

SE-CapsNet: Automated evaluation of plant disease severity based on feature extraction through Squeeze and Excitation (SE) networks and Capsule networks

Shradha Verma^{1,*}, Anuradha Chug¹, Ravinder P. Singh²,
Amit P. Singh¹, Dinesh Singh²

¹*University School of Information, Communication & Technology (USIC&T),
Guru Gobind Singh Indraprastha University (GGSIPU), New Delhi, India.*

²*Division of Plant Pathology,
Indian Agricultural Research Institute (IARI), New Delhi, India.*

**Corresponding author: verma.shradha@gmail.com*

Abstract

Diseases in plants harm the quantity of the overall food production as well as the quality of the yield. Early detection, diagnosis and treatment can greatly reduce losses, both economic and ecological. Intuitively, reduction in the use of agrochemicals due to timely detection of the disease, would greatly help in mitigating the environmental impact. In this paper, the authors have proposed an improved feature computation approach based on Squeeze and Excitation (SE) Networks, before processing by the original Capsule networks (CapsNet) for classification, for estimating the disease severity in plants. Two SE networks, one based on AlexNet and another on ResNet have been combined with Capsule networks. Leaf images for the devastating Late Blight disease occurring in the Tomato crop have been utilized from the PlantVillage dataset. The images, divided into four severity stages i.e. healthy, early, middle and end, are downscaled, enhanced and given as input to the SE networks. The feature maps generated from the two networks are separately given as input to the Capsule Network for classification and their performances are compared with the original CapsNet, on two image sizes 32X32 and 64X64. SE-Alex-CapsNet achieves the highest accuracy of 92.1% and SE-Res CapsNet achieves the highest accuracy of 93.75% with 64X64 image size, as compared to CapsNet that results in 85.53% accuracy. The classification accuracies of six state-of-the-art CNN models namely AlexNet, SqueezeNet, ResNet50, VGG16, VGG19 and Inception V3 are also presented for comparison purposes. Accuracy as well as precision, recall, F1-score, validation loss etc. measures have been recorded and compared. The findings have been validated by implementing the proposed approaches with another dataset, achieving similar resultant accuracy measures. The implementation was also accomplished with datasets after noise addition in six different variations, to verify the robustness of the proposed model. Based on the performances, the proposed techniques can be exploited for disease severity assessment in other crops as well and can be extended to other areas of applications such as

plant species classification, weed identification etc. In addition to improved performance, with reduced image size, the proposed methodology can be utilized to create a mobile application requiring low processing capabilities, to be installed on reasonably priced smartphones for practical usage by farmers.

Keywords: Capsule networks; convolutional neural networks; deep learning; plant disease severity; squeeze and excitation networks.

1. Introduction

Protecting plant health is crucial to terminate hunger, decrease poverty and boost the economic as well as ecological developments. According to FAO (Food and Agriculture Organization of the United Nations), year 2020 has been declared as the International Year of Plant Health (IYPH), emphasizing on the importance of protecting plants from pests and diseases, which result in around 40% loss of food crops. Food security is one of the key agendas in 2030 Sustainable Development Goals (SGDs). Also, recently a financing facility was launched by the Government of India, with an overall budget of Rs One Lakh Crore for refining agricultural infrastructure and providing a comprehensive platform to agri-entrepreneurs. Evidently for the past several decades, the agronomists as well as the research community has strived to find solutions for phytopathological complexities and pave the way for real-time detection of any anomalies found in a crop, and also to minimize the use of agrochemicals in the fields. In light of this, early detection becomes vital for maintaining ecological wealth. In numerous studies, machine learning techniques have been applied for plant disease classification and/or exploring properties of a plant, leading to a better understanding of the cause, such as applying SVM (Support Vector Machines) on Hyperspectral reflectance i.e. spectral vegetation indices for early detection in sugar beet leaves (Rumpf *et al.*, 2010), PLSR (partial least-squares regression) on leaf reflectance for predicting crop photosynthetic capacity of cabbage and corn (Heckmann *et al.*, 2017), SVM for predicting diseases resistive proteins in crops (Pal *et al.*, 2016), decision tree applied on intensity-based features for disease recognition in Tomato (Sabrol & Kumar, 2016) etc. In practical scenarios, visual inspection by agronomists and laboratory-based experiments are used to identify the symptoms and stage of the disease. The process is time-consuming and labour intensive, not to mention costly for the growers, with the rising severity of the disease as time passes and the eventual use of expensive pesticides. Also, due to a myriad of plant species and their diseases, even qualified and expert pathologists might conclude with a faulty diagnosis leading to an ineffective treatment. Hence, automation through the utility of computer vision and image processing techniques would play a vital role in effective disease recognition approaches. Image segmentation with the help of genetic algorithms (Singh & Misra, 2017), feature extraction and selection from leaf images, based on texture, shape, dispersion etc. (Camargo & Smith, 2009), utilizing depth information along with local and global feature statistics extracted from thermal and stereo visible light images (Prince *et al.*, 2015), image colour space transformation, image enhancement, segmentation and extracting ROI (region of interest) to identify diseased area (Camargo & Smith, 2009), are a few examples.

Computer Vision-based approaches aim at finding different representations that capture prominent features which in turn help in improving classification performance (Al-Hmouz, 2020). This paper aims at assessing disease severity in tomato crop with the help of Capsule networks, along with certain modifications in the feature computation phase. This approach has never been studied/proposed for plant disease severity estimation or any other application as per our research.

The contributions of this research article are listed as follows:

- Analysis of the original CapsNet architecture implemented for plant disease severity assessment.
- Implementing feature extraction through the use of squeeze and excitation (SE) networks, instead of a single convolution layer as in the Original CapsNet architecture.
- Exploring the capabilities of the two proposed SE-CapsNet architectures, SE-Alex-CapsNet, based on SE networks applied with operations similar to layers in AlexNet and SE-Res-CapsNet, based on SE networks applied with operations similar to ResNet layers, for estimating the plant disease severity.
- Investigating the performances of both architectures i.e. SE-Alex-CapsNet and SE-Res-CapsNet on multiple datasets.

The organization of the paper is as follows: Section 2 details the related works from the previous studies; Section 3 discusses the background of the proposed techniques; Section 4 lists the materials and methods wherein data selection, data preparation, basic methodology and the proposed techniques with the help of flowcharts are explained; Section 5 presents the experimental results along with graphs and discussion of the findings; and Section 6 outlines the conclusion and future scope.

2. Related Works

Among several advanced technologies, deep learning has proved to be a powerful and most effective tool for plant disease identification (Kamilaris & Prenafeta-Boldu, 2018), specifically convolutional neural networks (CNN) have been extremely successful with high accuracy measures in classifying crop diseases. In 2016, (Mohanty *et al.*, 2016) trained two deep learning models namely AlexNet and GoogLeNet on the leaf images from PlantVillage dataset (Hughes & Salathe, 2015) and achieved an accuracy of 99.35% with GoogLeNet. The authors implemented the models in both approaches i.e. transfer learning and training from scratch, for coloured, segmented and grayscale version of the images, in varying train : test ratios viz. 80:20, 60:40, 50:50 etc. This was the first study conducted where pre-trained CNNs were implemented on a large leaf image dataset to classify plant diseases. Another study demonstrated the capability of a network designed and trained from scratch to identify 22 crop and weed species. The images taken were via a mobile phone and had variations in lighting conditions and soil types (Dyrmann *et al.*, 2016). (Sladojevic *et al.*, 2016) executed CaffeNet and showcased the capability of augmentation process for dataset expansion along with achieving promising results for recognizing 15 classes. In 2017, (Wang *et al.*, 2017) studied the application of computer vision for assessing disease severity in a crop. They implemented four pre-trained CNN architectures namely VGG16, VGG19, Inception V3 and ResNet50, to diagnose the apple black rot disease severity from PlantVillage leaf images, and the performance varied between 80% to 90%. The best result of 90.4% was achieved with VGG16. The authors also concluded that transfer learning performs better when the number of training images available are low. (Ferentinos, 2018) executed five CNN models, AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat and VGG, on 87,848 leaf images of 25 distinct species and 58 pairs of different plant-disease groupings, with images taken in both controlled laboratory environment and field conditions. The best performance achieved was 99.53% in recognizing the correct plant-disease pair. The author also demonstrated the classification results when training with laboratory based images and testing being done on field images and

vice versa, with diminishing performance measures. (Too *et al.*, 2019) also evaluated and compared the performances of six pre-trained CNN architectures VGG16, ResNet50, ResNet101, ResNet152, Inception V4 and DenseNet via transfer learning approach, applied to the images from PlantVillage dataset and emphasized on the impact of fine-tuning the architectures for individual executions. DenseNet resulted into 99.75% accuracy, beating the rest. (Barbedo, 2019) utilized distinct lesions/spots on a leaf instead of an entire leaf for disease identification, arguing that it would increase the data size without additional images and more significantly, multiple diseases affecting the same leaf can be identified. GoogLeNet was implemented through transfer learning, achieving accuracies between 75% -100% in a series of experiments on original, modified (background removed) and manually expanded (based on symptoms and area coverage) image dataset.

An addition to various standard CNN architectures being exploited for plant species and disease identification, several studies have proposed their modified and improved versions, with better performances. (Zhang *et al.*, 2018) experimented by means of different pooling combinations, dropout percentages and ReLU operations at multiple stages with GoogLeNet and Cifar10 models. (Liu *et al.*, 2018) presented an end-to-end architecture comprising of AlexNet concatenated with two cascaded Inception modules at the end, replacing the first two FC (fully connected) layers. (Hang *et al.*, 2019) combined the structures of VGG16 (as the base), Squeeze and Excitation (SE) network and Inception module, replacing FC layer with global average pooling, which resulted into a reduced number of parameters and better classification accuracy. (Dong *et al.*, 2019) utilized Capsule networks (CapsNet) (Sabour *et al.*, 2017) for disease identification in peanut crop, whereby three convolutional layers were placed with the original CapsNet and performances compared with a standard CNN. CapsNets have also been implemented for hyper-spectral image classification (Paoletti *et al.*, 2018) and identification of UAV imagery (Li *et al.*, 2019). (Liang *et al.*, 2019) designed an image-based architecture PD²SE-Net for plant recognition, disease identification and disease severity assessment. The authors utilized ResNet50 as the base model along with parameter sharing between layers and shuffle blocks, enabling the three functionalities to be carried out with three separate FC layers and softmax function. More recently, (Chen *et al.*, 2020) combined the capabilities of VGG19, Inception module and global pooling via transfer learning, and created a hybrid structure to classify rice and maize crop diseases, from an enriched and augmented leaf image dataset. (Karlekar & Seal, 2020) proposed an image segmentation algorithm followed by SoyNet, a CNN model for soybean plant disease recognition. The authors argue that background subtraction and a reduction in the spatial resolution of the image with each layer can significantly enable the CNN to learn crucial features and thus improve the accuracy. (Zeng & Li, 2020) proposed SACNN, a basic CNN to extract global features with the aid of Residual Networks, followed by a self-attention (SA) network for extracting local and intricate features in later layers. Similarly, few other researchers have utilized CapsNet for disease recognition in potato crop (Verma *et al.*, 2020), newly developed CNN models EfficientNet and MixNet for automated recognition of fruits (Duong *et al.*, 2020), features extracted via pre-trained CNNs exploited to train traditional machine learning techniques such as SVM, Logistic Regression and XGBoost (Espejo-Garcia *et al.*, 2020) etc.

3. Background

3.1 Capsule networks

In the field of agriculture, CNN models have been effectively applied for plant species as well as weed identification, fruit counting, plant disease diagnosis, overall yield estimation etc. However, certain drawbacks such as discounting the spatial relationship between

various features in an image (such as relationship between eyes, nose and mouth in a face), instability towards affine transformations such as rotation, translation, scaling etc., have been identified, which are the main cause for the requirement of humungous dataset, covering all possible orientations of objects in images, for training. This leads to increased computational cost in terms of resources as well as training time. The pooling operation in CNNs is known to lose spatial information, rendering CNNs to be invariant rather than equivariant. In 2017, (Sabour *et al.*, 2017) investigated Capsule networks with dynamic routing between capsules, in order to conquer these flaws. Capsule networks have the capability to encode spatial data and distinguish between various textures, orientations and poses. A capsule is a group of neurons, whereby each capsule has an activity vector associated with it, that captures numerous instantiation parameters for detection of a particular type of object or its part. The length and orientation of the vector presents the likelihood or the probability of presence of that object and its generalized pose. These vectors are passed on to the upper level capsules from lower layer capsules. Coupling coefficients exist between these layers of capsules. As the name suggests, if the prediction by the lower level capsule matches the output of the current capsule, the value of the coupling coefficient between them increases, computed through the use of softmax function. That is to say, if the current capsule detects a tight cluster of previous predictions, strongly indicating the presence of that object, it results in a high probability, also known as routing by agreement.

Firstly, the prediction vector (equation 1) is computed as,

$$\hat{u}_{j|i} = W_{ij}u_i, \quad (1)$$

where $\hat{u}_{j|i}$ is the output/prediction vector of the upper-level j th capsule, W_{ij} and u_i are the weight matrix and prediction vector of capsule i in the lower layer respectively. It may capture spatial relationships and relation between sub-objects and objects. As in equation (2), based on the degree of agreement between adjoining layer capsules, the coupling coefficients are computed via the softmax function,

$$c_{ij} = \exp(b_{ij}) / \sum_k \exp(b_{ik}), \quad (2)$$

where b_{ij} is the log probability between the two capsules, initialized to zero and k denotes the number of capsules. The input vector s_j to the j th layer capsule, which a weighted sum of $\hat{u}_{j|i}$ vectors learned by the routing algorithm, is calculated as follows,

$$s_j = \sum_i c_{ij} \hat{u}_{j|i}, \quad (3)$$

Finally, a squashing function that combines squashing and unit scaling (equation 4), is applied to confine the value of the output in the range between 0 and 1, hence computing the probability as,

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}, \quad (4)$$

The loss function (as computed by equation 5) is associated with capsules in the last layer, where m^+ and m^- are set to 0.9 and 0.1 resp.

$$l_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2, \quad (5)$$

where the value T_k is 1 for correct labels and 0 otherwise, λ is a constant whose value is 0.5. The first term is computed for correct labels, the second term computes for incorrect labels. If T_k is 1, the second term becomes 0 and for T_k as 0, the first term becomes 0. Also, the loss value l_k will be 0 for correct predictions with v_k being greater than 0.9 and non-zero otherwise.

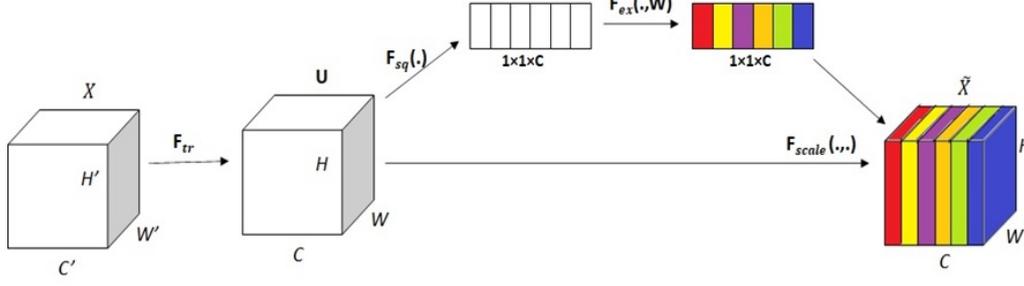


Fig. 1. Basic SE Network

3.2 Squeeze and Excitation Networks

CNNs exploit the convolution layer to generate feature maps from images, where several filters are used for extraction of hierarchical data. Lower layers identify edges, curves etc and higher layers are able to recognize faces, objects, text and so on. This task is accomplished by aggregating the spatial as well as channel-wise data of the image at every layer. Various filters are used to find specific spatial data in all separate channels, which is later combined and added across all output channels available. Hence, the feature maps are created as a result of network weights being equal for all channels. Squeeze and excitation networks (SENet) proposed by (Hu *et al.*, 2018), which also won first place at ILSVRC 2017, explores the interdependencies and relationship between the channels to improve performance. The authors introduced a simple yet effective building block that can be easily added to any CNN model without much increase in the computational cost. Figure 1 demonstrates the basic structure of the SE block. The aim is to scale each channel based on its significance rather than giving them the same importance by assigning equal network weights. Given the convolutional block and present number of channels as input, firstly, the feature maps are squeezed into a single numeric value by employing global average pooling (GAP). This is done for all feature maps. Hence, the resultant vector ‘n’ has size equal to the number of current channels. Then a fully connected (FC) layer followed by ReLU operation is used to introduce non-linearity. There is a second FC layer followed by Sigmoid function, which applies a smooth gating function to each channel. Lastly, the resultant vector of size ‘n’ is multiplied to the original convolutional block which was initially given as input, scaling the channel-wise data based on their respective importance. Thus, the feature maps are passed through a squeeze operation which generates a channel descriptor by combining the spatial data, then excitation operation is performed which results into channel-wise measured and proportional weights. The input image X is mapped into a set of feature maps $U \in R^{H \times W \times C}$ via some transformation represented by F_{tr} , where $H \times W$ denote the spatial dimensions and C is the number of channels. Firstly, the squeeze operation $F_{sq}(\cdot)$ is applied to U (equation 6), generating $z \in R^C$, that aggregates the feature maps throughout its spatial dimensions and produces a channel-wise descriptor of dimension $1 \times 1 \times C$, the aggregation function being global average pooling.

$$z_c = F_{sq}(u_c) = 1/(H \times W) \sum_{i=1}^H \sum_{j=1}^W u_c(i, j), \quad (6)$$

For a feature map u_c , corresponding to a specific channel c , i increments from 1 to H and j increments from 1 to W , adding all values of $u_c(i, j)$. The total sum divided by $H \times W$ gives the value of global average pooling, i.e. the entire feature map is aggregated to a single value denoted by z_c . In order to completely utilize the channel-wise information detected by the squeeze operation, excitation operation $F_{ex}(\cdot, W)$ is applied (equation 7) via the fully connected

layers and activation functions. Its purpose is to capture non-linearities amongst the channel interactions as well as to stress on multiple channels. Two FC layers are placed here in order to achieve that, first one followed by ReLU operation to introduce non-linearity, where the complexity is reduced by a certain amount governed by the value r (denotes the reduction ratio), followed by the second FC layer and sigmoid function for a smooth gating mechanism.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \quad (7)$$

where δ represents the ReLU operation, $W_1 \in R(\frac{C}{r} \times C)$ and $W_2 \in R(C \times \frac{C}{r})$. The final output is obtained by implementing the scaling operation (equation 8) that applies rescaling to U with respect to activations s .

$$\hat{x}_c = F_{scale}(u_c, s_c) = s_c u_c. \quad (8)$$

4. Materials and Methods

4.1 Dataset

Tomato crop (aka *Solanum Lycopersicum*) is a key dietary source of fiber, nutrition, vitamins and minerals, having various health benefits. It is a fruit, being consumed mostly as a vegetable. There is a gamut of viral, bacterial and fungal diseases affecting its growth and overall yield. Late Blight disease (Nowicki *et al.*, 2013) in Tomato plant, caused by the fungus *Phytophthora infestans*, is a fast spreading and devastating disease. The condition of the infected plant deteriorated rapidly, spreading and destroying entire fields, if not diagnosed and controlled on time. For this study, the late blight infected leaf images have been selected from the Plant Village dataset, mentioned previously. It is an openly available dataset that has been utilized by several previous studies, due to its large collection of leaf images (over 50,000), covering 14 crops including tomato, and their corresponding 26 diseases, leading to 36 crop-disease pairs, which deems ideal for training a CNN considering image-based identification. The coloured leaf images were categorized into four classes: healthy, early, middle and end stage of disease. Figure 2 showcases few samples of the images selected for each stage and table 1 summarizes their count. The selection process was manually conducted and disease severity was assessed by the visibly damaged area coverage of the infected leaf (Campbell & Neher, 1994) (as shown in figure 2). Leaf image with few and smaller spots was selected for early stage, an almost destroyed and dried leaf was considered for end stage, and leaves with few but larger spots were selected for middle stage. Healthy tomato leaves are available separately in the PlantVillage dataset.

Table 1. Total images selected in each category

Categories	No. of Images
Early Stage	354
Middle Stage	347
End Stage	382
Healthy	433

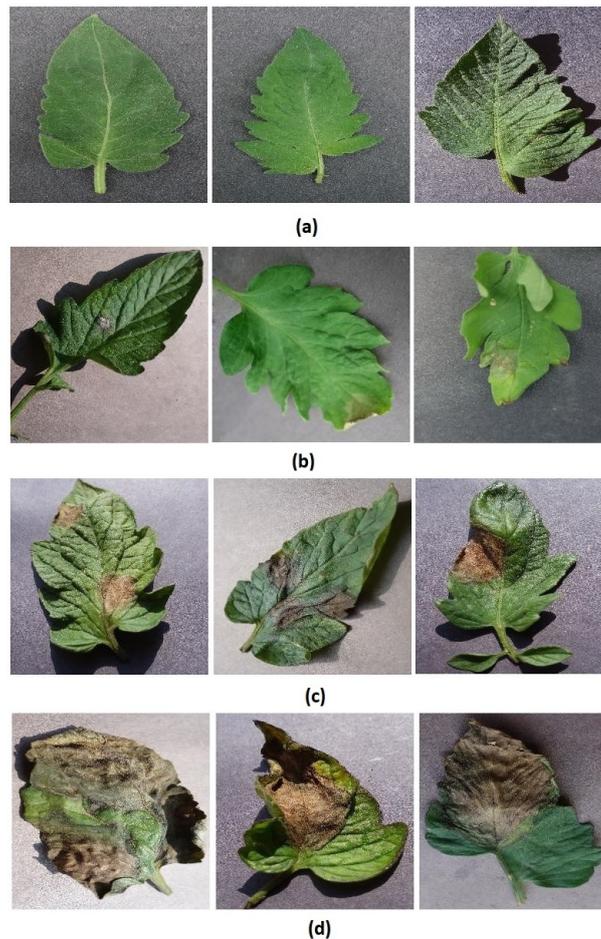


Fig. 2. Sample Images of (a) healthy, (b) early, (c) middle and (d) end stages of late blight disease in tomato crop, selected from PlantVillage dataset.

After the finalization of the dataset, all images were downsized in order to fasten the training process, reduce parameters and overcome hardware restrictions. The images were downscaled to two sizes, 32X32 and 64X64. The resulting images were enhanced by utilizing the block wise DCT (Discrete Cosine Transform) scaling method (Mukherjee & Mitra, 2008). DCT has previously been adopted for JPEG compression standards. Brightness, contrast and colour are the three essential blocks while displaying an image. Most of the past studies have worked upon the brightness, or contrast, or a combination of both to enhance images, but none of them contemplated colour preservation in the resulting image. The abovementioned image enhancement technique proposed to consider all three factors for compressed images. The approach revolves around uniform scaling of DCT coefficients in a block. The original RGB image is transformed into YCbCr colour space. Thereafter, the algorithm operates in three steps: adjustment of the background illumination by mapping the current brightness values (DC coefficients) in a block, to values in the chosen range; preserving the local contrast by scaling the pixel values in a block with the same factor, however, overflow of values outside the range needs to be prevented; and lastly preserving the original colours as a consequence of the previous step, wherein uniform scaling of brightness and contrast also scales the colour vectors pertaining to RGB space, with the same factor. Hence, the enhancement technique not only processes the luminance components but also chrominance components for better quality images. The image pre-processing has been depicted in figure 3.

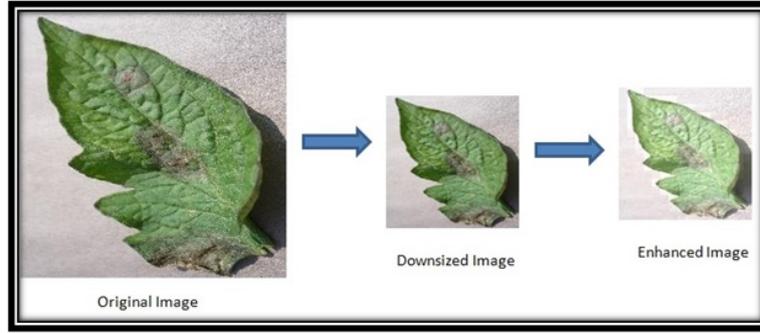


Fig. 3. Data Pre-processing.

4.2 Methodology and Implementation Details

Figure 4 portrays the flowchart of the basic steps and methodology used for this study, and figure 5 highlights the feature extraction phase in the implementation. As discussed previously, images from the original dataset were selected, for all late blight disease severity stages, along with healthy tomato leaf images. This image dataset was downscaled and enhanced for further processing. The labelled dataset constituting of four classes was then divided into train: validation data and fed as input to the SE networks, in order to derive feature maps, as opposed to the original CapsNet architecture having a single convolution layer. Thereafter, the resulting feature maps were processed by the primary capsule layer for further classification. The hyperparameters viz. learning rate, epochs, batch size etc. were varied within a range and the models were trained for most of the combinations. For eg. learning rate was varied between 0.01-0.0001, batch size was varied between 8, 16, 20 and 32. Similarly, epochs were varied between 30, 40 and 50. The values achieving highest performance measures were selected. All implementations were performed with 50 epochs and the mini batch size was set to be 16. Most common train:test ratio is 80:20 i.e. 80% of the data is used for training and the remaining 20% is used for testing. Hence, it was also set to 80:20. Table 2 describes the hyperparameters used for training. The Coefficient for the loss of Decoder and the Reduction ratio r (mentioned in the table for both architectures), were also selected empirically. The value r is used to control the computational cost for the SE block. It was varied between two values i.e. 8 and 16, previously investigated in (Hu *et al.*, 2018). The Keras implementation adapted from (GITHUB. <https://github.com/XifengGuo/CapsNet-Keras>, 2020) was performed on a workstation equipped with GTX1060 6GB GPU, using Python 3.7. The image pre-processing was implemented on MATLAB R2019b.

4.3 SE-CapsNet

The original CapsNet architecture comprises of a single convolution layer having 256 filters of 9X9X1 size. The features detected are analysed by the capsules in the subsequent layers. Two SE-CapsNet architectures have been proposed in this paper, namely, SE-Alex-CapsNet which is based on the basic operations of AlexNet with SE blocks, and SE-Res-CapsNet, which is based on the basic operations of ResNet with SE blocks, followed by the encoder and decoder operations of Capsule networks. AlexNet (Krizhevsky *et al.*, 2012) is a hugely powerful network capable of a high success rate with complex datasets. Its architecture is one of the foremost in several computer vision applications such as object detection. AlexNet has five convolution layers and the CNN architectures that came after, such as VGG and GoogLeNet were deeper with 19 and 22 layers, with the intent of learning more complex features. However, deeper networks, not to mention, are harder to train, also face the vanishing gradient problem, whereby the gradients that are back propagated to earlier layers end up being infinitely small due to the multiplication operation, thereby saturating or deteriorating the performance.

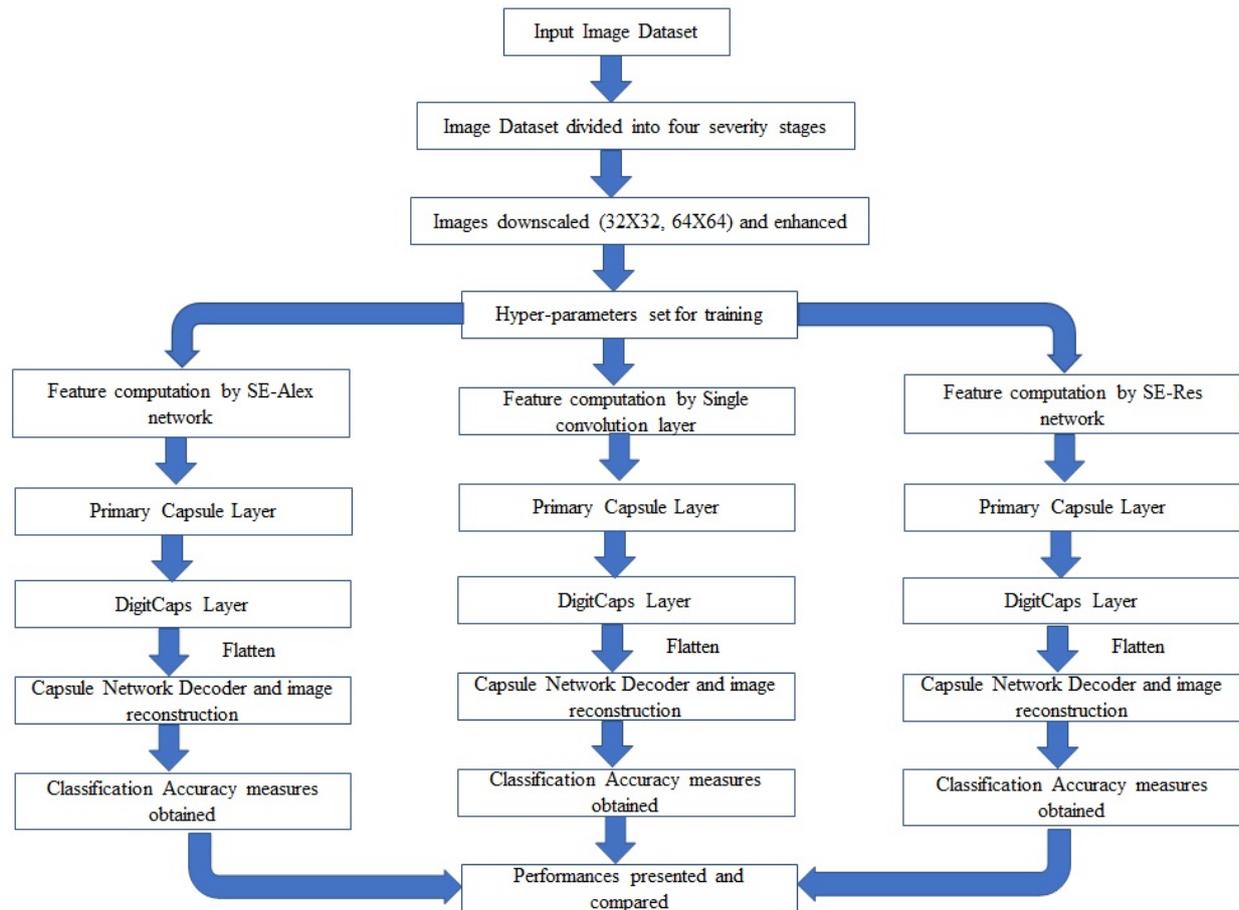


Fig. 4. Research Methodology

In order to tackle it, ResNet (He *et al.*, 2016) introduced ‘residual blocks’, which are identity shortcut connections that skip few layers (figure 6). The output of the previous layer is added to the next layer, with the idea to reuse the previous activations. Figure 7 lists the basic operations performed by the SE block on the input convolutional block, the output of which is multiplied with the input block for applying the scaling effect. The layers in SE-Res-CapsNet are demonstrated in figure 8 and figure 9. For simplifying the flow of operations, it has been divided in two parts, figure 8 depicting the core layers in SE-ResNet, which are embedded in figure 9 i.e. Block SE-ResNet 1 through 5, for final computation, later given as input to the CapsNet. Similarly, the layers in SE-Alex-CapsNet are showcased in figure 10, utilized to compute the feature maps, which are then processed by the primary capsule layer in the CapsNet. The kernel size in the primary caps layer was reduced to 2. Four convolution layers with 64, 128, 256 and 512 filters respectively, are implemented to enhance the feature extraction process of the network, followed by the SE block operations, batch normalization, ReLU and MaxPool applied to the input image. Dropout layer was added to both proposed models in order to minimize the overfitting. The processing by the CapsNet Encoder layers is explained as follows:

- **Primary Capsule Layer:** As discussed in section 2.3, this is the lower level capsule layer, constituting 32 distinct capsules. It detects more detailed features and patterns. The features detected by the previous layers are analyzed in this layer, where each capsule

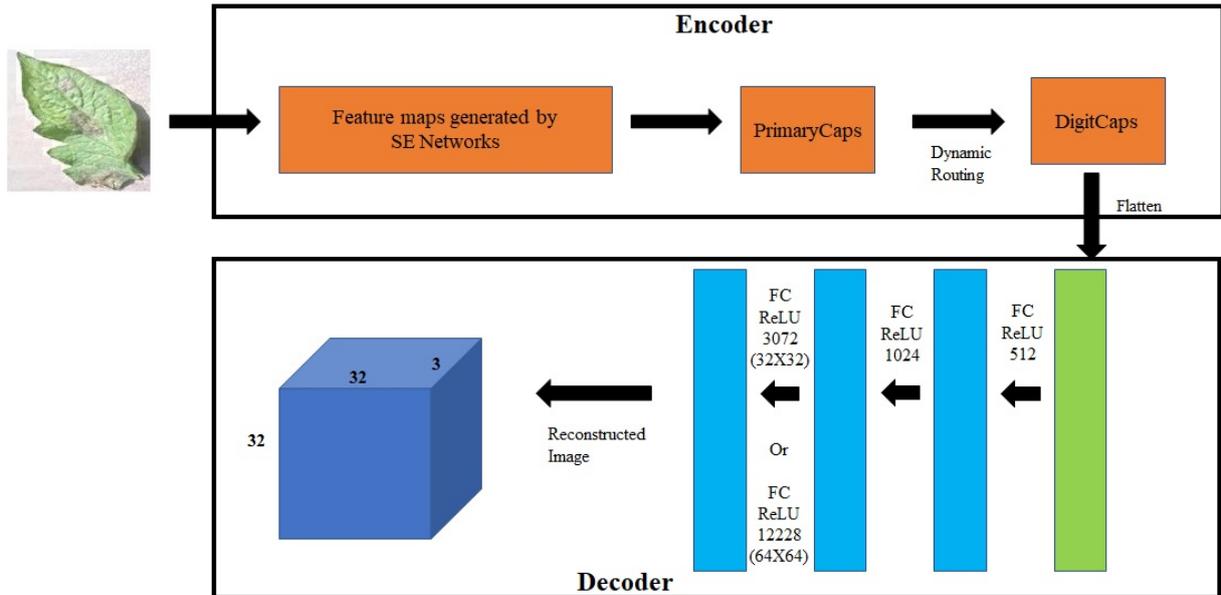


Fig. 5. Feature Extraction through SE Networks before processing by CapsNet.

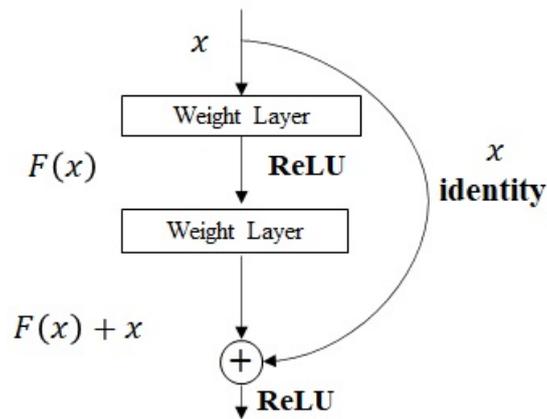


Fig. 6. Residual Block

applies eight $9 \times 9 \times 256$ convolutional kernels to the input, producing a 4D vector as output.

- DigitCaps: This is the higher level capsule layer resulting in 16D vectors containing all the instantiation parameters.

The 16D vector is given as input to the CapsNet Decoder network, which is basically a feed forward neural net, having three fully connected layers with 512, 1024, 3072 (for 32×32 images) or 12228 (for 64×64 images) neurons, each followed by ReLU operation. It also acts as a regularize. It learns to decode the instantiation parameters into the image it is supposed to detect. It utilizes Euclidean distance loss function for learning and training.

Table 2. Hyper-parameters set for training

Optimizer	Adam
Learning rate	0.001
Learning Rate Decay	0.9
Epochs	50
Train : Validation Ratio	80:20
Batch size	16
Ratio r (For SE-Alex-CapsNet)	8
Ratio r (For SE-Res-CapsNet)	16
Coefficient for the loss of Decoder	0.392

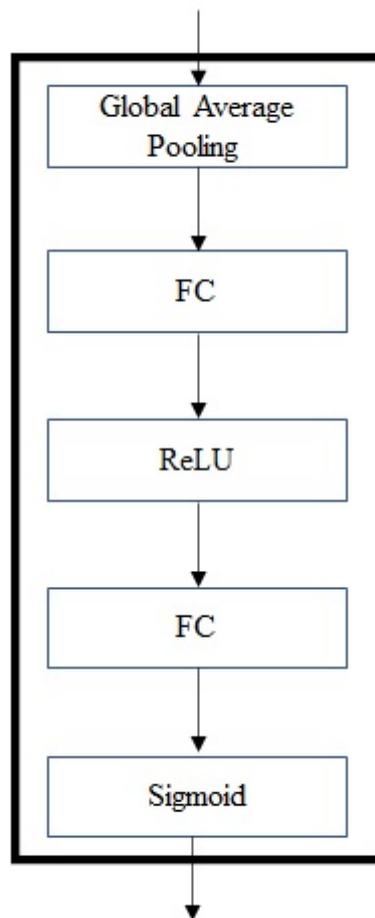


Fig. 7. Operations of SE Block, the output of which is multiplied to the input, for further processing

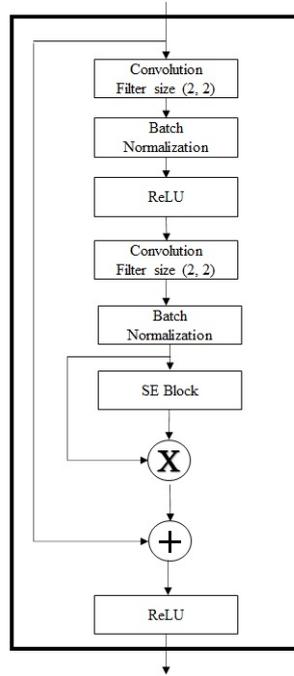


Fig. 8. Block for ResNet with SE.

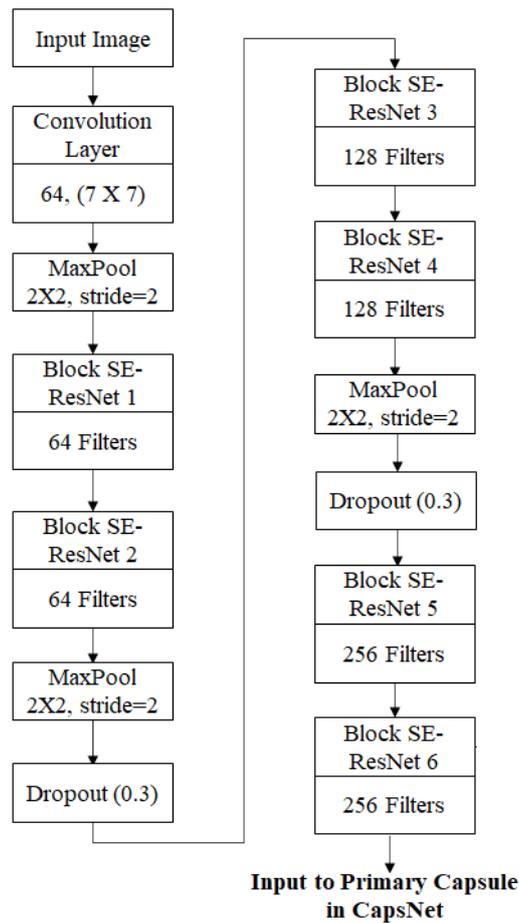


Fig. 9. Basic flow of SE-Res-CapsNet

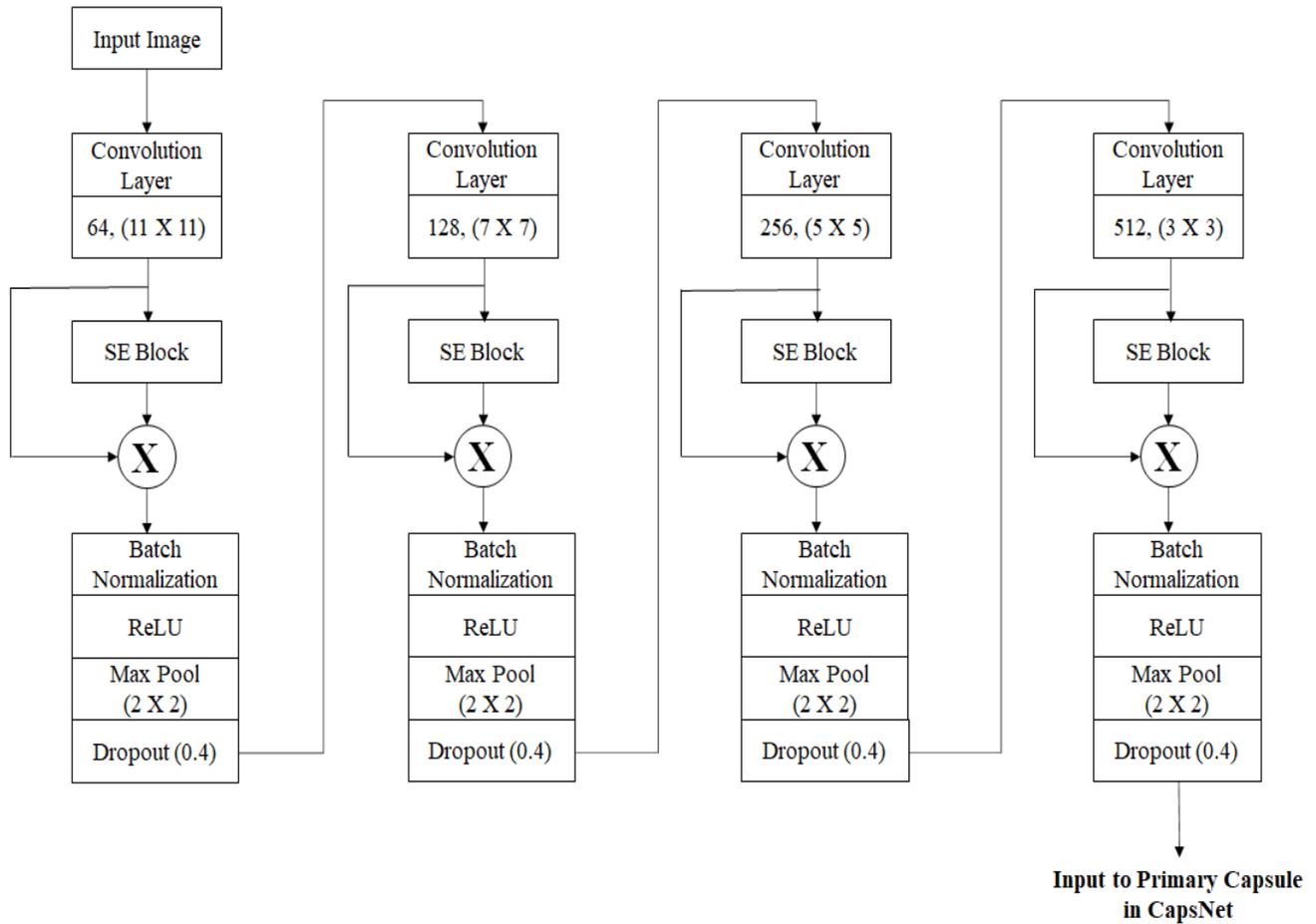


Fig. 10. Basic flow of SE-Alex-CapsNet

5. Results and Discussion

For evaluating the performances of the proposed techniques, the authors implemented the above mentioned approaches to the images selected from the PlantVillage dataset. The performance of the proposed architectures was compared with six pre-trained CNN models, implemented via transfer learning. The metrics considered for comparison purposes were accuracy, precision, recall, F1-score and validation loss. Firstly the implementation was completed with tomato late blight images and performance measures were recorded. Then, the approaches were implemented with another dataset namely grape black rot images, taken from PlantVillage dataset as well, in order to validate the findings. Lastly, implementation was performed after various percentages of Salt & Pepper noise were added to the original images in both the datasets, with image size 64X64 and accuracy as well as F1-score were recorded.

5.1 Impact of proposed feature extraction via SE networks

Six CNN models namely, AlexNet, SqueezeNet, ResNet50, VGG16, VGG19 and Inception V3 were also implemented on our dataset. Figures 11 – 12 demonstrate the resultant chart comparing the test accuracies obtained with each implementation along with the results of our proposed technique. It can be observed that both our approaches perform better in comparison, with SE-Res-CapsNet achieving the highest accuracy of 91.44% and 93.75% on the test set for both image sizes. Tables 3 and 4 present the resulting values for all experiments, performed on images sizes 32X32 and 64X64 respectively. Even in terms of precision, recall and F1-score, our approaches outperform other implementations, along with achieving the minimum values for validation loss. Original CapsNet is treated as the baseline architecture for our proposed techniques. It requires much less data to train as compared to traditional CNNs and can generalize successfully, leading to better performance measures, as in our case. Evidently, disease severity image data was selected from a mix of images manually and count in each class is in few hundreds. Images in PlantVillage dataset have an identical background, suitable for Cap-sNet implementation as it can't deal with complex backgrounds. As depicted by the improved results, SE-CapsNet captures the advantages of the SE networks along with the underlying CNNs and the Capsule networks. The first proposed network is based on the sequential layered processing of AlexNet, enhanced with the capabilities of SE network that captures the complex features within the dataset having high inter-class similarity. The SE block is placed after every convolution layer. It doesn't employ any transfer learning as the weights are assigned randomly, hence, the network is trained from scratch specifically for disease severity evaluation. Similarly, the next proposed network was based on ResNet, having identity mappings or skip connections, where SE block is applied systematically in every ResNet block. It acts as the feature extractor that results in a better performance in terms of predicting class of diseased leaf images. The effectiveness of adding SE blocks to ResNet have been investigated in (Hu *et al.*, 2018). Intuitively it can be observed that deeper networks with channel-wise scaling in feature maps results in better classification accuracy. Optimization is achieved by employing Adam with learning rate initially set as 0.001. Results obtained in (Wang *et al.*, 2017) for disease severity estimation in apple rot disease utilizing CNNs depicted the classification accuracy in the range of 80% -90%. With original CapsNet, implemented on our dataset, the accuracy achieved was 83.22% for 32X32 image size and 85.53% for 64X64 image size. As showcased with the results, both proposed models outperform CapsNet as well as pre-trained CNN models with better classification accuracy and hence, it can be concluded that the proposed approaches can prove to be effective in plant disease severity assessment and can be extended to other crops or other areas of application. Also, due to reduced image size, it can be implemented in low power devices for practical use by the growers. Table 5 presents the comparative values of Kappa coefficient as well as loss values related to CapsNet and table 6 lists the count of parameters in each implementation.

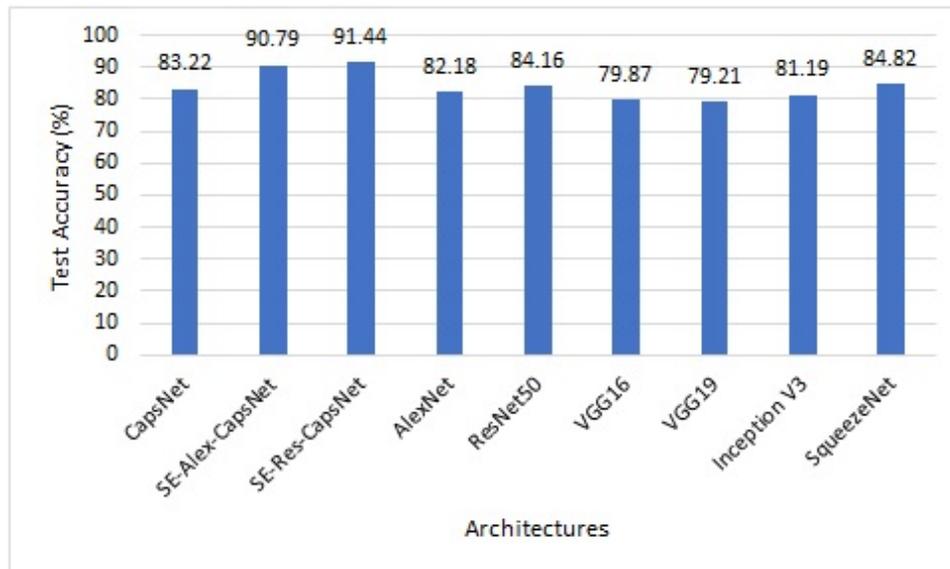


Fig. 11. Accuracy comparison for the implemented models for Tomato Late Blight Images size 32X32.

It can be observed that with improved Kappa coefficient and decreased losses as well as parameters, both our approaches present an enhanced model for implementation on such use cases. Figures 13 – 18 showcases the resultant accuracy and loss graphs for both image sizes i.e. 32X32 and 64X64. CapsNet also reconstructs the input image through the features extracted and learned representations during the earlier stages, in order to improve accuracy measures and compute loss. It is possible to do so due to the equivariance property as well as pose and orientation data captured in the activity vector. Figure 19 visualizes the sample real and reconstructed images after the CapsNet implementation. Afigure 20 demonstrates the test accuracy comparison of the state-of-the-art CNN architectures after 20 epochs, 80:20 split ratio and Adam employed as the optimizer, implemented via trans-fer learning on the original dataset, with image size 256×256 , for comparative purposes. As can be observed, the resultant classification accuracies are restricted between 80% - 90%, due to high interclass similarities and fewer images available for training. With reduced image size, the performances would diminish even further. The proposed methodology improves the perfor-mance substantially as compared to standard CNN models, as it benefits from the channel-wise scaling attributed to SE networks and also, doesn't require huge dataset for training and clas-sification due to the utilization and benefits of CAPSule networks. With a different dataset, the accuracy measures might differ but the performance of the proposed models, showcased for the two different data sizes were consistent. Hence, in order to validate our findings, the authors also implemented the proposed approaches on a different dataset, results of which are presented in the next section.

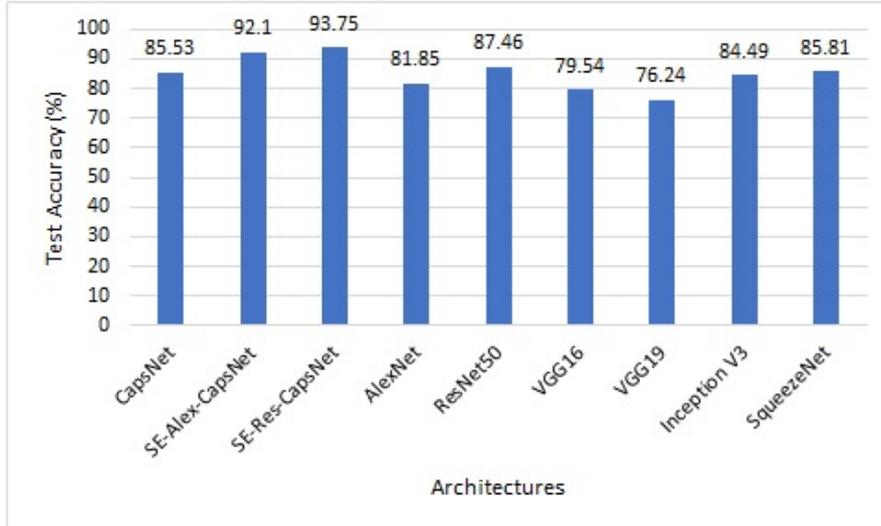


Fig. 12. Accuracy comparison for the implemented models for Tomato Late Blight Images size 64X64.

Table 3. Resultant Precision, Recall, F1-score and Validation Loss for the Tomato Late Blight Image size 32X32

Networks	Precision	Recall	F1-score	Validation Loss
Original CapsNet	0.82	0.82	0.82	0.1383
SE-Alex-CapsNet	0.9046	0.9054	0.9049	0.0794
SE-Res-CapsNet	0.9063	0.904	0.9051	0.08
AlexNet	0.8248	0.8138	0.8193	1.3537
ResNet50	0.8286	0.8299	0.8292	0.7525
VGG16	0.7916	0.7962	0.7939	0.9561
VGG19	0.7893	0.7913	0.7903	0.5428
Inception V3	0.8052	0.8032	0.8042	0.7859
SqueezeNet	0.8432	0.8389	0.8411	0.6766

Table 4. Resultant Precision, Recall, F1-score and Validation Loss for the Tomato Late Blight Image size 64X64

Networks	Precision	Recall	F1-score	Validation Loss
Original CapsNet	0.851	0.8512	0.8511	0.1112
SE-Alex-CapsNet	0.9179	0.9136	0.915	0.0691
SE-Res-CapsNet	0.9286	0.9274	0.928	0.06
AlexNet	0.8038	0.805	0.8044	1.095
ResNet50	0.8667	0.8662	0.8664	0.5807
VGG16	0.7837	0.7839	0.7838	0.7351
VGG19	0.7581	0.7572	0.7576	0.6688
Inception V3	0.8479	0.8392	0.8435	0.7942
SqueezeNet	0.8583	0.8526	0.8554	0.5728

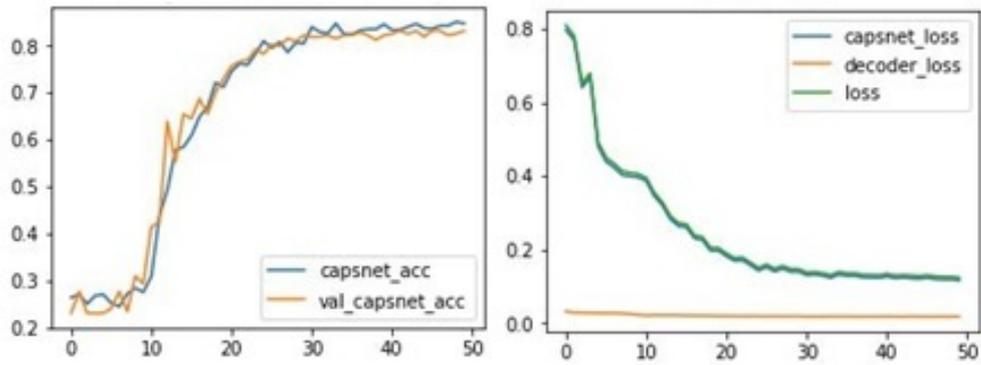


Fig. 13. Graphs depicting the training accuracy, validation accuracy and training loss for Original CapsNet for image size 32X32.

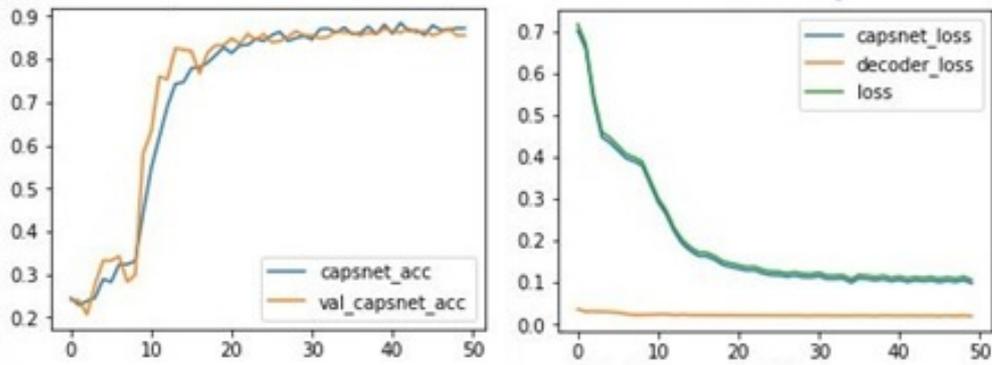


Fig. 14. Graphs depicting the training accuracy, validation accuracy and training loss for Original CapsNet for image size 64X64.

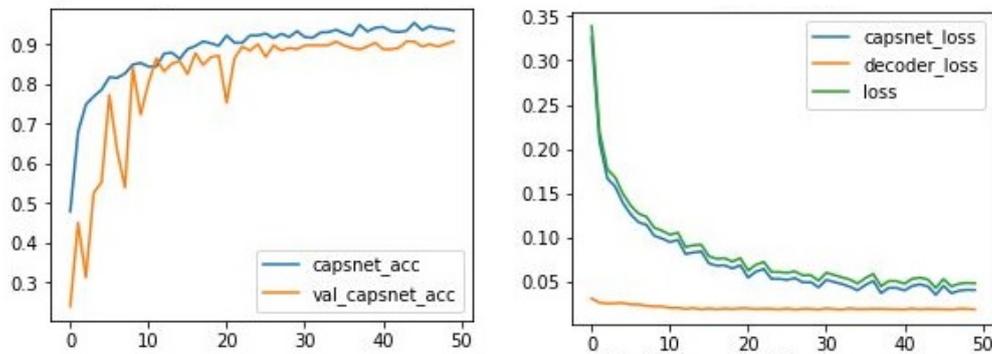


Fig. 15. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Alex-CapsNet for image size 64X64.

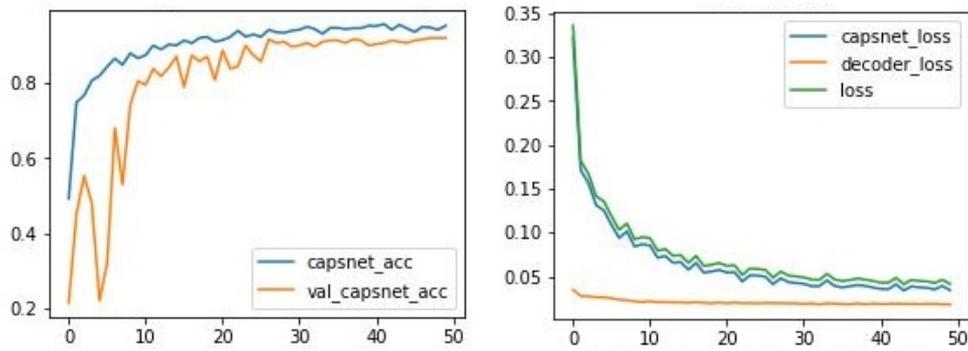


Fig. 16. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Alex-CapsNet for image size 64X64.

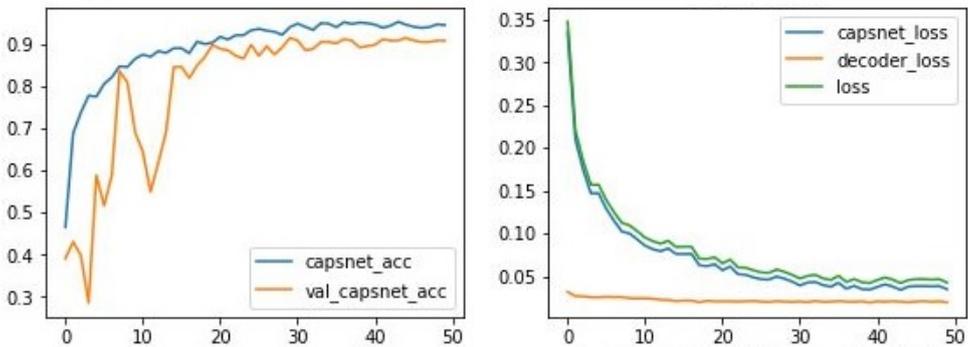


Fig. 17. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Res-CapsNet for image size 64X64.

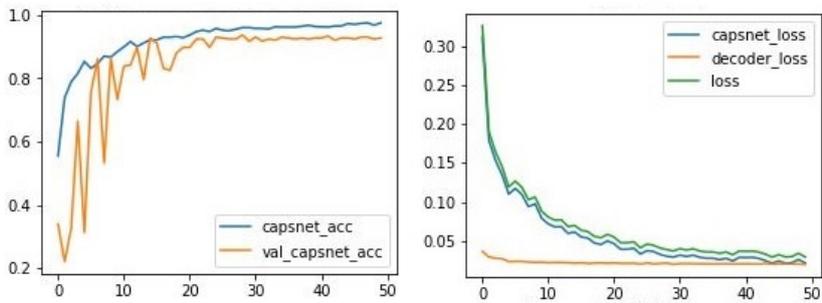


Fig. 18. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Res-CapsNet for image size 64X64.

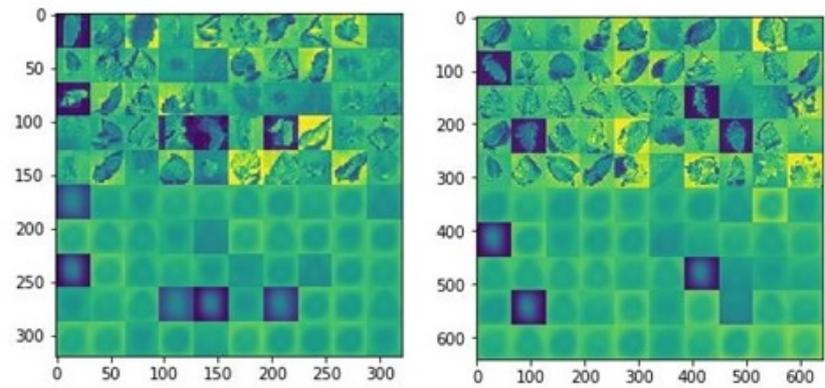


Fig. 19. Sample Real and reconstructed images from CapsNet implementation.

Table 5. Resultant Kappa Coefficient and Losses for the Tomato Late Blight dataset

Tomato Late Blight Images (32X32)						
Networks	Kappa coefficient	Loss	CapsNet loss	Decoder Loss	ValCapsNet loss	ValDecoder Loss
Original CapsNet	0.8021	0.1238	0.1167	0.0182	0.1303	0.0204
SE-Alex-CapsNet	0.877	0.049	0.0414	0.0194	0.0712	0.0209
SE-Res-CapsNet	0.8769	0.0426	0.0349	0.0195	0.0714	0.0218
Tomato Late Blight Images (64X64)						
Original CapsNet	0.8059	0.1065	0.0985	0.0204	0.1026	0.0221
SE-Alex-CapsNet	0.8938	0.0414	0.0343	0.0181	0.0613	0.02
SE-Res-CapsNet	0.9031	0.0295	0.0218	0.0197	0.0516	0.0214

Table 6. No. of parameters for the proposed architectures on our dataset.

Architectures	# of parameters for 32 X 32 image	# of parameters for 64 X 64 image
CapsNet	9,144,064	19,327,744
SE-Alex-CapsNet	8,343,932	17,986,940
SE-Res-CapsNet	6,425,456	16,396,144

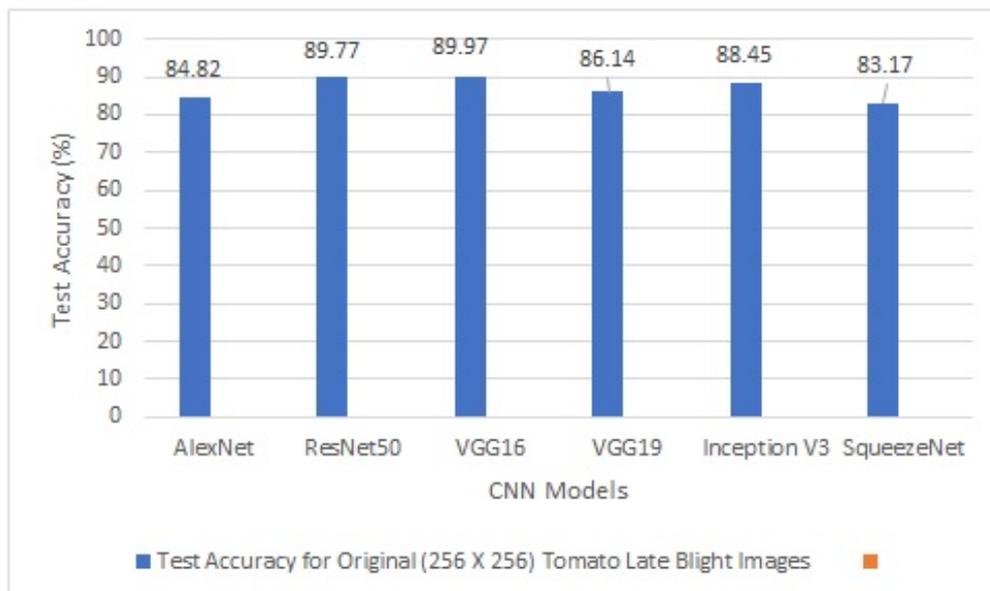


Fig. 20. Test Accuracy comparison of standard CNN models on the original Tomato Late Blight (256X256) images.

5.2 Validation of the proposed models on a different dataset

The next dataset selected for implementation is Grape Black Rot disease, images of which were again taken from the PlantVillage dataset. The images were selected similarly for three diseased stages i.e. early, middle and end stage, along with healthy images. The count of images selected in each category were: healthy - 320, early – 290, middle – 384 and end – 308. Figure 21 depicts the sample images in each category. The images in each class were resized to 32X32 and 64X64, and later enhanced using the block-wise DCT scaling technique.



Fig. 21. Sample images of Grape Black Rot in each category namely, healthy, early, middle and end, taken from the PlantVillage dataset.

As done with tomato dataset, six state-of-the-art CNN architectures namely, AlexNet, SqueezeNet, ResNet50, VGG16, VGG19 and Inception V3 were also implemented on this dataset. Figures 22 – 23 demonstrate the resultant chart comparing the test accuracies obtained with each implementation along with the results of our proposed technique. With SE-Res-CapsNet achieving the highest accuracy of 81.22% and 89.27% on the test set for both image sizes, it can be as-certained that both our approaches again perform better in comparison. Tables 7 and 8 present the resulting values i.e. precision, recall, F1-score and validation loss, for all experiments, performed on images sizes 32X32 and 64X64 respectively. Evidently, our approaches outperform other implementations with better performance measures in terms of precision, recall and F1-score as well, while also achieving the minimum values for validation loss. Similarly, table 9 lists the comparative values of Kappa coefficient as well as loss values related to CapsNet. With improved Kappa coefficient and decreased value of losses, it can be concluded that both our approaches improve upon the existing architecture in this scenario. Figures 24 -29 showcases the resultant accuracy and loss graphs for both image sizes i.e. 32X32 and 64X64

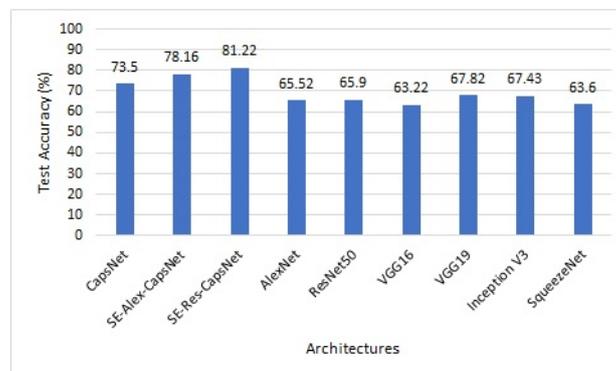


Fig. 22. Accuracy comparison for the implemented models on Grape Black Rot for Images size 32X32.

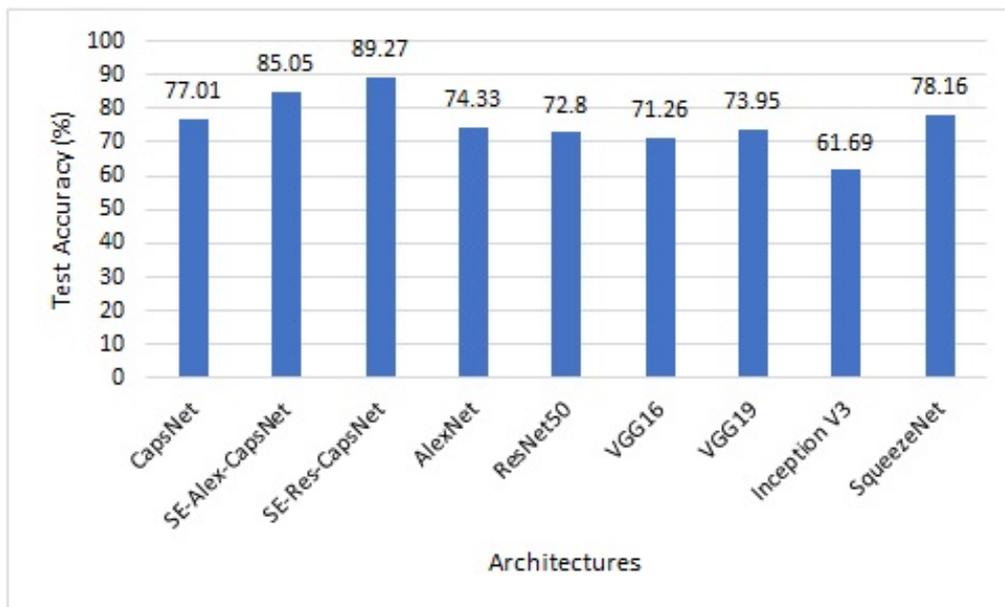


Fig. 23. Accuracy comparison for the implemented models on Grape Black Rot for Images size 64X64.

and the sample reconstructed image in demonstrated in Figure 30. Hence, through the above implementations, it can be concluded that even with a different dataset, SE-Res-CapsNet and SE-Alex-CapsNet outperforms the original CapsNet as well as CNN models with improved accuracy measures. Hence, it can be concluded that even with a different dataset, SE-Res-CapsNet and SE-Alex-CapsNet will outperform the original CapsNet as well as CNN models with improved accuracy measures.

5.3 Performance with the addition of noise

The authors added variation of Salt & Pepper noise, also known as impulse noise, to the original images before implementing with the proposed approaches, in order to put the model under stress and understand its robustness against such noise and disturbances. The noise added was such that it affected 1%, 3%, 5%, 7%, 10% and 20% of pixels in the image, as depicted in figures 31 and 32. Figure 31 showcases a sample tomato late blight original image (leftmost), as well as six variations of images affected by Salt & Pepper noise. Similarly, figure 32 demonstrates the grape black rot image affected by the noise addition. Firstly noise was injected to the original images selected from the PlantVillage dataset for both datasets i.e. tomato late blight and grape black rot, in six different variations for each disease. The modified dataset was then resized to 64X64 and enhanced in the similar way. The proposed approaches were then implemented with these datasets and test accuracy as well as F1-score were recorded for comparative purposes. Figures 33 - 36 showcase the comparative charts for all variations. It can be observed that SE-Res-CapsNet achieves the highest classification accuracy and F1-score in almost all experiments. Also, noise addition doesn't affect the performance of the implemented models that much. Hence, they can be said to be robust against such disturbances in input images.

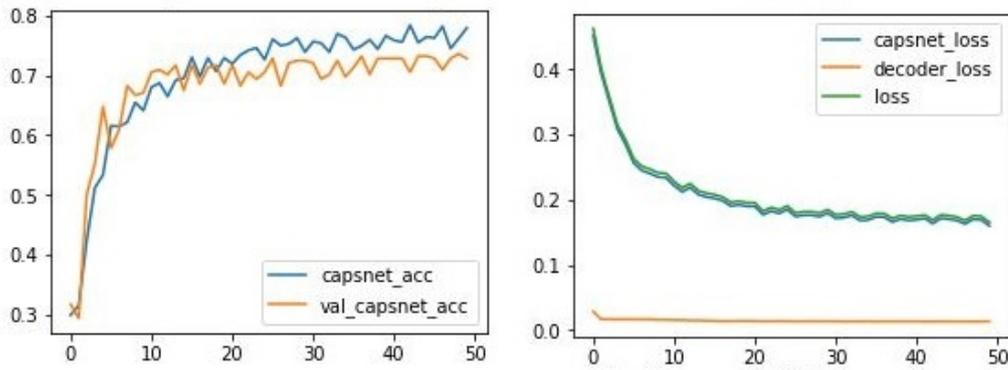


Fig. 24. Graphs depicting the training accuracy, validation accuracy and training loss for Original CapsNet for Grape Black Rot image size 32X32.

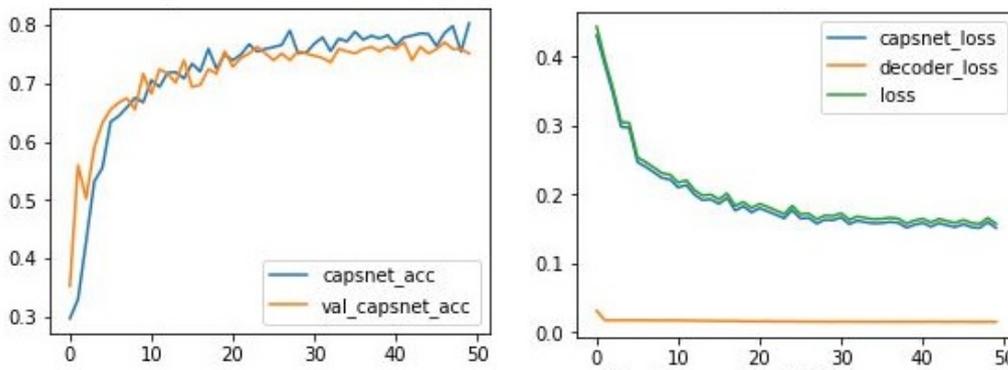


Fig. 25. Graphs depicting the training accuracy, validation accuracy and training loss for Original CapsNet for Grape Black Rot image size 64X64.

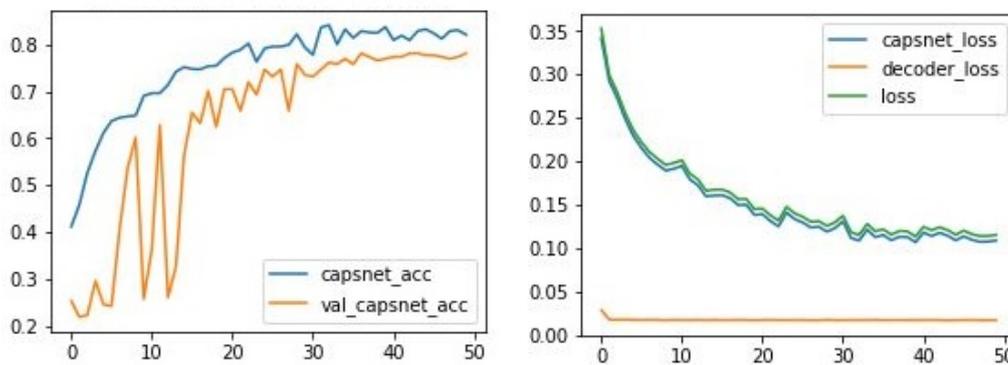


Fig. 26. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Alex-CapsNet for Grape Black Rot image size 32X32.

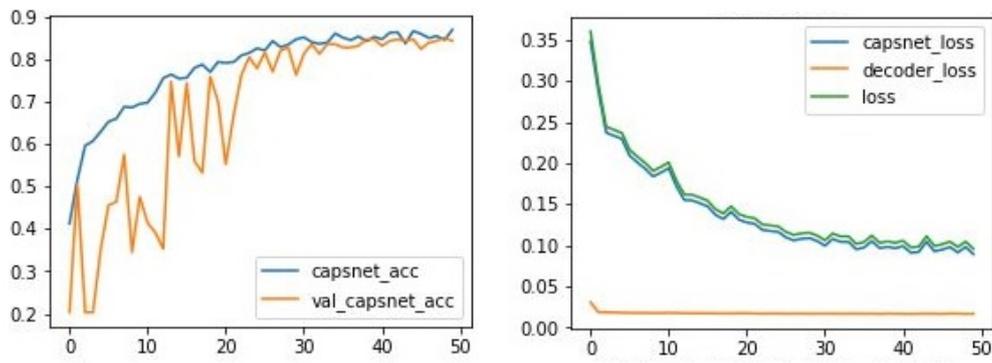


Fig. 27. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Alex-CapsNet for Grape Black Rot image size 64X64.

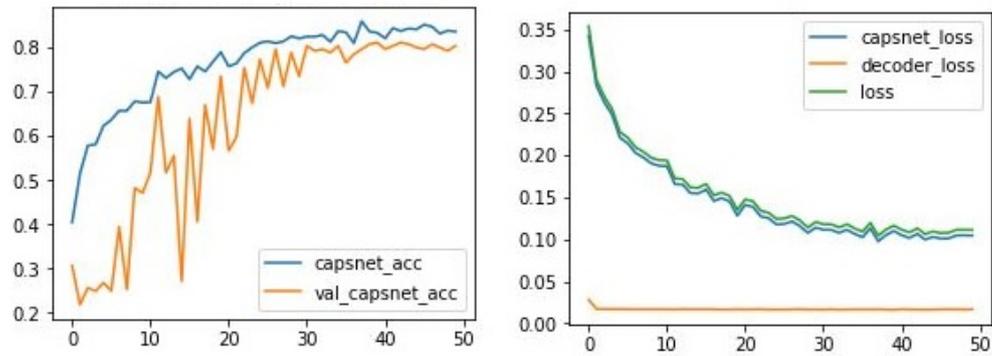


Fig. 28. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Res-CapsNet for Grape Black Rot image size 32X32.

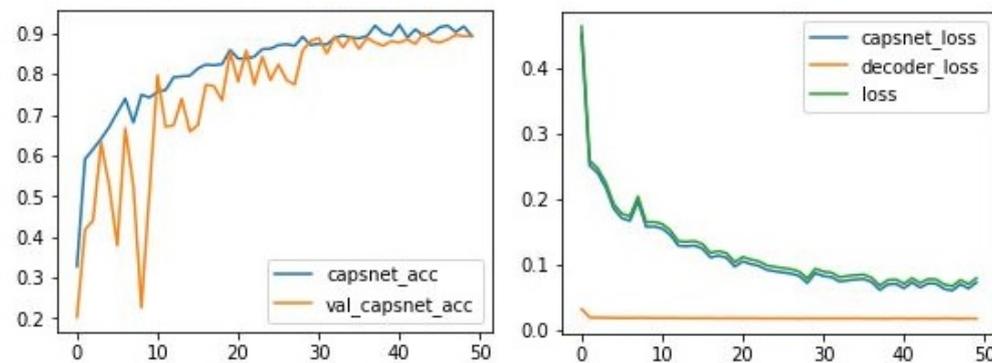


Fig. 29. Graphs depicting the training accuracy, validation accuracy and training loss for SE-Res-CapsNet for Grape Black Rot image size 64X64.

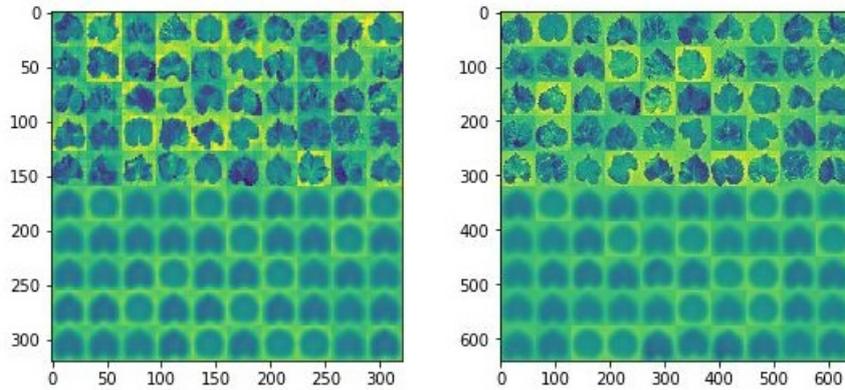


Fig. 30. Sample Real and reconstructed Grape Black Rot images from CapsNet implementation.

Table 7. Resultant Precision, Recall, F1-score and Validation Loss for the Grape Black Rot size 32X32

Networks	Precision	Recall	F1-score	Validation Loss
Original CapsNet	0.7326	0.7496	0.7379	0.1892
SE-Alex-CapsNet	0.7893	0.8033	0.7899	0.1461
SE-Res-CapsNet	0.815	0.8233	0.8191	0.1343
AlexNet	0.7322	0.6372	0.6814	2.5548
ResNet50	0.7011	0.6438	0.6712	1.4586
VGG16	0.7244	0.6134	0.6643	1.7321
VGG19	0.7397	0.6351	0.6834	0.8705
Inception V3	0.7477	0.6557	0.6987	1.3142
SqueezeNet	0.7502	0.6118	0.6739	1.5819

Table 8. Resultant Precision, Recall, F1-score and Validation Loss for the Grape Black Rot Image size 64X64

Networks	Precision	Recall	F1-score	Validation Loss
Original CapsNet	0.7481	0.7536	0.7505	0.1763
SE-Alex-CapsNet	0.8398	0.8607	0.8432	0.1097
SE-Res-CapsNet	0.8899	0.9033	0.8936	0.085
AlexNet	0.7826	0.7322	0.7566	1.9284
ResNet50	0.7916	0.7194	0.7538	1.0465
VGG16	0.7532	0.7115	0.7318	0.8966
VGG19	0.7554	0.7328	0.7439	0.7476
Inception V3	0.7267	0.5988	0.6566	1.1999
SqueezeNet	0.8056	0.7812	0.7932	0.7776

Table 9. Resultant Kappa Coefficient and Losses for the Grape Black Rot dataset

Grape Black Rot Images (32X32)						
Networks	Kappa coefficient	Loss	CapsNet loss	Decoder Loss	ValCapsNet loss	ValDecoder Loss
Original CapsNet	0.6365	0.1648	0.1594	0.0138	0.1837	0.0139
SE-Alex-CapsNet	0.7093	0.1135	0.107	0.0167	0.1401	0.0153
SE-Res-CapsNet	0.7398	0.1103	0.1037	0.017	0.1283	0.0153
Grape Black Rot Images (64X64)						
Original CapsNet	0.665	0.1576	0.1516	0.0154	0.1701	0.0157
SE-Alex-CapsNet	0.7907	0.0963	0.0897	0.0168	0.1037	0.0152
SE-Res-CapsNet	0.8566	0.0793	0.0727	0.017	0.0789	0.0154

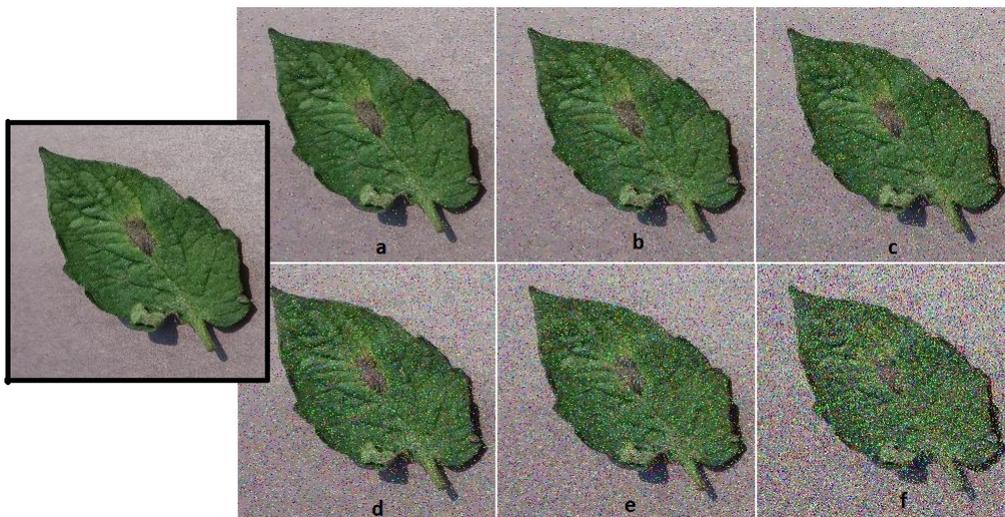


Fig. 31. Original Tomato Late Blight Image (Leftmost), with images impacted by a) 1% b) 3% c) 5% d) 7% e) 10% f) 20% of noise addition.

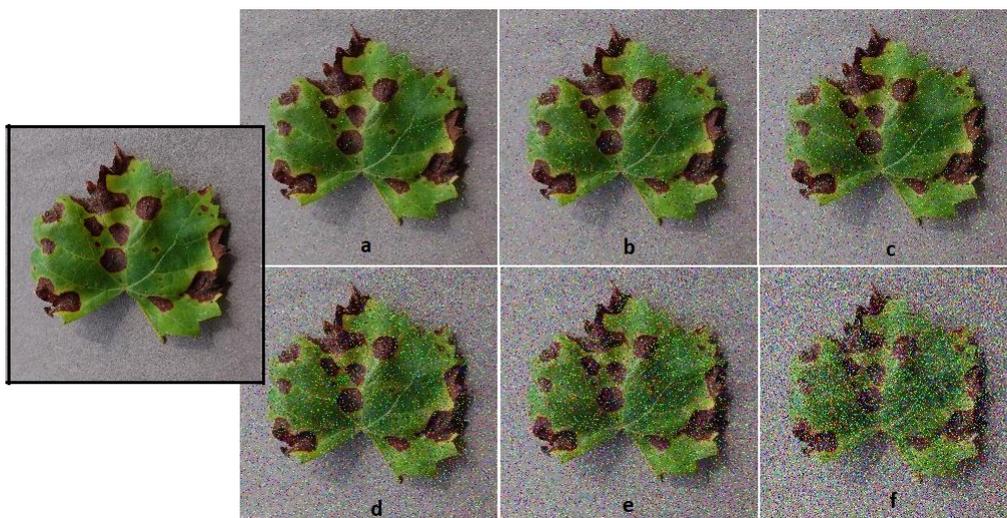


Fig. 32. Original Grape Black Rot Image (Leftmost), with images impacted by a) 1% b) 3% c) 5% d) 7% e) 10% f) 20% of noise addition.

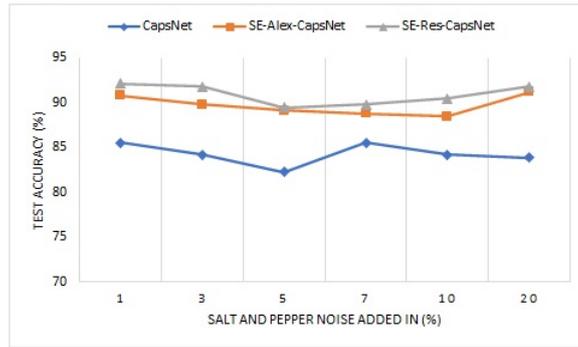


Fig. 33. Test accuracy comparison of Tomato Late Blight image size 64X64 after noise addition.

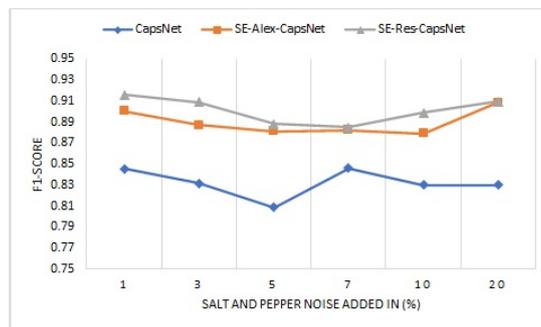


Fig. 34. F1-score comparison of Tomato Late Blight image size 64X64 after noise addition.

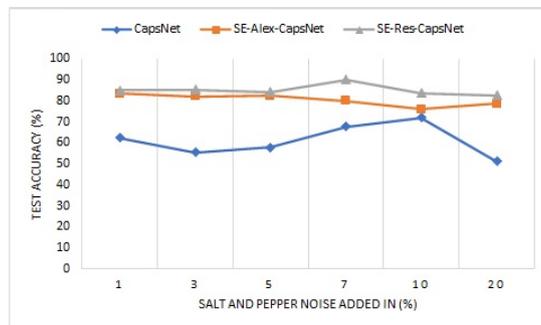


Fig. 35. Test accuracy comparison of Grape Black Rot image size 64X64 after noise addition.

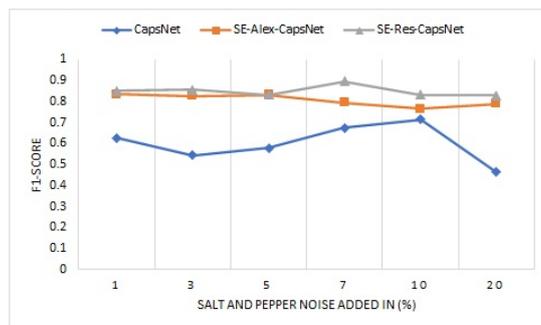


Fig. 36. F1-score comparison of Grape Black Rot image size 64X64 after noise addition.

6. Conclusion and Future Direction

In this paper, two architectures based on squeeze and excitation networks, one constructed with AlexNet and another with ResNet, combined with Capsule networks, were proposed for late blight disease severity estimation for tomato crop. Also, original Capsule Network was implemented for disease severity assessment in plants. The images utilized were taken from the PlantVillage dataset. Upon finalizing the images in each category, downscaling and image enhancement, the resultant images were utilized for feature computation by the SE networks. The performance measures viz. accuracy, precision, recall, F1-score, validation loss etc. of the proposed approaches as well as other pre-trained CNN models namely AlexNet, SqueezeNet, ResNet50, VGG16, VGG19 and Inception V3 were recorded. The results clearly demonstrate that the two proposed architectures exceed the performance of original CapsNet architecture, with SE-Res-CapsNet achieving the highest accuracy of 91.44% with 32X32 image size and 93.75% with 64X64 image size. SE-Alex-CapsNet achieved the accuracy of 90.879% with 32X32 image size and 92.1% with 64X64 image size. The classification accuracy measures of standard CNN architectures was also showcased for comparison. The proposed approaches were also implemented on Grape Black Rot images, achieving similar findings, with SE-Res-CapsNet achieving the highest accuracy of 81.22% with 32X32 image size and 89.27% with 64X64 image size. SE-Alex-CapsNet achieved the accuracy of 78.16% with 32X32 image size and 85.05% with 64X64 image size. Even with noise addition, the proposed models showcased robustness and consistent performance. This is still an active area of research. Our work aims at better performance of the system with less image size and limited computational capabilities, which can be later implemented on a mobile device for actual use by the growers. The proposed methodology improves the performance substantially as compared to standard CNN models, as it benefits from the channel-wise scaling attributed to SE networks and also, doesn't require huge dataset for training and classification due to the utilization and benefits of Capsule networks. Image-based classification methods have an extensive range of design opportunities, still remaining uncharted and unexplored, with abundant choices for restructuring the original architectures. For essentially robust models in future applications, utilization of real-life images is absolutely vital. Images accumulated from several regions, backgrounds, lighting conditions, resources etc. could give a clear understanding of the concrete applicability of the developed prototype. Also, estimating the disease severity correctly could lead to timely and appropriate preventive measures, along with saving the growers from unnecessary plights due to faulty deductions. The authors have proposed two architectures based on AlexNet and ResNet. More such SE networks based on various CNN architectures can be explored. Capsule Networks based on Expectation-Maximization (EM) routing can be analysed for disease severity. The aim is to minimize the computation costs in terms of memory, time and hardware, as well as maximising the performance. With fewer images and their reduced size that require low processing power, the proposed methodology can be exploited to create a mobile application, to be installed on a reasonably priced smartphones for practical usage by farmers and agronomists. The authors aim to collect images via digital and hyperspectral cameras of tomato crop, in laboratory as well as field conditions, for further research and explore the possibilities of early detection of plant diseases.

ACKNOWLEDGEMENTS

This study was supported by Department of Science and Technology (DST), Government of India, New Delhi, under Interdisciplinary Cyber Physical Systems (ICPS) Programme (Project Tilted "Application of Internet of Things (IoT) in Agriculture Sector", Reference No. T-319).

References

- Al-Hmouz, R. (2020).** Deep learning autoencoder approach: Automatic recognition of artistic Arabic calligraphy types. *Kuwait Journal of Science*, 47(3).
- Barbedo, J. G. A. (2019).** Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, 96-107.
- Camargo, A., & Smith, J. S. (2009).** An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems engineering*, 102(1), 9-21.
- Camargo, A., & Smith, J. S. (2009).** Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture*, 66(2), 121-125.
- Campbell, C. L., & Neher, D. A. (1994).** Estimating disease severity and incidence. In *Epidemiology and management of root diseases* (pp. 117-147). Springer, Berlin, Heidelberg.
- Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanekaran, Y. A. (2020).** Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173, 105393.
- Dong, M., Mu, S., Su, T., & Sun, W. (2019, August).** Image Recognition of Peanut Leaf Diseases Based on Capsule nAtworks. In *International CCF Conference on Artificial Intelligence* (pp. 43-52). Springer, Singapore.
- Duong, L. T., Nguyen, P. T., Di Sipio, C., & Di Ruscio, D. (2020).** Automated fruit recognition using EfficientNet and MixNet. *Computers and Electronics in Agriculture*, 171, 105326.
- Dyrmann, M., Karstoft, H., & Midtby, H. S. (2016).** Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151, 72-80.
- Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., & Vasilakoglou, I. (2020).** Towards weeds identification assistance through transfer learning. *Computers and Electronics in Agriculture*, 171, 105306.
- Ferentinos, K. P. (2018).** Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318.
- GITHUB.** <https://github.com/XifengGuo/CapsNet-Keras>. Last Accessed: July 2020.
- Hang, J., Zhang, D., Chen, P., Zhang, J., & Wang, B. (2019).** Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network. *Sensors*, 19(19), 4161.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016).** Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Heckmann, D., Schlüter, U., & Weber, A. P. (2017).** Machine learning techniques for predicting crop photosynthetic capacity from leaf reflectance spectra. *Molecular plant*, 10(6), 878-890.
- Hu, J., Shen, L., & Sun, G. (2018).** Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).
- Hughes, D., & Salathé, M. (2015).** An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018).** Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147, 70-90.
- Karlekar, A., & Seal, A. (2020).** SoyNet: Soybean leaf diseases classification. *Computers and Electronics in Agriculture*, 172, 105342.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012).** Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- Li, Y., Qian, M., Liu, P., Cai, Q., Li, X., Guo, J., ... & Zhou, Z. (2019).** The recognition of rice images by UAV based on capsule network. *Cluster Computing*, 22(4), 9515-9524.
- Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., & Sun, W. (2019).** PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Computers and electronics in agriculture*, 157, 518-529.
- Liu, B., Zhang, Y., He, D., & Li, Y. (2018).** Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1), 11.
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016).** Using deep learning for image-based plant disease detection, *Frontiers in plant science*, vol. 7, p. 1419.
- Mukherjee, J., & Mitra, S. K. (2008).** Enhancement of colour images by scaling the DCT coefficients. *IEEE Transactions on Image processing*, 17(10), 1783-1794.
- Nowicki, M., Kozik, E. U., & Foolad, M. R. (2013).** Late Blight of Tomato. In *Translational Genomics for Crop Breeding* (pp. 241–265), <https://doi.org/10.1002/9781118728475.ch13>.
- Pal, T., Jaiswal, V., & Chauhan, R. S. (2016).** DRPPP: A machine learning based tool for prediction of disease resistance proteins in plants. *Computers in biology and medicine*, 78, 42-48.
- Paoletti, M. E., Haut, J. M., Fernandez-Beltran, R., Plaza, J., Plaza, A., Li, J., & Pla, F. (2018).** Capsule networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4), 2145-2160.
- Prince, G., Clarkson, J. P., & Rajpoot, N. M. (2015).** Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PloS one*, 10(4), e0123262.
- Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., & Plümer, L. (2010).** Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Computers and electronics in agriculture*, 74(1), 91-99.
- Sabour, S., Frosst, N., & Hinton, G. E. (2017).** Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).
- Sabrol, H., & Kumar, S. (2016).** Intensity based feature extraction for tomato plant disease recognition by classification using decision tree. *International Journal of Computer Science and Information Security*, 14(9), 622.
- Singh, V., & Misra, A. K. (2017).** Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture*, 4(1), 41-49.

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

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272-279.

Verma, S., Chug, A., & Singh, A. P. (2020). Exploring capsule networks for disease classification in plants. *Journal of Statistics and Management Systems*, 23(2), 307-315.

Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational intelligence and neuroscience*, 2017.

Zeng, W., & Li, M. (2020). Crop leaf disease recognition based on Self-Attention convolutional neural network. *Computers and Electronics in Agriculture*, 172, 105341.

Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, 6, 30370-30377.

Submitted: 18/09/2020

Revised: 03/01/2021

Accepted: 31/01/2021

DOI: 10.48129/kjs.v49i1.10586