResNet50 and Inceptionv3. So, comparing all these above tables, it is found that ResNet50 performed better than the other two models in terms of accuracy with a larger difference. Thus, further fine-tuning was done on the ResNet50 pre-trained model.

Figure 2 represents the loss of different pre-trained models trained on the new dataset that was compared. The validation loss values are recorded to finalize the selection of the best pre-trained models. Almost, it took nearly 30 epochs for the convergence of loss for all the models. InceptionRestnetv2 has more losses ranging from 4 to 0.5. However, another evidence that again ResNet50 proved to be better than other models since it has fewer losses and on average it has losses ranging from 0.087 to 0.13.

From the above analysis, it was concluded that Resnet50 performed better than the other two pre-trained models. Thus, the following fine-tuning methods were carried out on Resnet50. The fine-tuning methods used were:

- Altering the optimizer of the model and training.
- Altering the layers of the model and training.
- Altering the learning rate of the model and training.

Table-4: Performance analysis of Resnet50 in different optimizers

Metrics	Metric	Adam	SGD	RMSProp	AdaDelta
Accuracy	Accuracy	93.610	97.721	75.121	99.778
Precision	Precision	0.9177	0.9434	0.7722	0.9479
Recall	Recall	0.9063	0.9467	0.7040	0.9417
F1 Score	F1 SCORE	0.9101	0.9444	0.6901	0.9441

1. Performance analysis of Resnet50 with different optimizers such as Adam, SGD, RMSProp, and Adadelta for epochs 50 is executed.

From table 4, the performance metrics with different optimizers in ResNet50 is evaluated and it is found that the Adadelta optimizer performs better than others by overcoming the problems like decaying learning rate. Thus, for further fine-tuning, the ResNet50 model with optimizer adadelta is being used.

2. Performance analysis of Resnet50 by adding different layers is evaluated and the values are recorded in the below table 5.

Table-5. Performance analysis of Resnet50 with different layers added.

Metrics	Model	Modification (BatchNorm)	Modification (Dropout)
Accuracy	99.778	70.434	73.097
Precision	0.9479	0.8236	0.8881
Recall	0.9417	0.7392	0.8107
F1 Score	0.9441	0.7680	0.8486

From the above table, it is found that the layers being added did not perform better than the model on which the dataset was trained.

3. Performance analysis of Resnet50 with different learning rates.

Table 6: Performance analysis of Resnet50 with different learning rates

Metric	LR(0.001)	LR(0.05)	LR(0.01)
Accuracy	99.77	96.79	98.01

Table 7: Performance analysis of proposed methodology with existing work

Work	Accuracy
Proposed Method	99.77
Arsenovic *et al.* (2019)	93.67

By examining tables 4 – 6, it is inferred that the Resnet50 with adadelta optimizer, learning rate at 0.001 and dense layer added produced 99.77% of accuracy with the precision of 94.79%, recall of 94.17% and an f1 score of 91.41%. Our proposed model has achieved better accuracy than the existing work of Arsenovic *et al.* (2019). This model can be used to improve the productivity of precision agriculture.

Finally, a computer-aided diagnostic tool to detect the presence of plant disease in precision agriculture is designed and evaluated. The performance of three different pre-trained models such as Inceptionv3, ResNet50 and InceptionResNetv2 are compared on the dataset extracted from New Plant Disease Dataset. We used DCGAN to generate more images and the dataset was trained on three pre-trained models. On comparing the three models it was found that ResNet50 performed better than the other models. Thus, after fine-tuning was done the performance of ResNet50 improved in terms of accuracy, precision, recall and F1 Score. Thus, this paper shows that ResNet50 with adadelta optimizer produced better results of 99.77% of accuracy. Future works in this model can be an integration of deep learning in the back end and android studio in the front end and leading to the development of a mobile application. This model can also be extended to train the images taken directly from the field and laboratory images and also computing time has to be investigated further.

Acknowledgement

I would like to acknowledge the undergraduate students M. Indu Priya and M.Hareni for their support. And also,I would like to thank Sri Sivasubramaniya Nadar College of Engineering for providing the NVlink GPU Server facility in the Machine Learning Research Lab.

References:

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A. & Alvarez-Gila, A. (2020): Few-shot learning approach for plant disease classification using images taken in the field. Comp. Electr. Agricul., 175:105542.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A. & Stefanovic, D. (2019): Solving current limitations of deep learning-based approaches for plant disease detection. Symmetry, 11(7):939.

Bhagat, M., Kumar, D., Mahmood, R., Pati, B. & Kumar, M. (2020): Bell pepper leaf disease classification using CNN, pp. 1-5. In: *2nd international conference on data, engineering and applications (IDEA)*. IEEE.

Chen, H.L., Yang, B., Wang, G., Wang, S.J., Liu, J. & Liu, D.Y. (2012): Support vector machine based diagnostic system for breast cancer using swarm intelligence. J. Med. Sys., 36(4), 2505-2519.

Chen, J., Chen, J., Zhang, D., Sun, Y. & Nanehkaran, Y.A. (2020): Using deep transfer learning for image-based plant disease identification. Comp. Electr. Agri., 173:105393.

Ferentinos, K.P. (2018): Deep learning models for plant disease detection and diagnosis. Comp. Electr. Agri., 145:311-318.

Fina, F., Birch, P., Young, R., Obu, J., Faithpraise, B. & Chatwin, C. (2013). Automatic plant pest detection & recognition using k-means clustering algorithm & correspondence filters. Int. J. Adv. Biotechnol. Res., 4(2):1052-1062

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014): Generative adversarial nets. Adv. Neural Info. Process. Sys., arXiv:1406.2661.

Harvey, C.A., Rakotobe, Z.L., Rao, N.S., Dave, R., Razafimahatratra, H., Rabarijohn, R. H. & MacKinnon, J.L. (2014): Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar. Philos. Transac. Royal Soc.: B, 369(1639):20130089.

He, K., Zhang, X., Ren, S. & Sun, J. (2016): Deep residual learning for image recognition, pp. 770-778. In: Proc. IEEE Conf. Comp. Vision Pattern Recog.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks, pp. 630-645. In: **European**

Ambient Science, 2022: Vol. 09(3); 34-39
DOI:10.21276/ambi.2022.09.3.ta02

Conference on Computer Vision. Pub. by: Springer, Cham.

Hitawala, S. (2018): Comparative study on generative adversarial networks. *arXiv*:1801.04271.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely connected convolutional networks. *arXiv*:1608.06993, 2.

Ji, Q., Huang, J., He, W., & Sun, Y. (2019). Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*, 12(3):51.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neu. Iinfo. Process. Sys.*, 25.

Mohanty, S. P., Hughes, D.P. & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Front. Plant Sci., 7:1419.

Nguyen, L.D., Lin, D., Lin, Z. & Cao, J. (2018): Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation, pp. 1- 5. In: 2018 IEEE Int. Symp. Circuits Sys. (ISCAS). IEEE.

Radford, A., Metz, L. & Chintala, S. (2015): Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*:1511.06434.

Simonyan, K. & Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. *arXiv*:1409.1556.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. & Stefanovic, D. (2016). Deep neural networks-based recognition of plant diseases by leaf image classification. *Comp. Intell. Neurosci*., #3289801

Strange, R.N. & Scott, P.R. (2005): Plant disease: a threat to global food security. *Ann. Rev. Phytopathol*., 43(1):83-116.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A.A. (2017): Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence. *arXiv*:1602.07261.

Szegedy, C., Vanhoucke, V., Ioffe, S, Shlens, J. & Wojna, J. (2016): Rethinking the inception architecture for computer vision, pp. 2818-2826. In: *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR).

Tiwari, D., Ashish, M., Gangwar, N., Sharma, A., Patel, S. & Bhardwaj, S. (2020): Potato leaf diseases detection using deep learning, pp. 461-466. In: *2020 4th International Conference on Intelligent Computing and Control Systems* (ICICCS) IEEE.

Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N.B. & Koolagudi, S.G. (2018): Tomato leaf disease detection using convolutional neural networks, pp. 1-5. In: *2018 eleventh international conference on contemporary computing* (IC3) IEEE.