DLCA-CapsNet: Dual-Lane CDH Atrous CapsNet for the Detection of Plant Diseases

Steve Okyere-Gyamfi[1]*, Michael Asante[2], Yaw Marfo Missah[3], Kwame Ofosuhene Peasah[4], Vivian Akoto-Adjepong[5]
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana[1, 2, 3, 4]
Catholic University of Ghana, Sunyani, Ghana[1]
University of Energy and Natural Resources, Sunyani, Ghana[5]

Abstract—**Humanity's survival, development, and existence are deeply intertwined with agriculture, the source of most of our food. Plant disease detection helps in securing food, but manual plant disease detection is error-prone and labor-intensive. Convolutional Neural Networks (CNNs) are highly effective for automated plant disease classification, but their difficulty in recognizing differently oriented images means they need large datasets with many variations to work best. Capsule Networks (CapsNets) were developed to overcome the shortcomings of CNNs and can function effectively with smaller datasets. However, CapsNets process every part of an input image, so their performance can suffer when dealing with complex visuals. To tackle this challenge, DLCA-CapsNet was introduced. DLCA-CapsNet integrates a Color Difference Histogram (CDH) layer for key feature extraction, atrous convolution layers to enlarge receptive fields while maintaining spatial details, along with max-pooling, standard convolutional layers, and a dropout layer. The proposed DLCA-CapsNet method was evaluated on datasets including apple, banana, grape, maize, mango, pepper, potato, rice, tomato, as well as CIFAR-10 and Fashion-MNIST. The model demonstrated strong performance with high test accuracies in plant disease detection and on CIFAR-10 and Fashion-MNIST. It improved test accuracies by 6.78%, 14.82%, 6.14%, 5.07%, 21.12%, 40.32%, 4.64%, 0.76%, 10.23%, 13.73%, and 2.03%, while also reducing the number of parameters in millions by 6.16M, 6.16M, 6.16M, 6.16M, 7.14M, 5.68M, 5.92M, 7.62M, 7.62M, and 6.54M respectively when compared with the original CapsNet. On sensitivity, F1-Score, precision, specificity, Receiver Operating Characteristics, Precision-Recall values, accuracy, disk size, and parameters generated, etc., the DLCA-CapsNet achieved better performance compared to the original CapsNet and other advanced CapsNets reported in the literature. The findings suggest that this efficient and computationally less demanding method can significantly enhance plant disease classification and contribute incrementally to efforts aligned with the SDG 2 goal by offering a lightweight, scalable solution that can be adapted for field use in resource-constrained settings.**

Keywords—Color Difference Histogram (CDH); Convolutional Neural Network (CNN); atrous Convolution; Capsule Neural Network; plant disease detection; dynamic routing; AI in agriculture

I. Introduction

Agriculture has consistently been a crucial social and economic sector for humanity. The production of food is particularly critical, with high demand from every household. As a result, employing innovative technologies to improve the sector is essential for the agri-food industry. Today, artificial intelligence stands out as a significant technological tool extensively utilized in contemporary society. Specifically, Deep Learning (DL) has numerous applications owing to its capability to learn strong representations from images [1] [2] [3].

CNNs are the primary DL architecture for image classification, primarily due to the significant attention they have received in recent years. Multiple CNN architectures have been developed to enhance their performance. However, traditional CNNs still have many limitations. They do not emphasize the arrangement or spatial connections among the components of the image, so they require extensive datasets in various variations to achieve high performance [4], which results in the augmentation of data [5] [6].

To address this issue, a novel architecture named CapsNet, which mimics the human brain, has been introduced for extracting information from images [7]. Despite their numerous advantages, such as performing well with fewer data, CapsNets try to identify every object in an image, which reduces their ability to generalize effectively, especially on unseen and complex images [8][9][10]. To address the issues with CapsNet, researchers sometimes deepen and widen their models, resulting in longer training times and more parameters. Moreover, increasing the CapsNet model depth does not necessarily improve performance. Selecting efficient feature extractors for the encoder network is crucial for enhancing the CapsNet model's performance. Therefore, this study seeks to answer the following research questions:

- Can a well-performing CapsNet be designed to perform well on plant diseases and complex images?
- Can good feature extractors be incorporated into the CapsNet to improve the encoder network for better feature extraction of the CapsNet Model?
- How can we explain the "black box" phenomenon of this AI model to enhance comprehension of the model and confidence in the model's output?
- Can the CapsNet model be optimized to make it deployable on resource-constrained devices?

This study suggests a low-parameterized shallow CapsNet with efficient feature extraction abilities to help classify plant diseases efficiently using CDH [11] and atrous convolutions [12] [13]. Results from experiments conducted on 11 publicly accessible datasets demonstrate that the proposed DLCA-CapsNet model performs on par with the most advanced models available for detecting complex images and other plant diseases.

The visualization of class capsule clusters, activation maps, and reconstructed images demonstrates the superior efficiency of the feature extractors of DLCA-CapsNet compared to the feature extraction capabilities found in the traditional CapsNet model.

This study's contributions are:

- A novel and well-performing CapsNet design named DLCA-CapsNet is introduced.
- CDH is utilized to extract significant features in CapsNet models. Additionally, the DLCA-CapsNet employs atrous convolution to enhance spatial representation.
- We conducted thorough visualizations of plant diseases and other datasets to enhance the explainable artificial intelligence (XAI) field.
- The DLCA-CapsNet model generated fewer parameters and had a smaller size, enhancing its suitability for deployment on devices with limited resources.

The remainder of the paper is structured as follows: Section II reviews related literature, Section III outlines the methodology employed, Section IV discusses the experimental results, and Section V provides the conclusion and suggestions for future research.

II. Related Works

A. Introduction

In recent years, plant disease detection has emerged as a crucial area of research in agriculture, meeting the demand for timely and precise diagnosis to ensure crop health and productivity. CNNs have been widely used in this domain, demonstrating notable success in image classification tasks compared to CapsNet. This section reviews the advancements and applications of Capsule networks for plant disease detection, highlighting their effectiveness in improving detection accuracy and generalisation.

B. Review of CapsNet for Plant Disease Detection in Literature

Mensah et al. proposed combining CapsNet and Gabor to detect deformed and unclear diseases in citrus and tomatoes, utilising the PlantVillage dataset. Their system attained 93.33% validation accuracy for classifying citrus diseases and 98.13% with 12 million parameters for classifying tomato diseases [14]. Verma and their team created an optimized technique for computing features by leveraging Squeeze and Excitation (SE) Networks. These networks are applied before the original CapsNet in the classification process to assess the severity of plant diseases. They incorporated AlexNet and ResNet into CapsNet to get two SE CapsNets. They used a tomato dataset from PlantVillage and classified Late Blight diseases into late, middle, early, and healthy stages. The SE-ALexNet CapsNet had 9 million parameters with an accuracy of 92.1%, and SE-Res CapsNet had 19 million parameters with an accuracy of 93.75% [15]. Vasudevan and Karthick proposed a composite technique for detecting grape diseases. They obtain a leaf area by utilizing a method based on a graph. To augment the dataset, they employ a Generative Adversarial Network. Although the model was computationally complex, it attained a validation accuracy of 97.63% on diseases in grapes using captured data and data from the PlantVillage dataset [16]. Xu et al. proposed a model, combining CapsNet with inception modules to amplify receptive fields using diverse dilation rates. This helps to improve the multi-level characteristics of diseases in apple leaves. The model attained 93.16% validation accuracy [17]. Mensah et al. introduced a model by replacing sigmoid with SoftMax, CNN with Local Binary Pattern, and dynamic routing with k-means routing. Experimentally on CIFAR-10, fashion-MNIST, MNIST, citrus, maize, and tomato, 75.80%, 92.72%, 99.68%, 99.41%, 96.79%, and 98.06% validation accuracies were attained, generating 5.2 million for CIFAR-10, 2.8 million parameters for fashion MNIST and MNIST, and 8.4 million for tomato and citrus [18]. Verma and co-authors used the conventional CapsNet to classify potato leaf images from the PlantVillage dataset. The experimental results reveal 9,856,768 parameters produced by the model and 91.83% validation accuracy attained [19]. Oladejo and Ademola classified banana diseases by proposing a CapsNet model, in which they changed the neuron number in the fully connected layer of the original CapsNet to two 512 and 1024 instead of 3, hence reducing the time for training the network. Also, they used an optimizer for momentum to upgrade the network processing speed. On banana leaf disease detection, the model attained a validation accuracy of 95% [20]. Atlan changed the traditional CapsNet model by adjusting the three fully connected layers to 960, 768, and 4096. On assessing the model, it gained a sensitivity of 96.37%, an accuracy of 95.76%, and a specificity of 97.49% when used to classify bell pepper diseases [21]. Mensah and colleagues chose to use the Gabor filter rather than CNNs. This change allowed the initial layers to capture spatial and texture relationships, improving overall performance. Max Pooling was used to select the most important features after the Gabor filter convolution, reducing the feature vectors' dimensionality. Tests conducted on tomato datasets from plant villages showed a validation accuracy of 97.98%, with a total parameter count of 8,708,128 [22]. Anant suggested a technique called AppleCaps that overcomes the spatial invariance issue discovered in CNNs. The model gained 87.06% accuracy on an augmented dataset containing diseased apple leaves [23]. Mensah et al. proposed a CapsNet model that uses dual input. The output from these inputs is merged and submitted to the layers of the original CapsNet. Upon assessing the system on the tomato and CIFAR-10 datasets, it generated 5.48 million (M) and 6.04 M parameters and 76.58% and 93.03% validation accuracies, respectively [24]. Peker suggested a CapsNet that uses multiple ensemble channels by implementing different CapsNet models on images with various pre-processing methods. By merging the networks to learn diverse features from data, the model increases in performance and attains 98.15% on 10 disease classes of the tomato dataset [25]. Abouelmagd and colleagues developed a computer vision approach using an improved CapsNet to recognize and categorize ten distinct diseases affecting tomato leaves from standard datasets. To minimize overfitting, they employed pre-processing and data augmentation approaches when training. Their CapsNet method achieved an accuracy of 96.39% [26]. Zang and colleagues proposed a method for detecting plant leaf diseases by combining residual networks (ResNet) and CapsNet. They improved the traditional ResNet by replacing the kernel with a set of 3 × 3 convolutional kernels and incorporating an attention mechanism to guide the model in prioritizing important features. The upgraded ResNet was then integrated with CapsNet. This combined model, SE-SK-CapResNet, achieved accuracy rates of 98.58%, 97.19%, and 95.08% on the PlantVillage, Tomato Leaf Disease, and AI Challenger 2018 datasets, respectively [27]. Mensah and co-authors proposed Shallow and Multi-input CapsNets. The shallow CapsNet used LBP, the squash function, and a normalizer in the standard CapsNet model. The model was assessed on datasets, CIFAR-10, fashion-MNIST, and tomato, and achieved validation accuracies of 75.75%, 92.70%, and 97.33% with 4.6M, 2.5 M, and 4.1M parameters, respectively. Furthermore, the multi-input CapsNet merged the results of three (3) convolution layers before feeding them to the standard CapsNet model. The model was also assessed on the same dataset as shallow CapsNet and attained validation accuracies of 63.95 %, 91.45%, and 94.04% with 4.3 M, 2.2 M, and 4.0 M parameters, respectively [28]. Andrushia and co-authors proposed a technique for classifying grape leaf diseases by inserting convolution layers before the primary capsules. This resulted in speeding up the dynamic process by reducing the number of capsules. On both the non-augmented and augmented grape leaf disease datasets from PlantVillage, the model achieved 99.12% accuracy [29].

C. Summary

These models have performed well on different plant disease datasets. Still, these current CapsNet models for plant disease identification are often limited by scalability, processing speed, many parameters, large size, and robustness to complex backgrounds. Developing a more advanced CapsNet model could enhance accuracy, improve generalization, reduce parameter count and size on disk, and ultimately aid in more effective crop management.

III. Methodology

A. Capsule Network

Unlike CNNs, CapsNets [7] can identify orientation, texture, and pose. In CapsNets, neurons are organised into capsules, each of which has an activity vector that encodes different instantiation parameters for detecting a specific object type. These capsules provide a probability of the object's presence and its generalised pose. Capsules receive these activity vectors from the capsules in the preceding layer, and the connections between these layers, called coupling coefficients, have varying values. If the present capsule identifies a dense cluster of earlier predictions, strongly suggesting the object's presence, it generates a high probability, a process referred to as routing by agreement. Therefore, if the lower layer capsule's prediction aligns with the current capsule's real output, the coefficient among them rises, calculated using the softmax function as illustrated in Eq. (1).

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

Where $\hat{u}_{j|i}$ represents the j^{th} capsule output or prediction vector, W_{ij} and u_i denote the weight matrix and output vector of the capsule i in the lower-level layer, respectively. Coupling coefficients are determined through the softmax function in Eq. (2), reflecting the level of alignment between capsules in adjacent layers.

$$C_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (2)$$

Here, b_{ij} represents the probability amongst the two (2) capsules based on logarithmic values, set to zero. The j^{th} capsules input vector s_j is determined as in Eq. (3),

$$s_j = \sum_{i=1}^{N} c_{ij} \,.\, \hat{u}_{j|i} \qquad (3)$$

Ultimately, the subsequent squash function in Eq. (4) is utilised to confine the output within the range of 0 to 1.

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

Eq. (5) calculates the loss function for the capsules in the final layer. Here, T_k equals 1 if class k is active, and 0 if otherwise. The values of λ, m^-, and m^+ are determined during the learning process.

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

B. UnitsColour Difference Histogram (CDH) Feature Map Detection

The CDH method prioritizes color, how edges line up, and color changes that look natural to us. It captures these features in a way that mimics how our eyes and brain understand them. This method introduces a novel visual descriptor that combines color, edge direction, and how we visually perceive color differences, all while taking into account how these elements are arranged in space. With edge orientation identified and colors quantized via CDH, the next stage is to find small structural elements. This is done by looking at each pixel and then using a 3x3 filter to apply the CDH framework for color and edge quantization [11]. Examining a filter's central value in relation to its eight surrounding values yields a set of edge and color maps. From these maps, we can derive features representing color and edge characteristics by computing the difference (Δ) in color intensity and edge direction for each L*a*b* component. In the final step, we integrate these two properties into one histogram. In a quantized image, the brightness or color levels of each pixel, denoted as $C(x, y)$, are limited to a specific set of discrete values. These values span from a minimum of 0 up to a maximum of $W - 1$. Neighbouring pixels, identified by their coordinates (x, y) and $(x'y')$ have associated color index values as $C(x, y) = w_1$ and $C(x'y') = w_2$. The orientation image for edge by $\theta(x, y)$ stores orientation information as discrete values v (ranging from 0 to $V - 1$) for each pixel (x, y). At specific coordinates (x, y) and $(x'y')$, the orientation angles are as $\theta(x, y) = v_1$ and $\theta(x'y')$=v_2.

Color difference histograms for neighboring pixels separated by distance D, with color quantization W and edge orientation quantization V, are defined using Eq. (6) and Eq. (7).

$$H_{color}(\, C(x, y)\,) = \begin{cases} \sum \sqrt{(\Delta L^2) + (\Delta a^2) + (\Delta b^2)} \\ where\ \theta(x, y) = \theta(x'y'); max(|x - x'|), (|y - y'| = D) \end{cases} \quad (6)$$

$$H_{ori}(\, \theta(x, y)\,) = \begin{cases} \sum \sqrt{(\Delta L^2) + (\Delta a^2) + (\Delta b^2)} \\ where\ C(x, y) = C(x'y'); max(|x - x'|), (|y - y'| = D) \end{cases} \quad (7)$$

Here, ΔL, Δa, and Δb denote the differences between two color pixels, with D set to a value of 1. When the edge orientation is W and the color quantization level is V, the CDH feature is computed as shown in Eq. (8):

$$H_{CDH} = H_{color}(0), H_{color}(1) \dots H_{color}(W - 1), H_{ori}(0), H_{ori}(1) \dots H_{ori}(V - 1) \quad (8)$$

Image retrieval features can be created by combining a color histogram H_{color} with a histogram of edge orientations H_{ori}. If, for instance, we divide the color space into 72 bins and the edge orientations into 18 bins, the combined feature vector, represented as H, will have a total of 90 dimensions (72 + 18).

C. Atrous Convolutions

Atrous convolution or dilated convolution is crafted to see a wider input area without needing more computing power and parameter count. They are primarily used in tasks like semantic segmentation [12] [13]. Regular deep CNN combines convolutional layers with max-pooling. The downside is that the feature maps shrink in size by 50% every time a max-pooling operation happens. As a result, projecting the processed feature information back onto the initial image leads to less detailed feature extraction as the neural network goes deeper. Atrous (Dilated) convolution tackles this problem by allowing for more thorough feature extraction. It introduces a new parameter known as the rate (r). Atrous convolution functions much like standard convolution, with the key difference being that its kernel weights are distributed with a spacing of *r* positions, creating sparsely connected convolution layers. Atrous convolution expands its field of view by changing how spread out its sampling points are. Different levels of this spread can capture information at various scales without more computational resources. For example, a standard 3x3 convolution can be modified to see the same area as a 5x5 or 7x7 convolution, allowing it to pick up image features of different sizes. By applying dilated convolutions with varying dilation rates, receptive fields at multiple scales can be effectively captured, and dilated convolution functions can serve as a mechanism for multi-scale convolutional processing. Eq. (9) and Eq. (10) present the formulations for computing the dilated convolution kernel and its corresponding receptive field, respectively.

$$n = k + (k - 1)(r - 1) \quad (9)$$

$$l_m = l_{m-1} + [(f_m - 1)\prod_{i=1}^{m-1} S_i] \quad (10)$$

In this context n and k denote the sizes of the dilated and standard convolution kernels, respectively. The term l_{m-1} refers to the receptive field size of the l_{m-1}th layer, while the receptive field at the m^{th} layer is determined after applying convolution. f_m represents the convolution kernel size at the m^{th} layer, and S_i indicates the stride of layer l. Fig. 1 illustrates the concept of atrous (dilated) convolutions utilizing a 3×3 kernel with dilation rates of 1, 2, and 4.

D. Proposed Architecture

The proposed DLCA-CapsNet model found in Fig. 2 comprises a CDH layer, two astrous convolution layers, two traditional convolution layers, max-pooling layers, batch normalizers, and a dropout layer. The input image, rescaled to 32x32x3, is subjected to the CDH layer, and does not add any parameters. The 32x32x3 feature map from the CDH layer is sent through two different lanes (Lane1(L1) and Lane2(L2)), each starting with atrous convolutions (Atrous_Conv1 and Atrous_Conv2) with a dilation rate of 2, which processes the feature map subjected to them to produce a 32x32x32 feature map. Each feature map from these astrous convolutions from L1 and L2 is sent to a max-pooling layer (MP1 and MP3), producing feature maps of dimensions 16x16x32. The outputs from the max-pooling layers from L1 and L2 are forwarded to the convolution layers (Conv1 and Conv2) of 64 filters, a 3×3-sized kernel operating with a stride length of 1, respectively. They produce 14x14x64 feature maps sent to max pooling layers (MP2 and MP4) in these separate lanes to produce 7x7x64 feature maps each for L1 and L2. Feature maps from L1 and L2 are then merged or concatenated to produce a 7x7x128 feature map. This merging allows the model to gain the ability to learn integrated features. The merged output is then sent to a dropout layer, then to a Primary Capsule (PC) layer with eight dimensions and 16 channels, a 3×3-sized kernel operating with a stride length 2. Output from the PC layer is sent to the class capsule (Plant_DiseaseCaps), which produces its output for reconstruction. This class capsule layer consists of 16D capsules and considers every class within a dataset. The decoder layer, comprised of three fully connected layers encompassing 512, 1024, and 3072 neurons, respectively, gets the feature map and decodes the entity's properties. The batch normalization layers incorporated in the model helped in the consistent data distribution and improved and simplified the training. The atrous, CDH, dropout, and max-pooling layers help retrieve very salient features submitted to the model from the image for better categorization.

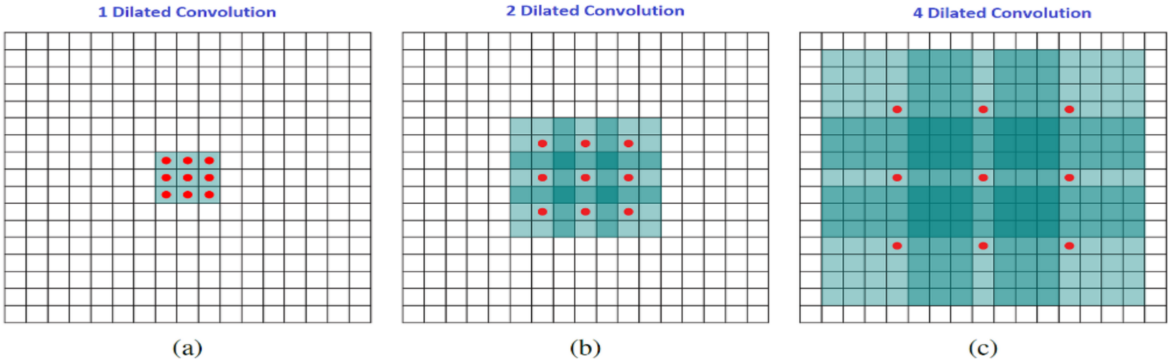


Fig. 1. Demonstrating the idea of astrous convolution. Dilation significantly increases the receptive fields while maintaining full resolution. (a) 1-rate dilated convolution (b) 2-rate dilated convolution and (c) 4-rate dilated convolution.

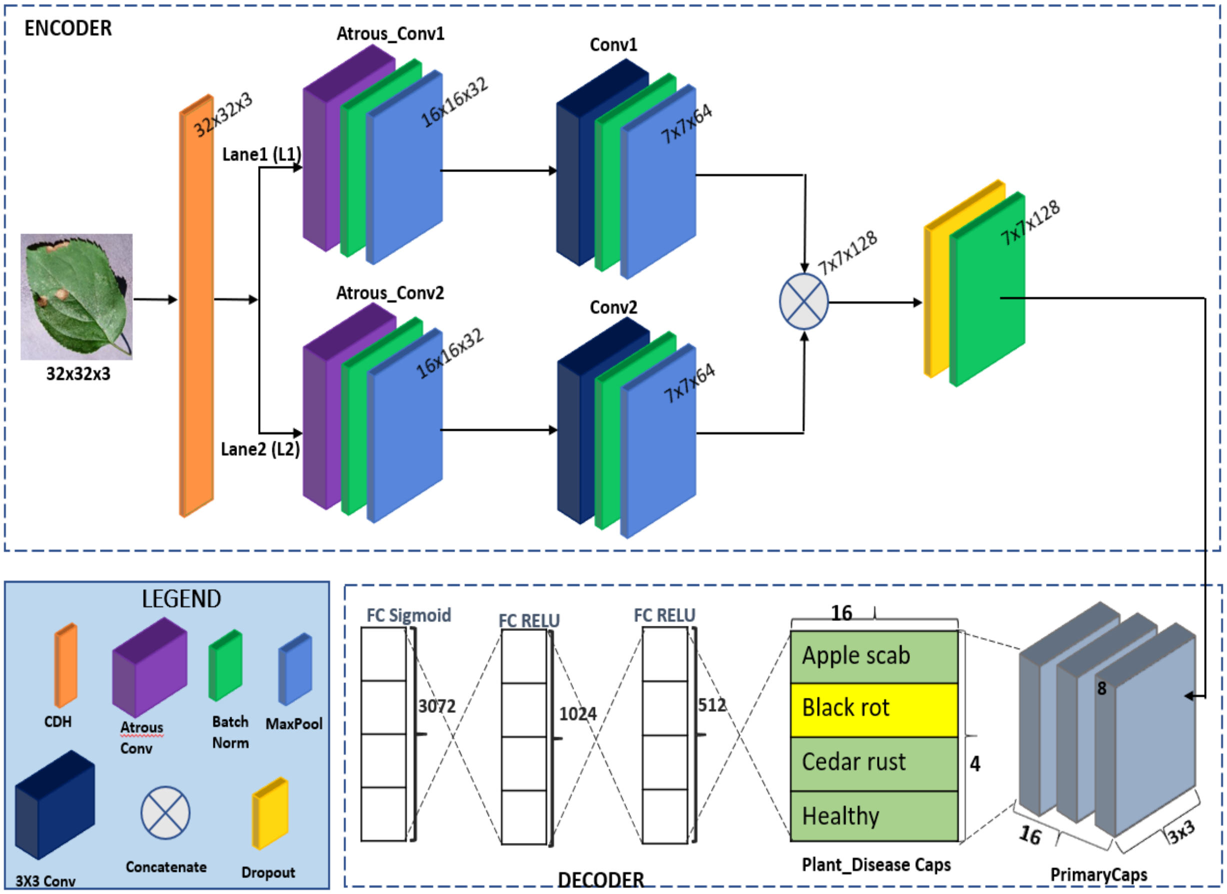


Fig. 2. Architecture of the suggested DLCA-CapsNet model.

E. Datasets

All datasets used are publicly available datasets.

The apple, grape, corn, pepper, potato, and tomato datasets, scaled at 256x256, are a portion of the Plant Village [30].

Apple: It comprises 3171 images sorted into four groups: three infected classes and one healthy class. 0,1,2, and 3 represents Apple_scab, Black_rot, Cedar_apple_rust, and healthy respectively.

Grape: It contains 4062 images in four groups; 0,1,2, and 3 represents Black_rot, Esca_(Black_Measles), Leaf_blight_(Isariopsis_Leaf_Spot, and healthy respectively.

Corn: It is made up of 3852 images sorted into four groups; three infected classes and one healthy class. 0,1,2, and 3 represents Cercospora_leaf_spot Gray_leaf_spot, Common_rust, Northern_Leaf_Blight, and healthy respectively.

Pepper: It is made up of 2475 images sorted into two groups; one infected class and one healthy class. 0 and 1 represent healthy and Bacterial_spot, respectively.

Potato: comprises 2152 images sorted into three groups: two infected classes and one healthy class. 0,1, and 2 represent Healthy, Early_blight, and Late_blight, respectively.

Tomato: It comprises 18160 images sorted into ten groups; nine infected and one healthy class. 0,1,2,3,4,5,6,7,8 and 9 represents Bacterial_spot, Early_blight, Late_blight, Leaf_Mold, Septoria_leaf_spot, Spider_mites Two-spotted_spider_mite, Target_Spot, Tomato_mosaic_virus, Tomato_Yellow_Leaf_Curl_Virus, and Healthy respectively.

Banana comprises 937 images sorted into four groups: three infected classes and one healthy class. 0, 1, 2, and 3 represent Cordana, Pestalotiopsis, Sigatoka, and Healthy [31].

Mango comprises 4000 images sorted into eight groups: seven infected classes and one healthy class. 0,1,2, 3, 4, 5, 6, and 7 represent Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Sooty Mould, and Healthy, respectively [32].

Rice: It comprises 5932 images sorted into four infected classes. 0, 1, 2, and 3 represent Bacterial_blight, Blast, Brown_spot, and Tungro, respectively [36].

Fashion-MNIST: contains 70,000 grayscale fashion product images, each with dimensions of 28x28 pixels. It is divided into ten categories, each containing 7,000 images. The training set includes 60,000 images, while the remaining 10,000 images form the test set. This dataset is more complex than MNIST [33].

CIFAR-10: The dataset contains 50,000 images for training and 10,000 images for testing. Each image has a resolution of 32x32x3 and includes diverse backgrounds and objects, making it more complex than the Fashion-MNIST dataset [34].

To standardize input across datasets, images were resized to 32 × 32 × 3 and split into training and testing sets in an 80:20 ratio. Data augmentation was not applied, as CapsNet is suitable for limited data, and the DLCA-CapsNet is optimized for extracting critical features. Given that most of the ten datasets are imbalanced, high performance would underscore the model's effectiveness in dealing with real-world data imbalance.

F. Implementation Details

The setup involved utilizing a Windows machine with an RTX 2080 SUPER GPU featuring 8GB of dedicated memory and 32GB of RAM. The implementation utilized Keras and Python through Anaconda, with TensorFlow as the backend. For training. The optimization process used Adam, operating with a learning rate of 0.001 and a decay rate of 0.9, with training conducted in batches of 100 samples. To ensure optimal progress during training, the model with the best performance was saved at each iteration. The authors referenced the architecture of the original CapsNet, which is available at the following GitHub repository: https://github.com/XifengGuo/CapsNet-Keras.

IV. Results and Discussion

In this part of the article, we assessed the proposed DLCA-CapsNet model by conducting a comparative analysis with the traditional CapsNet and other cutting-edge CapsNet techniques for classifying plant disease. This comparison aims to determine the most effective model for classifying diseases from plant images.

A. Performance Evaluation

The outcomes of the DLCA-CapsNet architecture, which was trained on images from the plants, CIFAR-10, and fashion-MNIST datasets, are presented here. Confusion matrices are also p, loss and validation curves that depict accuracy (ACC), Area Under the Curve (AUC) for both Precision-Recall (PR) and Receiver Operating Characteristics (ROC) curves, sensitivity (SEN), precision (PRE), specificity (SPE), and F1-Score (FS) are presented to assess the model's performance. Eq. (11)-(15) represent the various performance metrics calculated by considering the confusion matrix's TP, FP, TN, FN, representing True Positive, False Positive, True Negative, False Negative.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{11}$$

$$Precision(P) = \frac{TP}{TP+FP} \tag{12}$$

$$Recall(R)/Sensitivity = \frac{TP}{TP+FN} \tag{13}$$

$$Specificity = \frac{TN}{TN+FP} \tag{14}$$

$$F1 - Score = 2\left(\frac{P*R}{P+R}\right) \tag{15}$$

Additionally, discussions cover visual representations of reconstructed images, clusters of class capsules, and layer activation maps, which illustrate the model's internal mechanisms and help users have confidence in the model. Again, an ablation study demonstrates the model's flexibility and robustness, highlighting the components that significantly impact its performance. Moreover, the quantity of parameters and the model's disk size are also discussed.

Based on the confusion matrices in Fig. 3 for the tomato and mango datasets, it is evident that the DLCA-CapsNet model categorised the images into their respective classes more accurately than the traditional model. From the confusion matrices, accuracy per class, sensitivity, precision, F1-score, and specificity values can be calculated. Such metrics ensure a well-rounded analysis of model performance, which is crucial for small and unevenly distributed dataset scenarios, where relying solely on accuracy might overlook critical details. Based on Table I and Table II, it can be noticed that the proposed DLCA-CapsNet model achieved higher accuracy, sensitivity, precision, F1-Score, and specificity, surpassing values attained by the traditional model when the tomato and mango datasets are considered respectively. This suggests that the proposed model outperforms the traditional model and effectively generalizes unseen data.

Also, Fig. 4 depicts the PR and ROC curves of the tomato dataset, providing insights into the model's resilience and effectiveness on imbalanced datasets. The suggested model outperformed the traditional model, as indicated by superior curve shapes and values. The DLCA-CapsNet significantly outperformed the original CapsNet on the Tomato dataset, achieving an overall ROC of 100% and a PR of 99.5%, compared to the original CapsNet's lower ROC of 97.7% and PR of 88%.

The training and validation loss and accuracy for both the DLCA-CapsNet and traditional CapsNet are shown in Fig. 5 for tomato, mango, and CIFAR-10 datasets. The DLCA-CapsNet model converged faster and achieved higher validation accuracies than the traditional model on the various datasets. It can be seen that, the DLCA-CapsNet significantly outperformed the original CapsNet in validation accuracy across eleven diverse datasets (apple, banana, grape, corn, mango, pepper, potato, rice, tomato, CIFAR-10, and Fashion-MNIST), achieving consistently higher validation accuracy scores of 99.69%, 95.77%, 99.63%, 97.66%, 99.50%, 100%, 100%, 100%, 98.82%, 77.31%, and 93.01% compared to that of the traditional CapsNet model that achieved lower validation accuracies of 92.91%, 80.95%, 93.49%, 92.59%, 78.38%, 59.68%, 95.36%, 99.24%, 88.59%, 63.58%, and 90.98% for the same dataset. This suggests the proposed model generalizes better on unseen data than the traditional model across all the datasets. Examining the original or traditional CapsNets validation accuracy for the CIFAR-10 dataset reveals its initial increase but decline around the 20th epoch. In contrast, the proposed model maintained its peak accuracy until the end.

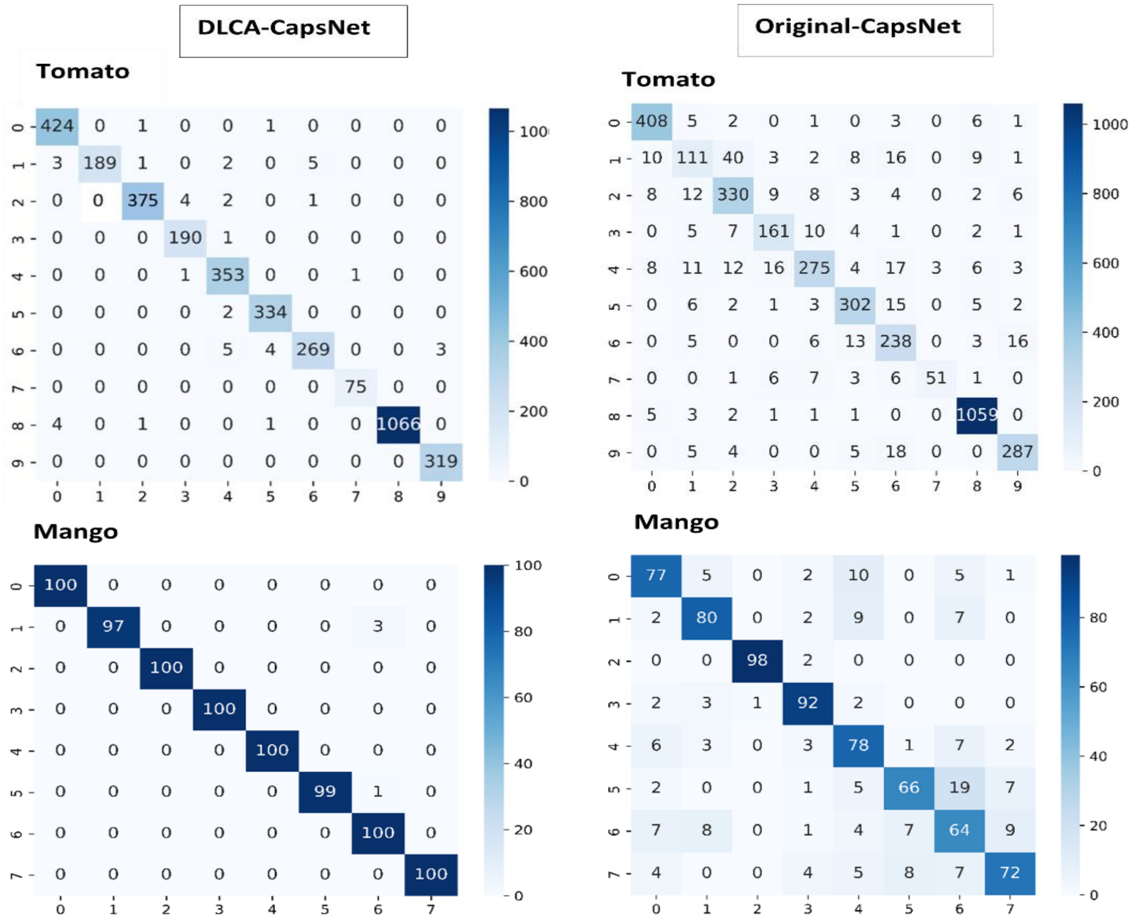


Fig. 3. Confusion matrices comparing the performance of the DLCA-CapsNet and the original CapsNet models on the tomato and mango datasets.

TABLE I. PERFORMANCE RESULTS OF DLCA-CAPSNET AND THE ORIGINAL CAPSNET ON THE TOMATO DATASET

Model (Dataset)	Class	TP	FP	FN	TN	ACC (%)	PRE (%)	SEN (%)	SPE (%)	FS (%)	Data Size
Original-CapsNet (Tomato)	0	408	31	18	3180	98.65	92.94	95.78	99.04	94.34	426
	1	111	52	89	3385	96.12	68.10	55.50	98.49	61.16	200
	2	330	70	52	3185	96.65	82.50	86.39	97.85	84.40	382
	3	161	36	30	3410	98.19	81.73	84.29	98.96	82.99	191
	4	275	38	80	3244	96.76	87.12	77.47	98.84	82.01	355
	5	302	41	34	3260	97.94	88.05	89.88	98.76	88.96	336
	6	238	80	43	3276	96.62	74.84	84.70	97.62	79.47	281
	7	51	3	24	3559	99.26	94.44	68.00	99.92	79.07	75
	8	1059	34	13	2531	98.71	96.89	98.79	98.68	97.83	1072
	9	287	30	32	3288	98.30	90.54	89.97	99.10	90.25	319
DLCA-CapsNet (Tomato)	0	424	7	2	3204	99.75	98.38	99.53	99.78	98.95	426
	1	189	0	11	3437	99.70	100	94.50	100	97.17	200
	2	375	3	7	3252	99.73	99.21	98.17	99.91	98.69	382
	3	190	5	1	3441	99.84	97.44	99.48	99.86	98.45	191
	4	353	12	2	3270	99.62	96.71	99.44	99.63	98.06	355
	5	334	6	2	3295	99.78	98.24	99.41	99.82	98.82	336
	6	269	6	12	3350	99.51	97.82	95.73	99.82	96.76	281
	7	75	1	0	3561	99.97	98.68	100	99.97	99.34	75
	8	1066	0	6	2565	99.84	100	99.44	100	99.72	1072
	9	319	3	0	3315	99.92	99.07	100	99.91	99.53	319

TABLE II. PERFORMANCE RESULTS OF DLCA-CAPSNET AND THE ORIGINAL CAPSNET ON THE MANGO DATASET

Model (Dataset)	Class	TP	FP	FN	TN	ACC (%)	PRE (%)	SEN (%)	SPE (%)	FS (%)	Data Size
Original-CapsNet (Mango)	0	77	23	23	677	94.25	77.00	77.00	96.71	77.00	100
	1	80	19	20	681	95.13	80.81	80.00	97.29	80.40	100
	2	98	1	2	699	99.63	98.99	98.00	99.86	98.49	100
	3	92	15	8	685	97.13	85.98	92.00	97.86	88.89	100
	4	78	35	22	665	92.88	69.03	78.00	95.00	73.24	100
	5	66	16	34	684	93.75	80.49	66.00	97.71	72.53	100
	6	64	45	36	655	89.88	58.72	64.00	93.57	61.25	100
	7	72	19	28	681	94.13	79.12	72.00	97.29	75.39	100
DLCA-CapsNet (Mango)	0	100	0	0	700	100	100	100	100	100	100
	1	97	0	3	700	99.63	100	97.00	100	98.48	100
	2	100	0	0	700	100	100	100	100	100	100
	3	100	0	0	700	100	100	100	100	100	100
	4	100	0	0	700	100	100	100	100	100	100
	5	99	0	1	700	99.88	100	99.00	100	99.50	100
	6	100	4	0	696	99.50	96.15	100	99.43	98.04	100
	7	100	0	0	700	100	100	100	100	100	100

B. *Number of Parameters and Size of Disk*

Increasing model complexity (by adding layers or increasing the size of layers) often improves performance on intricate images. However, this makes the models larger and computationally demanding, hindering their use on devices with limited resources like phones and embedded systems. DLCA-CapsNet is smaller in size and has fewer parameters than the original CapsNet and other top models (see Table III). Additionally, the CDH component we used does not add any extra parameters. The DLCA-CapsNet achieves a notable decrease in parameter count (in millions (M)) by 6.16M, 6.16M, 6.16M, 6.16M, 7.14M, 5.68M, 5.92M, 7.62M, 7.62M, and 6.54M respectively, when compared to the conventional CapsNet considering apple, banana, grape, corn, mango, pepper, potato, rice, tomato, CIFAR-10, and Fashion-MNIST datasets. Furthermore, it results in a reduction in disk space usage by 23.5MB, 23.5MB, 23.5MB, 23.5MB, 27.2MB, 21.6MB, 22.6MB, 23.5MB, 29.1MB, 29.1MB, and 24.8MB, respectively, for the same datasets.

TABLE III. COMPARISON OF DISK SIZE (S) IN MB AND NUMBER OF PARAMETERS (P) IN MILLIONS (M)

CapsNet Models/Reference	P(M) and S(MB)										
		Apple	Banana	Grape	Maize	Mango	Pepper	Potato	Rice	Tomato	CIFAR-10	Fashion-MNIST
Gabor CapsNet [14]	P	-	-	-	-	-	-	-	-	12.00	-	-
CapsNet[18]	P	-	-	-	8.40	-	-	-	-	8.40	5.20	2.80
K-Means CapsNet[37]	P	-	-	-	-	-	-	-	-	5.12	-	-
CapsNet [19]	P	-	-	-	-	-	-	9.86	-	-	-	-
Gabor-Maxpooled CapsNet [22]	P	-	-	-	-	-	-	-	-	8.71	-	-
Shallow/Multi-Input CapsNet [28]	P	-	-	-	-	-	-	-	-	4.10/ 4.00	4.60/ 4.30	2.50/ 2.20
Dual-Input CapsNet [24]	P	-	-	-	-	-	-	-	-	6.04	5.48	-
Original CapsNet [7]	P	10.13	10.13	10.13	10.13	11.21	9.59	9.86	10.13	11.75	11.75	8.22
	S	38.6	38.6	38.6	38.6	42.7	36.5	37.6	38.6	44.8	44.8	31.3
DCLA-CapsNet proposed	P	3.97	3.97	3.97	3.97	4.07	3.91	3.94	3.97	4.13	4.13	1.68
	S	15.1	15.1	15.1	15.1	15.5	14.9	15.0	15.1	15.7	15.7	6.5

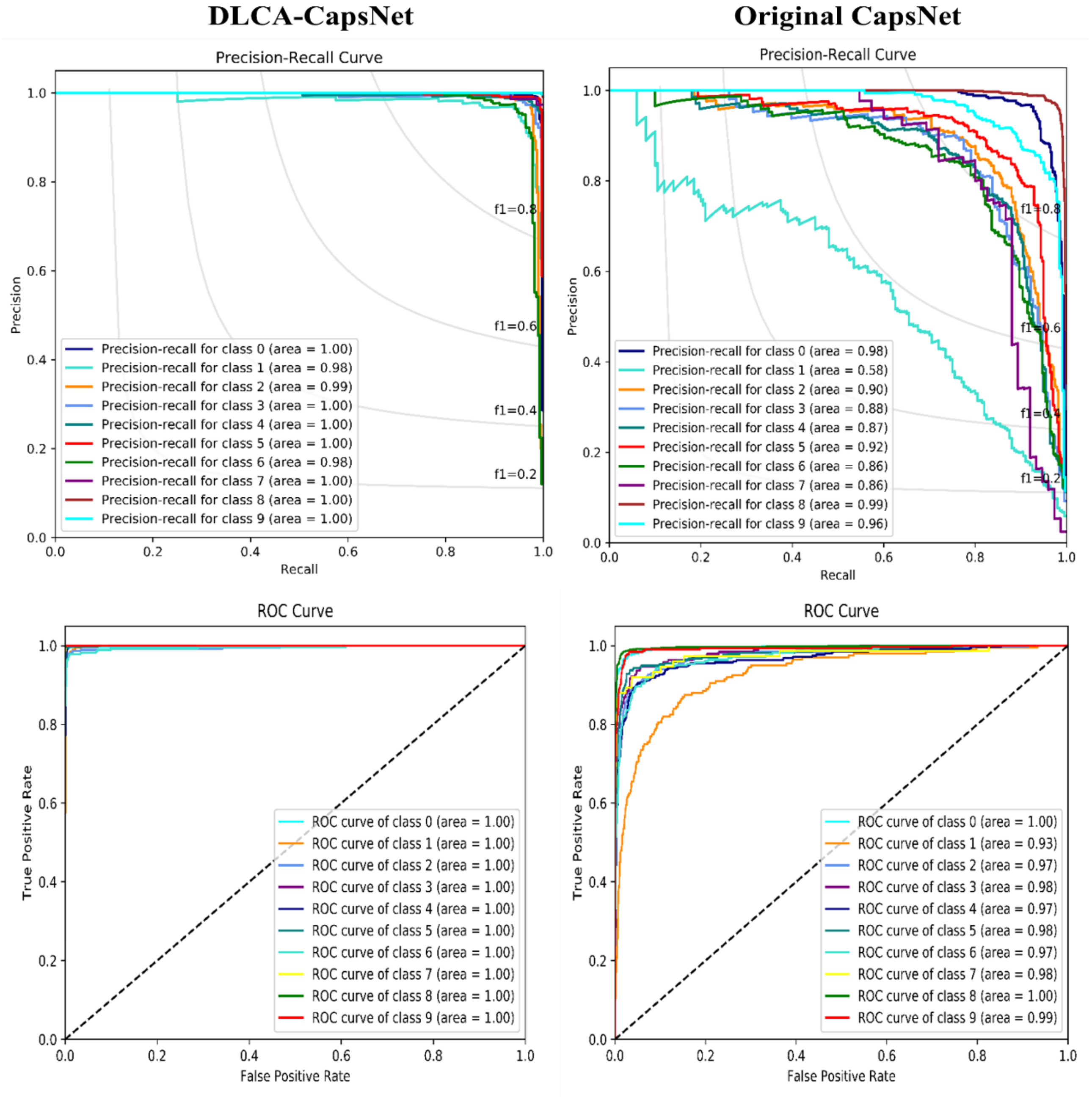


Fig. 4. PR and ROC curves comparing the performance of DLCA-CapsNet and Original CapsNet models on the Tomato dataset.

C. *Ablation Study*

To assess the model parts that affect its performance, an ablation study was performed [35]. The layers are removed one by one to check the layers of the model that significantly affect the performance of the model. Considering the tomato and potato leave diseases, it can be seen in Table IV that the model's performance is highly impacted by the CDH layer.

TABLE IV. ABLATION RESULTS

Layers	Validation accuracy (%)	
	Tomato	Potato
-CDH	82.12	90.17
-Atrous_Conv1/ Atrou_Conv2	97.75	95.95
-MP1 & MP3	98.62	98.88
-Conv1 & Conv2	97.88	97.17
-MP2 & MP4	98.75	98.28
-Dropout	98.88	99.14
+All Layers	98.82	100

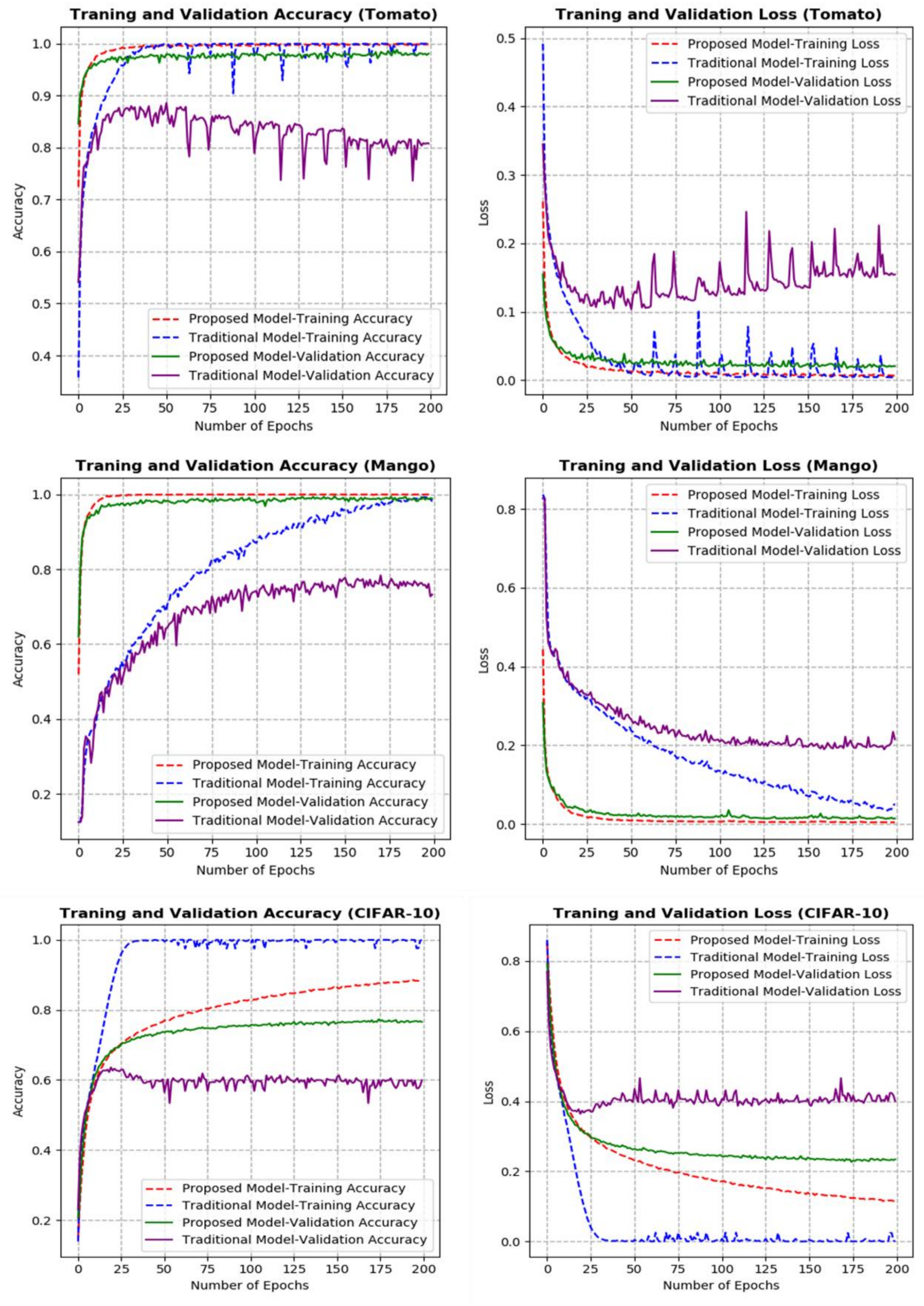


Fig. 5. Accuracy and loss performance for the proposed DLCA-CapsNet and the original CapsNet models across three datasets: tomato, mango, and CIFAR-10.

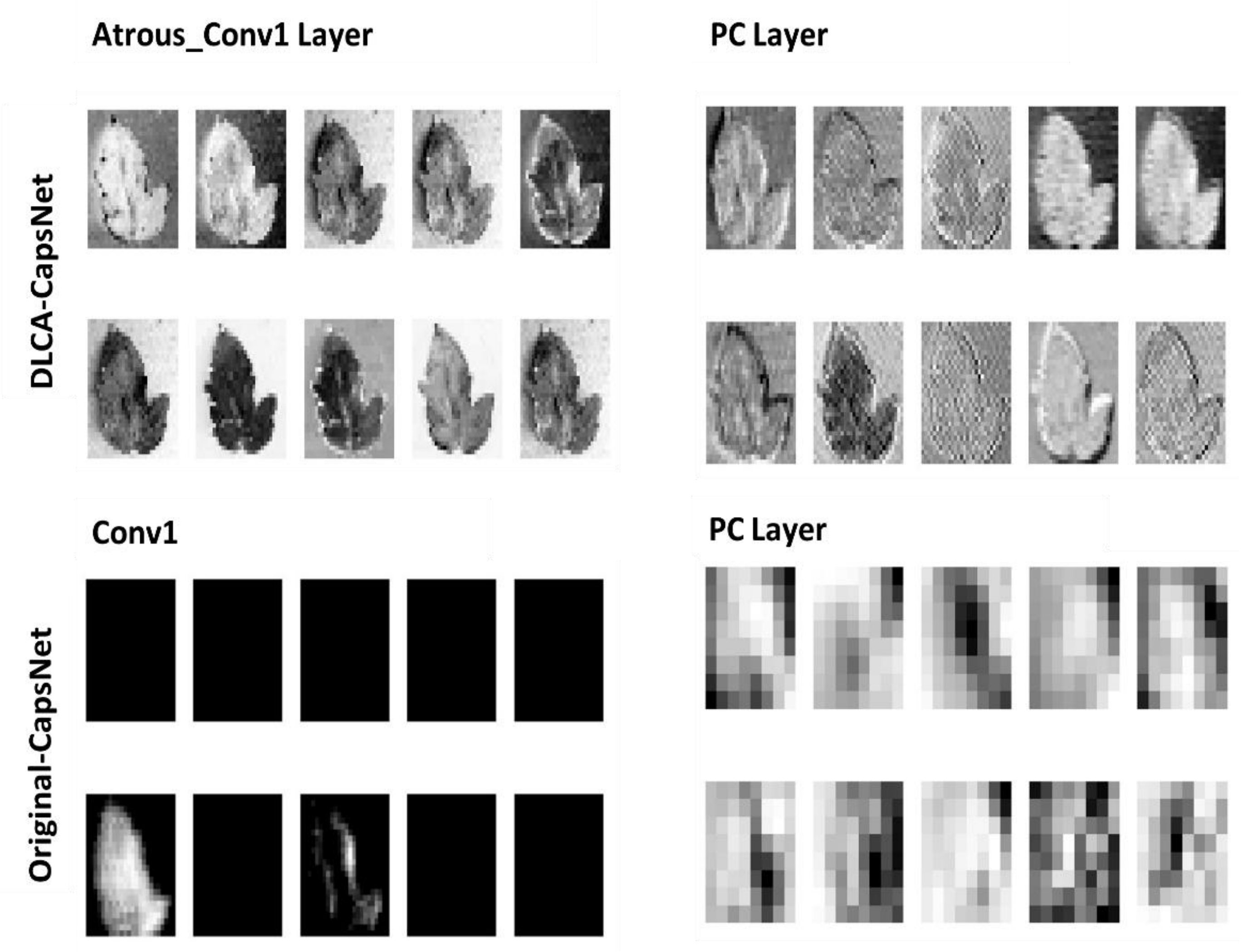


Fig. 6. Activation maps of the DLCA-CapsNet and the original CapsNet models on the Tomato dataset.

D. *Model Interpretability*

Fig. 6 visually compares activation maps from one of the atrous convolution layers in the proposed DLCA-CapsNet model (which takes input from the CDH layer using the tomato dataset) to the activation map from the convolution layer of the traditional CapsNet. A comparison of these maps reveals that the atrous convolution in the DLCA-CapsNet captures more detailed features, suggesting that the CDH layer plays a crucial role in extracting significant information, something the convolution layer of the traditional CapsNet alone fails to achieve. Furthermore, comparing the activation maps from the primary capsule layer of the proposed DLCA-CapsNet and the original model reveals that DLCA-CapsNet captures more relevant features. This improvement is attributed to the proposed model's ability to extract significant features earlier in the network, which enhances the quality of the features passed to its primary capsule layer, unlike the traditional CapsNet, whose convolutional layer was less effective in feature extraction.

The clusters generated at the class capsule layers were also visualized using t-distributed stochastic neighbor embedding (t-SNE). As shown in Fig. 7, considering the DLCA-CapsNet model, the clusters at the class capsule layer are more distinctly grouped by class, with fewer outliers compared to the traditional model, particularly for the tomato, mango, and pepper datasets. This indicates that the DLCA-CapsNet model demonstrates a stronger ability to distinguish between different classes in the dataset than the traditional model. Also, the reconstruction technique aids in identifying the predicted class of an image and the confidence of that prediction. Referring to Fig. 8, which presents three rows of reconstructed images from the tomato and banana datasets for both the proposed and traditional CapsNet models, it is evident that the proposed model generates images of slightly better quality with higher class probabilities. These visual outputs from the model layers enhance interpretability and support the goals of Explainable Artificial Intelligence (XAI).

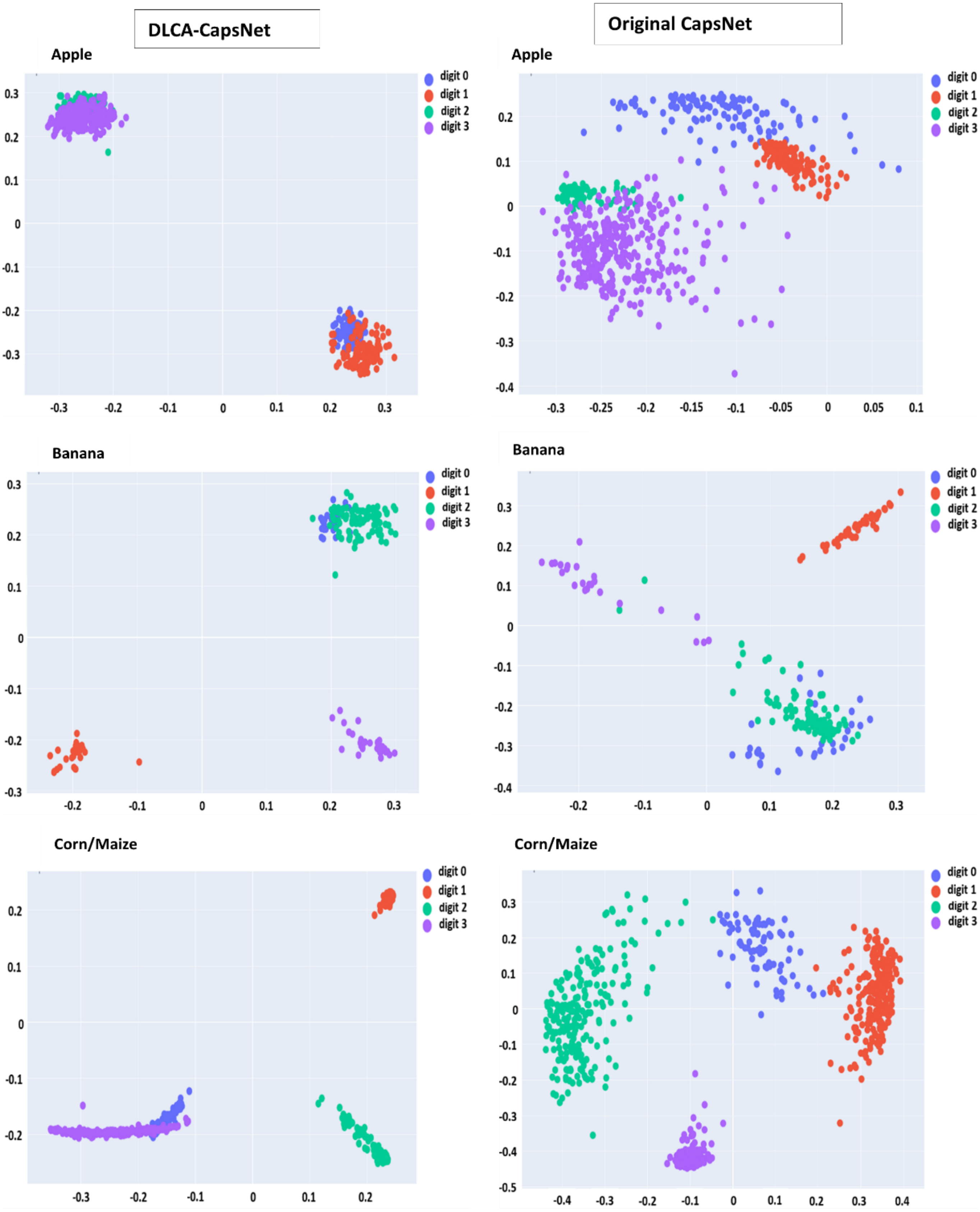
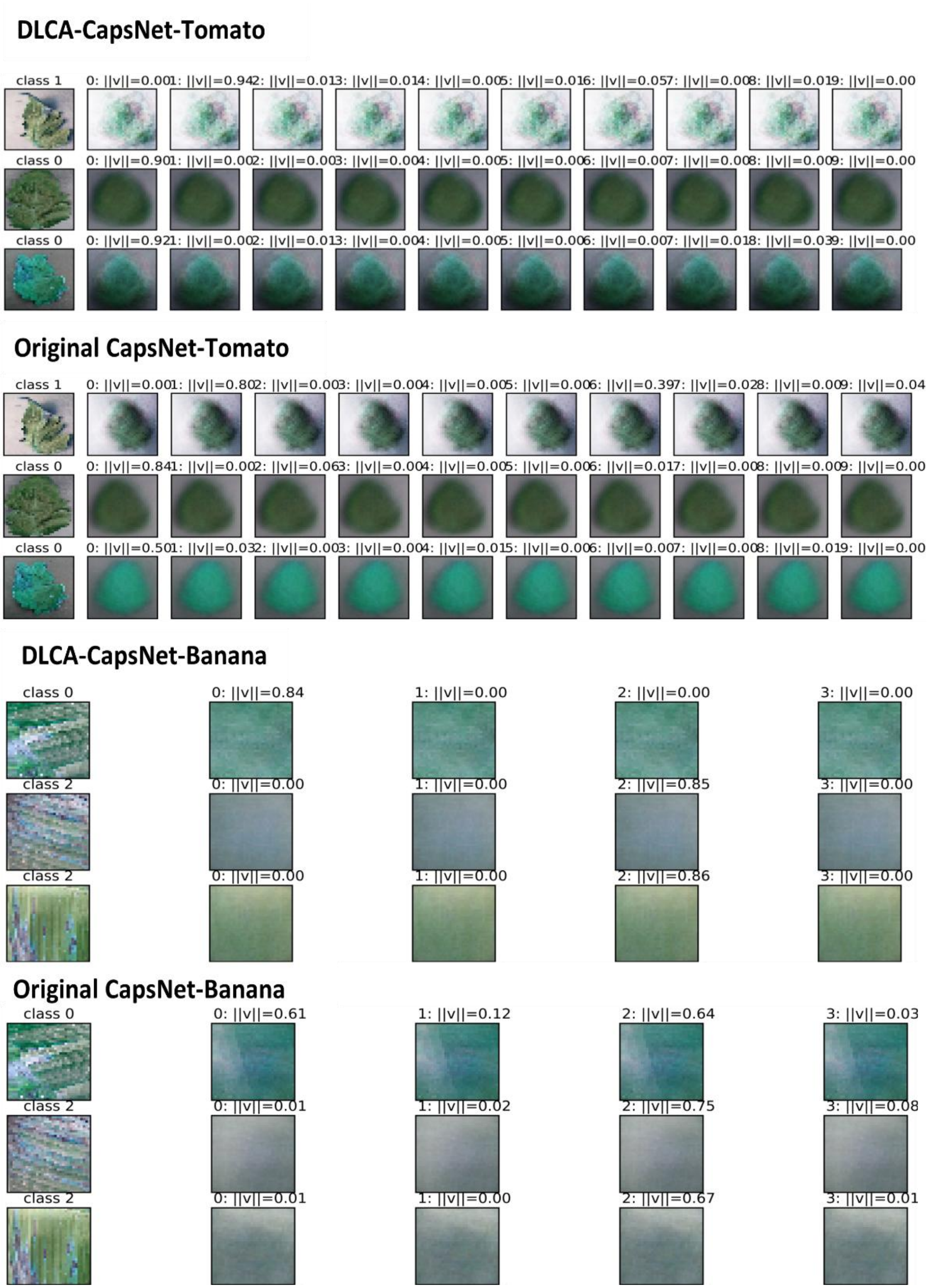


Fig. 7. Class capsule clusters for Apple, Banana, and Corn in DLCA-CapsNet and the original CapsNet models.



DLCA-CapsNet-Tomato

class 1
0: ||v||=0.001: ||v||=0.942: ||v||=0.013: ||v||=0.014: ||v||=0.005: ||v||=0.016: ||v||=0.057: ||v||=0.008: ||v||=0.019: ||v||=0.00
class 0
0: ||v||=0.901: ||v||=0.002: ||v||=0.003: ||v||=0.004: ||v||=0.005: ||v||=0.006: ||v||=0.007: ||v||=0.008: ||v||=0.009: ||v||=0.00
class 0
0: ||v||=0.921: ||v||=0.002: ||v||=0.013: ||v||=0.004: ||v||=0.005: ||v||=0.006: ||v||=0.007: ||v||=0.018: ||v||=0.039: ||v||=0.00

Original CapsNet-Tomato

class 1
0: ||v||=0.001: ||v||=0.802: ||v||=0.003: ||v||=0.004: ||v||=0.005: ||v||=0.006: ||v||=0.397: ||v||=0.028: ||v||=0.009: ||v||=0.04
class 0
0: ||v||=0.841: ||v||=0.002: ||v||=0.063: ||v||=0.004: ||v||=0.005: ||v||=0.006: ||v||=0.017: ||v||=0.008: ||v||=0.009: ||v||=0.00
class 0
0: ||v||=0.501: ||v||=0.032: ||v||=0.003: ||v||=0.004: ||v||=0.015: ||v||=0.006: ||v||=0.007: ||v||=0.008: ||v||=0.019: ||v||=0.00

DLCA-CapsNet-Banana

class 0
0: ||v||=0.84
1: ||v||=0.00
2: ||v||=0.00
3: ||v||=0.00
class 2
0: ||v||=0.00
1: ||v||=0.00
2: ||v||=0.85
3: ||v||=0.00
class 2
0: ||v||=0.00
1: ||v||=0.00
2: ||v||=0.86
3: ||v||=0.00

Original CapsNet-Banana

class 0
0: ||v||=0.61
1: ||v||=0.12
2: ||v||=0.64
3: ||v||=0.03
class 2
0: ||v||=0.01
1: ||v||=0.02
2: ||v||=0.75
3: ||v||=0.08
class 2
0: ||v||=0.01
1: ||v||=0.00
2: ||v||=0.67
3: ||v||=0.01

Fig. 8. Reconstructed images generated by the DLCA-CapsNet and Original CapsNet models when applied to the Tomato and Banana datasets.

E. Comparison of Results

Table V compares the proposed DLCA-CapsNet model and state-of-the-art models applied to CIFAR-10, Fashion-MNIST, and the nine plant disease datasets from the literature. The comparison also includes various routing algorithms, despite the DLCA-CapsNet model utilizing a dynamic routing technique. Table V shows that the less parameterized DLCA-CapsNet model, with enhanced feature extraction capabilities, achieved higher validation accuracies across various datasets, surpassing the original CapsNet by 6.78%, 14.82%, 6.14%, 5.07%, 21.12%, 40.32%, 4.64%, 0.76%, 10.23%, 13.73%, and 2.03% for the apple, banana, grape, corn, mango, pepper, potato, rice, tomato, CIFAR-10, and Fashion-MNIST datasets, respectively. It also resulted in a reduction of 6.16M, 6.16M, 6.16M, 6.16M, 7.14M, 5.68M, 5.92M, 7.62M, 7.62M, and 6.54M in model size (in millions) compared to the traditional CapsNet. Furthermore, the DLCA-CapsNet model led to a decrease in disk size by 23.5MB, 23.5MB, 23.5MB, 23.5MB, 27.2MB, 21.6MB, 22.6MB, 23.5MB, 29.1MB, 29.1MB, and 24.8MB for the same datasets compared to the traditional CapsNet, as shown in Table III. Also, comparing the proposed DLCA-CapsNet with existing models found in the literature for plant disease detection, the DLCA-CapsNet also outperformed them, as shown in Table V and Table III. These results demonstrate the superior generalization ability and lower computational complexity of the DLCA-CapsNet over the original CapsNet. This exceptional performance, necessary for complex images, can be credited to the superior feature extraction methods we used to isolate only the most relevant features from the images.

TABLE V. COMPARISON BETWEEN PREVIOUS STUDIES AND THE PROPOSED DCLA-CAPSNET MODELS

CapsNet Models/Reference	**Validation Accuracy (%)**										
	Apple	***Banana***	***Grape***	***Corn***	***Mango***	***Pepper***	***Potato***	***Rice***	***Tomato***	***CIFAR-10***	***Fashion-MNIST***
Gabor CapsNet [14]	-	-	-	-	-	-	-	-	98.13	-	-
E-GAN CapsNet [16]	-	-	97.63	-	-	-	-	-	-	-	-
CapsNet [18]	-	-	-	96.79	-	-	-	-	98.06	75.80	92.72
Dilated CapsNet [17]	93.16	-	-	-	-	-	-	-	-	-	-
K-Means CapsNet [37]	-	-	-	97.99	-	-	-	-	98.80	-	-
ConvCapsNet [29]	-	-	99.12	-	-	-	-	-	-	-	-
CapsNet [19]	-	-	-	-	-	-	91.83	-	-	-	-
CapsNet [21]	-	-	-	-	-	95.76	-	-	-	-	-
CapsNet [20]	-	95.00	-	-	-	-	-	-	-	-	-
Gabor-Maxpooled CapsNet [22]	-	-	-	-	-	-	-	-	97.98	-	-
Shallow/Multi-Input CapsNet [28]	-	-	-	-	-	-	-	-	97.33/ 94.04	75.75/ 63.95	92.7/ 91.45
Dual-Input CapsNet [24]	-	-	-	-	-	-	-	-	93.03	76.58	-
Multi-Channel CapsNet [25]	-	-	-	-	-	-	-	-	98.15	-	-
CapsNet [26]	-	-	-	-	-	-	-	-	96.39	-	-
SE-SK CapsNet [27]	-	-	-	-	-	-	-	-	97.19	-	-
Original CapsNet [7]	92.91	80.95	93.49	92.59	78.38	59.68	95.36	99.24	88.59	63.58	90.98
DCLA-CapsNet Proposed	99.69	95.77	99.63	97.66	99.50	100	100	100	98.82	77.31	93.01

V. CONCLUSION

This study proposed an optimized DLCA-CapsNet model for classifying plant diseases, CIFAR-10, and fashion-MNIST. Modifications were made by adding a CDH, which does not add any parameters, atrous convolutions, max-pooling, or a dropout layer. All these layers contributed to the efficient feature extraction abilities of the proposed model. The DLCA-CapsNet model was assessed using evaluation metrics such as sensitivity, F1-score, precision, specificity, ROC and PR values, accuracy, disk size, and parameters generated. The DLCA-CapsNet model results were compared with those of the traditional CapsNet and other models found in the literature, and outperformed them as shown in Table III and Table V. The proposed DLCA-CapsNet model's fewer parameters make it usable on IoT and resource-constrained devices, and the better validation accuracies show its generalization ability on unseen data. An ablation study was performed to ascertain the layers of the model that influence its performance. Again, the model interpretability regarding visualizing clusters at the class capsule, activation maps, and reconstruction of images was discussed. The proposed model's better performance shows its effectiveness in detecting plant diseases from plant leaf images than those found in literature as analyzed in Table III and Table V. The findings suggest that this efficient and computationally less demanding method can significantly enhance plant disease classification and contributes incrementally to efforts aligned with the SDG 2 goal by offering a lightweight, scalable solution that can be adapted for field use in resource-constrained settings.

Nonetheless, environmental variables like uneven lighting and intricate backgrounds in real-world conditions can hinder or limit model performance. Subsequent research will aim to enhance generalizability and real-world applicability by evaluating the model across more diverse, challenging settings, incorporating additional datasets, and investigating real-time implementation on edge and mobile platforms for agricultural use.

REFERENCES

[1] T. Li et al, "Applications of Deep Learning in Fundus Images: A Review," *Medical Image Analysis*, 69, 101971, 2021.

[2] X. S. Wei et al, "Fine-Grained Image Analysis with Deep Learning: A Survey," *IEEE transactions on pattern analysis and machine intelligence*, 44(12), 8927-8948, Nov. 2021.

[3] X. Chen et al, "Recent advances and clinical applications of deep learning in medical image analysis," *Medical image analysis,* 79: 102444, 2021.

[4] A. Farahat, F. Effenberger, and M. Vinck, "A novel feature-scrambling approach reveals the capacity of convolutional neural networks to learn spatial relations," *Neural Networks*, vol. 160, pp. 400-414, Aug. 2023.

[5] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *J. Imaging*, vol. 6, no. 6, pp. 1–19, 2020, doi: 10.3390/JIMAGING6060052.

[6] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.

[7] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Appl. Biosaf.*, vol. 22, no. 4, pp. 185–186, 2017, doi: 10.1177/1535676017742133.

[8] M. Mitterreiter, M. Koch, J. Giesen, and S. Laue, "Why Capsule Neural Networks Do Not Scale: Challenging the Dynamic Parse-Tree Assumption," in *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI),* vol. 37, no. 8, 2023.

[9] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-CapsNet: Capsule Network with Self-Attention Routing," *Scientific reports,* 11.1: 14634, 2021.

[10] S. Cao, Y. Yao, and G. An, "E2-capsule neural networks for facial expression recognition using AU-aware attention," *IET Image Process.*, vol. 14, no. 11, pp. 2417–2424, 2020, doi: 10.1049/iet-ipr.2020.0063.

[11] G. H. Liu and J. Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognit.*, vol. 46, no. 1, pp. 188–198, 2013, doi: 10.1016/j.patcog.2012.06.001.

[12] S. Qiao, L. C. Chen, and A. Yuille, "Detectors: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 10213-10224, 2021.

[13] L. Zhang, J. Zhang, Z. Li, and Y. Song, "A multiple-channel and atrous convolution network for ultrasound image segmentation," *Medical Physics*, vol. 47, no. 12, pp. 6270–6285, Dec. 2020.

[14] P. K. Mensah, B. A. Weyori, and M. A. Ayidzoe, "Gabor Capsule Network for Plant Disease Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 388–395, 2020.

[15] S. Verma, A. Chug, R. P. Singh, A. P. Singh, and D. Singh, "SE-CapsNet : Automated evaluation of plant disease severity based on feature extraction through Squeeze and Excitation (SE) networks and Capsule networks University School of Information , Communication & Technology (USIC & T), Guru Gobind Singh Indra," *Kuwait J. Sci.*, vol. 49, no. 1, pp. 1–31, 2022.

[16] N. Vasudevan and T. Karthick, "A Hybrid Approach for Plant Disease Detection Using E-GAN and CapsNet," *Comput. Syst. Sci. Eng.*, vol. 46, no. 1, pp. 337–356, 2023, doi: 10.32604/csse.2023.034242.

[17] C. Xu, X. Wang, and S. Zhang, "Dilated convolution capsule network for apple leaf disease identification," *Front. Plant Sci.*, vol. 13, no. November, pp. 1–13, 2022, doi: 10.3389/fpls.2022.1002312.

[18] P. Mensah, B. Asubam, and A. Abra, "Exploring the performance of LBP-capsule networks with K-Means routing on complex images," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2574–2588, 2022, doi: 10.1016/j.jksuci.2020.10.006.

[19] S. Verma, A. Chug, and A. P. Singh, "Exploring capsule networks for disease classification in plants," *J. Stat. Manag. Syst.*, vol. 23, no. 2, pp. 307–315, 2020, doi: 10.1080/09720510.2020.1724628.

[20] B. F. Oladejo and O. O. Ademola, "Automated Classification of Banana Leaf Diseases using an Optimized Capsule Network Model," pp. 119–130, 2020, doi: 10.5121/csit.2020.100910.

[21] G. ALTAN, "Performance Evaluation of Capsule Networks for Classification of Plant Leaf Diseases," *Int. J. Appl. Math. Electron. Comput.*, vol. 8, no. 3, pp. 57–63, Sep. 2020, doi: 10.18100/ijamec.797392.

[22] P. K. Mensah, B. A. Weyori, and A. A. Mighty, "Max-pooled fast learning gabor capsule network," *2020 Int. Conf. Artif. Intell. Big Data, Comput. Data Commun. Syst. icABCD 2020 - Proc.*, 2020, doi: 10.1109/icABCD49160.2020.9183823.

[23] A. Anant, "AppleCaps : A Capsule Model for Classification of Foliar Diseases in Apple Leaves Rakhi Ashok Sonkusare National College of Ireland Supervisor :," *Natl. Coll. Irel.*, 2021.

[24] P. K. Mensah and M. A. Ayidzoe, "Overview of CapsNet Performance Evaluation Methods for Image Classification using a Dual Input Capsule Network as a Case Study," *Int. J. Comput. Digit. Syst.*, vol. 1, no. 1, 2022.

[25] M. Peker, "Multi-channel capsule network ensemble for plant disease detection," *SN Appl. Sci.*, vol. 3, no. 7, 2021, doi: 10.1007/s42452-021-04694-2.

[26] L. M. Abouelmagd, M. Y. Shams, H. S. Marie, and A. E. Hassanien, "An optimized capsule neural networks for tomato leaf disease classification," *Eurasip J. Image Video Process.*, vol. 2024, no. 1, 2024, doi: 10.1186/s13640-023-00618-9.

[27] X. Zhang, Y. Mao, Q. Yang, and X. Zhang, "A Plant Leaf Disease Image Classification Method Integrating Capsule Network and Residual Network," *IEEE Access*, vol. 12, no. February, pp. 44573–44585, 2024, doi: 10.1109/ACCESS.2024.3377230.

[28] P. K. Mensah, B. A. Weyori, and M. A. Ayidzoe, "Evaluating shallow capsule networks on complex images," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 1047–1057, 2021, doi: 10.1007/s41870-021-00694-y.

[29] A. D. Andrushia, T. M. Neebha, A. T. Patricia, S.Umadevi, N.Anand, and A. Varshney, "Image Based Disease Classiication in Grape Leaves Using Convolutional Capsule Network Image based Disease Classification in Grape Leaves using Convolutional Capsule Network," *Soft Comput.*, 2022.

[30] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015.

[31] S. E. Arman, M. A. B. Bhuiyan, H. M. Abdullah, S. Islam, T. T. Chowdhury, and M. A. Hossain, "BananaLSD: A banana leaf images dataset for classification of banana leaf diseases using machine learning," *Data Br.*, vol. 50, p. 109608, 2023, doi: 10.1016/j.dib.2023.109608.

[32] S. I. Ahmed *et al.*, "MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves," *Data Br.*, vol. 47, p. 108941, 2023, doi: 10.1016/j.dib.2023.108941.

[33] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *Mach. Learn.*, pp. 1–6, 2017.

[34] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," *ASHA*, vol. 34, no. 4, 2009.

[35] M. Li and L. Janson, "Optimal Ablation for Interpretability," *Advances in Neural Information Processing Systems*, vol 37, pp.109233-109282, Sep. 16, 2024.

[36] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera. "Deep feature-based rice leaf disease identification using support vector machine." *Computers and Electronics in Agriculture* 175 (2020):105527, doi: 10.1016/j.compag.2020.105527

[37] K. P. Mensah, B. A. Weyori, and A. M. Ayidzoe, "Capsule network with K-Means routing for plant disease recognition," *J. Intell. Fuzzy Syst.*, pp. 1–12, 2020, doi: 10.3233/JIFS-201226.