This study focuses on tomato leaf disease classification using an optimized deep learning architecture. This study proposes an improved architecture called DenseNet-SEGR, which integrates a novel Squeeze-and-Excitation (SE) block with a customized growth rate of 48 to improve feature selection and classification accuracy. Unlike standard methods, this model replaces Global Average Pooling (GAP) with an integral-based squeeze method, thus enabling a more continuous and accurate feature representation. The use of SE blocks dynamically recalibrates the importance of features such as texture, color, and tissue patterns, thereby increasing sensitivity to disease symptoms. The model was trained using the PlantVillage dataset, which includes 12,246 images spanning 10 tomato leaf disease categories, such as bacterial spot, early blight, late blight, mosaic virus, and healthy leaves. Various augmentation techniques, including rotation, scaling, and contrast adjustment, were employed to strengthen generalization and improve robustness against environmental variations. Furthermore, batch normalization and adaptive learning rate scheduling were integrated to enhance model stability and prevent overfitting. As a result, the DenseNet-SEGR architecture is able to achieve a classification accuracy of 98.22 %, outperforming DenseNet-121, DenseNet-201, and MobileNetV2. This result is explained by the integration of adaptive attention mechanisms, sophisticated data augmentation strategies, and optimized architecture. The results can be effectively applied in real-world precision agriculture, especially in edge-based or mobile disease detection systems for early intervention and crop protection

Keywords: tomato classification, tomato leaf disease, DenseNet-SEGR, squeeze-and-excitation, growth rate, deep learning

UDC 004

DOI: 10.15587/1729-4061.2025.323176

DENSENET DEVELOPMENT WITH SQUEEZE-AND-EXCITATION BLOCK FOR TOMATO PLANT DISEASE CLASSIFICATION

Pandi Barita Nauli Simangunsong
Doctoral Program of Computer Science, Doctoral Student*

Poltak Sihombing
Doctor of Philosophy (PhD) in Computer Science, Main Supervisor*

Syahril Efendi
Doctor of Philosophy (PhD) in Mathematics, Co-Supervisor*

Fahmi Fahmi
Corresponding author
PhD in Biomedical Engineering, Co-Supervisor
Department of Electrical Engineering**
E-mail: fahmimn@usu.ac.id

*Department of Computer Science**
**Universitas Sumatera Utara
Dr. T. Mansur str., 9, Padang Bulan, North Sumatera, Indonesia, 20222

Received 24.01.2025
Received in revised form 12.03.2025
Accepted date 31.03.2025
Published date 22.04.2025

How to Cite: *Simangunsong, P. B. N., Sihombing, P., Effendi, S., Fahmi, F. (2025). DenseNet development with Squeeze-And-Excitation block for tomato plant disease classification. Eastern-European Journal of Enterprise Technologies, 2 (2 (134)), 28–38. https://doi.org/10.15587/1729-4061.2025.323176*

1. Introduction

Tomato leaf diseases contribute to global crop losses of 20–40 % (FAO), highlighting the need for effective treatment to maintain agricultural quality. However, limited diverse datasets hinder detection accuracy. Therefore, developing a more accurate and applicable detection model is essential, making this research highly relevant for improving agricultural solutions [1].

The need for a lightweight and rapid architecture for tomato foliar disease classification is critical, as disease-related production declines threaten global food security. Traditional methods often prove ineffective, and farmers in remote areas struggle with disease detection. In addition, existing solutions have limitations in running on low-power devices. Therefore, developing a lightweight and accurate model is essential for increasing agricultural yields [2]. Deep learning models can be adapted to recognize image variations, such as differences in lighting, shooting angles, and backgrounds [3].

Applying more effective preprocessing techniques, such as noise reduction and contrast enhancement, can improve the quality of the input images while improving the model's overall performance [4].

Squeeze-and-Excitation (SE) blocks play a crucial role in improving training efficiency by preventing gradient loss and increasing the model's sensitivity to essential features by providing adaptive weights, enabling the model to focus on relevant information [5, 6].

Densely Connected Convolutional Network (DenseNet), despite its advantages, has a complex architecture that leads to longer training times and higher computational demands, making it less efficient for large datasets or high-resolution images [7–9]. Additionally, its ability to generalize decreases when handling diverse and complex data, affecting classification accuracy. SE blocks have been shown to improve model performance by considering inter-channel relationships, yet little research has explored their integration into Densely Connected Convolutional Network- Squeeze-and-Excitation and Growth Rate (DenseNet-SEGR) for detecting tomato plant diseases, highlighting the need for further exploration [10].

Integrating SE blocks into DenseNet-SEGR enhances tomato plant disease classification by improving accuracy, reducing overfitting, and optimizing computation [11]. SE blocks also refine feature analysis, stabilize training, and improve adaptability to visual variations. Performance evaluation using accuracy, precision, recall, and F1-score, along with comparisons to models without SE blocks, confirms its effectiveness. Additionally, data augmentation techniques like rotation, cropping, rescaling, and color adjustment enhance dataset diversity, helping the model capture data variations more effectively [12].

Data augmentation also helps address class imbalance issues, ensuring better disease recognition and improved classification accuracy. The use of diverse datasets, such as tomato leaf image datasets covering various disease conditions, is crucial for training models that can adapt to real-world variations [13, 14]. Training a typical tomato leaf classification model uses thousands of images covering a wide range of leaf conditions, including diseases such as bacterial spot, early blight, and mosaic virus [1, 6, 15].

This research also contributes to the theoretical understanding of the effect of data augmentation techniques on model performance. By exploring various augmentation methods, this research can provide the best practical guidance for preparing optimal datasets for training deep learning models. Deep learning requires high (Graphics Processing Unit) GPU resources for computation during the deep learning trainer process [17], optimizing architectures and preprocessing techniques becomes even more critical.

Therefore, research on developing a lightweight, efficient, and accurate deep learning model for tomato leaf disease classification is highly relevant to improving agricultural solutions and ensuring global food security.

2. Literature review and problem statement

The DenseNet architecture has been shown to provide competitive results in various image classification tasks, including tomato plant disease classification [10]. and even in medical and biological image analysis [18]. One enhancement introduced in recent research is the Squeeze-and-Excitation (SE) module, which adaptively reweights feature channels based on their importance. Studies have shown that integrating SE blocks into convolutional neural networks (CNNs) can improve feature representation and boost classification accuracy across several domains [19], including agriculture and histopathology.

Despite these advances, several unresolved issues remain in the literature. First, while DenseNet combined with SE blocks (SE-DenseNet) can enhance sensitivity to relevant features [19]. Many studies fail to optimize architectural parameters for specific domains like plant pathology. For instance, the impact of varying growth rates, SE placement strategies, or customized feature recalibration for tomato leaf datasets is not widely addressed [12–23]. Additionally, although SE blocks improve performance, their inclusion increases model complexity and computational cost, making it harder to deploy such models in real-time or resource-constrained agricultural settings [24, 25]. Furthermore, most publicly available tomato disease datasets are captured under controlled conditions, lacking the diversity in lighting, leaf orientation, and backgrounds found in real agricultural environments. This leads to poor generalization in the field [1, 2, 13, 14]. The limited size and diversity of labeled datasets also contribute to overfitting problems in deep learning models [20, 21]. While techniques like data augmentation have been proposed to mitigate this [12], their implementation in existing literature is often generic and lacks domain-specific adaptation.

These issues persist due to both objective reasons, such as the scarcity of large, diverse, labeled datasets and the high computational demands of deep models like DenseNet, and subjective reasons, such as the focus of previous researchers on improving accuracy in ideal conditions rather than considering real-world deployment constraints like inference speed or device limitations [2, 15, 17]. Therefore, the main unresolved problem emerging from the reviewed literature is: the absence of a lightweight, accurate, and adaptable tomato disease classification model that generalizes well to environmental variation and can be efficiently deployed on real-time or edge-based systems.

There is a lack of a lightweight, accurate, and adaptable tomato leaf disease classification model that can generalize to diverse real-world conditions and is suitable for deployment in resource-constrained environments. This gap motivates the current research to develop an enhanced deep learning model that addresses both accuracy and practical deployment constraints.

3. The aim and objectives of the study

The aim of this study is to develop a scientifically optimized deep learning classification model for tomato leaf diseases by modifying the DenseNet architecture with Squeeze-and-Excitation (SE) blocks and a customized growth rate, in order to address limitations in accuracy and generalization found in prior models.

To achieve this aim, the following objectives are accomplished:

– to preprocess the data by resizing tomato leaf images to a standardized dimension and applying data augmentation techniques to improve model generalization and reduce the risk of overfitting;

– to modify the DenseNet architecture by utilizing a growth rate of 48, optimizing feature propagation and computational efficiency;

– to integrate SE Blocks within each Dense Block in the DenseNet architecture to improve feature selection and enhance classification performance.

4. Materials and methods

4. 1. Object and hypothesis of the study

This study focuses on a dataset of tomato leaf images containing various disease categories, used to develop and evaluate a deep learning classification model. The main hypothesis is that integrating Squeeze-and-Excitation (SE) blocks into the DenseNet architecture, along with preprocessing and augmentation, will improve classification accuracy and generalization compared to standard models like DenseNet-121 and MobileNetV2. The study assumes that disease symptoms on tomato leaves show distinct visual patterns that can be captured by convolutional neural networks. It also assumes that the dataset is representative enough for generalization. Simplifications include using clean images captured under controlled conditions and focusing only on single-label classification without incorporating environmental or contextual data.

4. 2. Preprocessing and augmentation data

In this study, data preprocessing and augmentation were applied to enhance the quality and diversity of the dataset before training the DenseNet-SEGR model. Preprocessing involved resizing all tomato leaf images to 224×224 pixels for standardization, normalizing pixel values to [0, 1], and applying noise reduction if necessary. To improve model generalization and reduce overfitting, data augmentation techniques were used, including rotation (±25°), flipping, zooming (up to 20 %), brightness adjustments, and translat tion to simulate real-world variations. These steps ensured a more balanced and diverse dataset, enabling the model to learn robust disease features effectively. As shown in Fig. 1, the processed dataset was then fed into the classification pipeline, ultimately distinguishing between Healthy and Not Healthy leaves with improved accuracy and reliability.

4. 3. Dataset

The dataset used in this research uses a secondary dataset from PlantVillage. The tomato leaf dataset in this study consists of 10.639 training samples, 1.607 validation samples and 3211 samples. The dataset is evaluated using standard deviation as a reference for dataset stability before model testing. The dataset in this study uses 10 tomato leaf classes. Dataset dataset can be accessed from website: https://www.kaggle.com/datasets/emmarex/plantdisease.

The proposed classification framework is illustrated in Fig. 1, which outlines the sequential steps from data preprocessing to classification.

This flow illustrates the classification process using the DenseNet-SEGR model. Starting with a dataset, it goes through a preprocessing stage, which includes normalization and image resizing to ensure the data is ready to use. After preprocessing, the data undergoes augmentation, such as rotation and flipping, to expand the dataset's variation and improve the model's generalization ability. Next, the processed data is fed to the DenseNet-SEGR model, modified with a Squeeze-and-Excitation (SE) block to improve feature efficiency. The model then predicts the image category through a classification stage, dividing the results into Healthy and Unhealthy classes. This flow is designed to accurately analyze the health of tomato leaf plants based on input images.

4. 4. Model architecture

The DenseNet-SEGR architecture model is a modification of DenseNet that integrates Squeeze-and-Excitation (SE) blocks to improve feature relevance. The model consists of 4 dense blocks with 58 convolutional layers, each with a growth rate of 48. After each Dense Block, there is a Transition Layer that uses batch normalization, Rectified Linear Unit (ReLU) as the activation function, and 1×1 convolution to reduce the number of dimensions of the features. The initial block starts with a convolution of 7×7 with a stride of 2, followed by batch normalization, ReLU, and max pooling 3×3. Each convolution layer in the Dense Block uses a 3×3 kernel with a stride of 1. The model ends with Global Average Pooling (GAP) to summarize the spatial features in one dimension and a fully connected layer to generate the final prediction based on the number of classes. DenseNet-SEGR is very good at classifying images with many different features because it has both the efficient DenseNet structure and the adaptive SE Block.

Squeeze-and-Excitation (SE) Block, where SE Block filters feature channel-wise with the formula:

a) squeeze (global average pooling) integral:

$$z_c = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j), \tag{1}$$

where z_c is the average intensity of pixels in the image, H and W are the height and width of the image, respectively, and $x_c(i, j)$ is the pixel value at coordinates (i, j);

b) excitation (fully connected layers):

$$s_c = \sigma\left(W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot z_c\right)\right), \tag{2}$$

where s_c is the result of the transformation, W_1 and W_2 are weight matrices, z_c is the input, ReLU is activation function, and σ is the sigmoid activation function;

c) reweighting :

$$X_c^{scaled} = s_c \cdot X_c, \tag{3}$$

where X_c^{scaled} is the scaled version of X_c, with scale factor s_c.

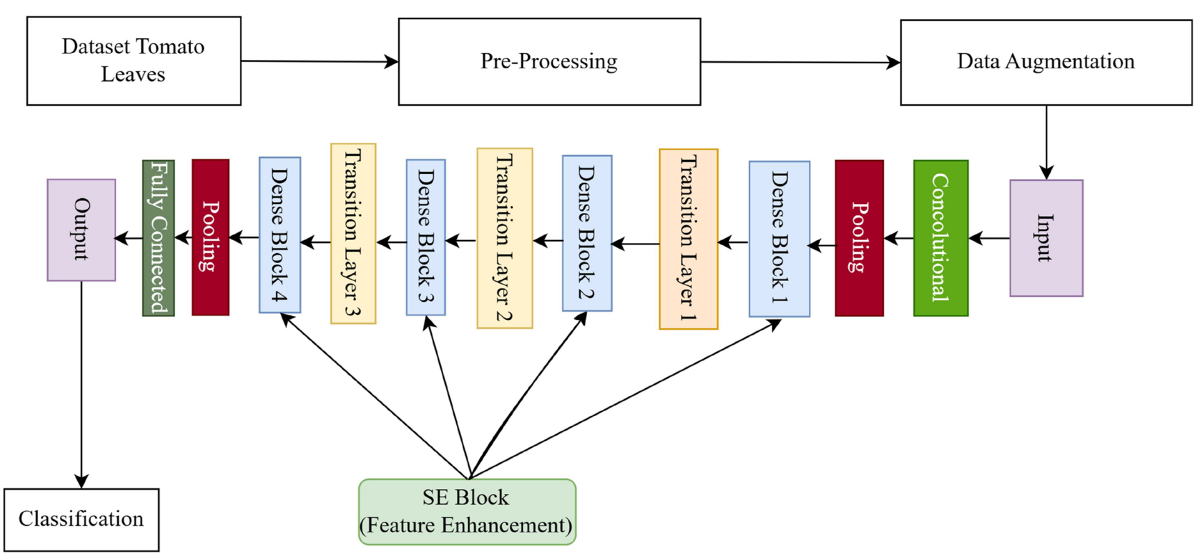

Fig. 1. Block diagram of overall system

Squeeze-and-Excitation (SE) blocks are integrated into the DenseNet-SEGR architecture by adding them after each Dense Block to improve feature representation. SE Block works through an adaptive attention mechanism to the feature channel. After the Dense Block generates features, the SE Block summarizes the global information of the features using the Global Average Pooling (GAP) operation. The global representation is then put through two fully connected layers that use ReLU and Sigmoid activations to make attention weights that show how important each feature channel is. These weights are used to recalibrate the feature channel through element-by-element scaling operations. Integrating SE Block in DenseNet-SEGR allows the model to amplify significant feature channels and attenuate less relevant features, thereby improving efficiency in understanding complex patterns in the data and producing a more informative feature representation for classification tasks.

4. 5. Training procedure

The training parameters used in the DenseNet-SEGR model are designed to optimize the training process and produce an accurate model. The number of epochs used is 50, which provides sufficient training time for the model to learn data patterns without overfitting. The batch size is set at 16, balancing computational efficiency and stability in weight updates during training. The initial learning rate is set at 1×10^{-4}, which is chosen to ensure that the weight update process is gradual and stable. Additionally, learning rate adjustments are carried out using schedulers such as Cosine Annealing, which dynamically decreases the learning rate value during training to improve model convergence. These parameters are combined with the AdamW optimizer, which has weight decay regulation capabilities, to control model complexity better and prevent overfitting. This combination of parameters is designed to ensure that training runs efficiently and produces optimal classification performance.

4. 6. Evaluation metrics

In evaluating the performance of the DenseNet-SEGR model, main metrics such as accuracy, precision, recall, and F1-score are used. Accuracy measures the percentage of correct predictions over the total sample, providing an overview of model performance. Precision assesses the model's ability to avoid false positives by calculating the ratio between true positives and total positive predictions. Recall or sensitivity measures the extent to which the model can detect all positive cases, while the F1-score, as the harmonic average of precision and recall, provides a balanced evaluation, especially when the class distribution is uneven. Model evaluation is done by dividing the dataset into training and validation sets. Cross-validation can be applied to increase the reliability of evaluation results by dividing the dataset into multiple folds and training the model on different combinations of training and validation sets. After training, the model is tested on the validation set to calculate evaluation metrics. Additionally, if the dataset is not large enough, data augmentation is applied to increase the diversity and number of training samples, thereby helping to reduce bias.

4. 7. Experimental setup

DenseNet-SEGR model training is carried out using hardware in the form of a processor (CPU) for light computing tasks and an NVIDIA GPU with CUDA support (such as the Tesla T4 in Google Colab) to speed up training [26]. A minimum of 12 GB RAM is used for batch processing of data, with data and model storage in Google Drive. The main software includes PyTorch for model development, CUDA for GPU acceleration, and supporting libraries such as torch-vision for data preprocessing, scikit-learn for metric evaluation, and matplotlib for visualization. Cloud-based development environment using Google Colab with Python 3.8+wa is used to leverage data transformations such as resizing, augmentation, and normalization to improve the quality of training data. This combination of hardware and software ensures that model training runs efficiently and produces optimal performance.

4. 8. Comparison with existing models

Standard DenseNet without Squeeze-and-Excitation (SE) block integration is used as a comparison model. Standard DenseNet is an architecture that connects each layer in a Dense Block directly to all subsequent layers via concatenation, enabling feature reuse and parameter efficiency. This model has a Dense Block, Transition Layer, Global Average Pooling, and Fully Connected Layer structure, with the same kernel, stride, and growth rate configurations as DenseNet-SEGR. However, the absence of SE blocks means that standard DenseNet does not have an adaptive attention mechanism for feature channels and that all channels are processed with the same weight without considering the relevance of features to the classification target. It impacts the model's performance in recognizing complex patterns, especially in datasets focusing on specific details, such as disease symptoms on tomato leaves. This study compares standard DenseNet and DenseNet-SEGR to show how adding SE blocks to the model makes it better at representing features in a more relevant way, expressed by higher accuracy, precision, recall, and F1-score.

Performance analysis was performed by comparing key evaluation metrics, such as accuracy, precision, recall, and F1-score, between DenseNet-SEGR and standard DenseNet using the same dataset to ensure a fair comparison. The results are analyzed through classification reports, confusion matrices, and learning curves to check training stability and model generalization.

5. Results of the DenseNet-SEGR model

5. 1. Data preprocessing and augmentation data

Fig. 2 shows the validation accuracy comparison between models trained without and with augmentation and preprocessing for 50 epochs.

Based on the performance comparison graph of the model with and without augmentation and pre-processing, it can be seen that the augmented approach yields better performance overall. The model using augmentation (marked in blue) shows a consistent increase in validation accuracy from the beginning of training and peaks at 99.25 % around the 38th epoch. This accuracy then stabilized until the end of training, indicating that the model was able to learn well and maintain its performance. In contrast, the model without augmentation (in orange) shows considerable accuracy fluctuations at the beginning of training and only starts to stabilize after mid-epoch, with the highest accuracy of 98.63 %, which is slightly lower than the model with augmentation.

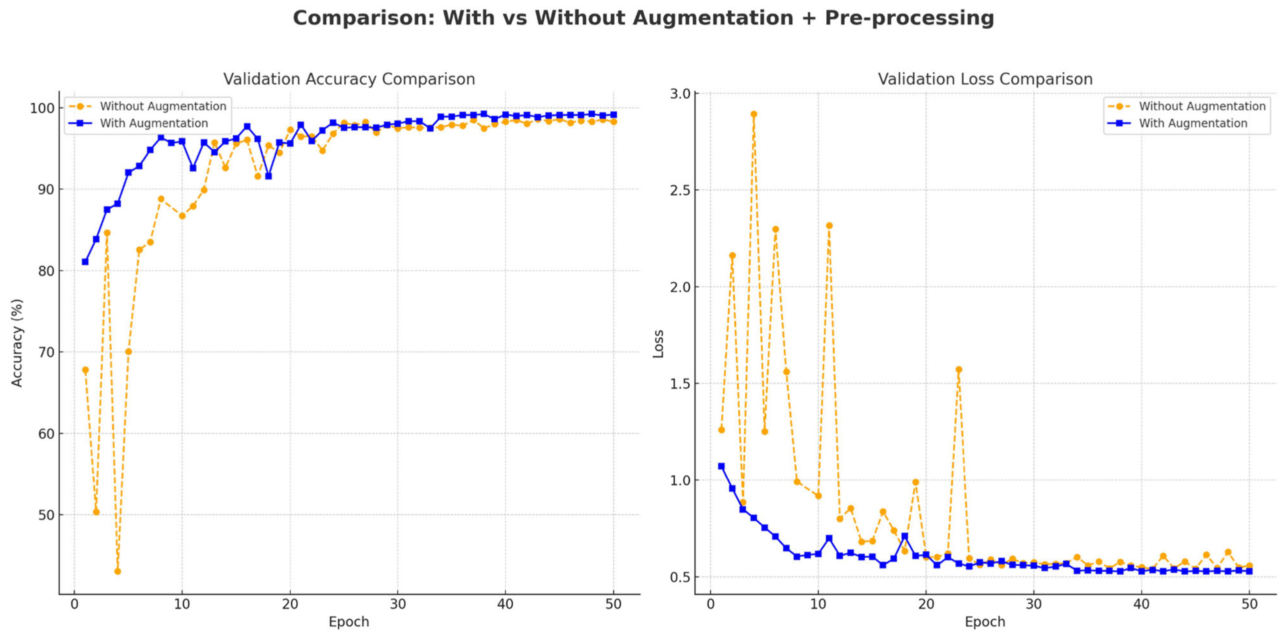

Fig. 2. Comparison: with vs without augmentation & preprocessing

Table 1 shows that the model without pre-processing and augmentation actually shows slightly superior performance compared to the model that uses both.

Table 1

Comparison: with augmentation & preprocessing

Model variant	Accuracy	Precision	Recall	F1-Score	Val Loss
With preprocessing+ +augmentation	98.22 %	98.25 %	98.22 %	98.22 %	0.5602
Without prepro-cessing+ +augmentation	98.51 %	98.53 %	98.22 %	98.40 %	0.5449

Although the model without preprocessing and augmentation achieved a slightly higher final accuracy (98.51 %) compared to the model with augmentation and preprocessing (98.22 %), the stability and generalization capability of a model are more critical than a single accuracy metric. As illustrated in the previous graph, the model without augmentation exhibited significant fluctuations in accuracy and loss during the early training stages, indicating instability and a higher risk of overfitting to the training data. In contrast, the model with augmentation and preprocessing demonstrated a more stable and consistent training trend, with progressively increasing accuracy and better-controlled loss values. Augmentation helps expand the diversity of training data artificially, while preprocessing enhances important features and reduces noise, thereby improving the model's ability to generalize to real-world, unseen data. Moreover, the 0.29 % difference in accuracy is practically negligible and not statistically significant, making the model with augmentation and preprocessing a more reliable and robust choice overall.

These experimental findings clearly demonstrate the effectiveness of data preprocessing and augmentation in improving model generalization and training stability. Despite a slightly lower final accuracy in one case, the model with preprocessing and augmentation exhibits more consistent performance across epochs.

5. 2. Modification of the Densenet architecture

DenseNet's growth rate determines the number of new features added at each layer, controlling the density of connections between layers. An optimal growth rate for tomato leaf image classification ensures efficient extraction of disease-related features, such as color and texture changes, without excessively increasing model complexity:

$$F_{output} = F_{input} + L \cdot k + \alpha \cdot L, \tag{4}$$

where F_{out} is the output feature, F_{in} is the input feature, L is the number of layers, k is the growth rate, and α is the additional influence coefficient.

A network structure consisting of four dense blocks with a growth rate of 48 and three transition layers is designed to reduce the dimensionality of the features. The network starts with an initial convolutional layer of size 7×7 with stride 2, followed by a batch normalization process (BatchNorm) and ReLU activation function. Each dense block has an increasing number of layers, namely 6, 12, 24, and 16, which enables learning of complex features. Batch normalization, ReLU, and convolution form a transition layer that reduces the feature size by a ratio of 50 %. After the last block, the network uses global average pooling to reduce the spatial dimension to 1×1 before entering the fully connected layer for class number prediction.

Table 2 shows the performance comparison between DenseNet-201, DenseNet-121, and DenseNet-SEGR on the tomato leaf disease classification task shown in Table 1. The evaluation is based on the main metrics, namely Accuracy, Precision, Recall, and F1-Score, considering the different Growth Rates and Squeeze-and-Excitation (SE) Block integration.

Results show that DenseNet-SEGR with a Growth Rate of 48 outperforms the other models, achieving the highest accuracy of 98.22 %. This indicates that modifying the architecture through increasing the growth rate and adding SE blocks increases the efficiency of feature extraction, making the model more effective in recognizing disease patterns in tomato leaves.

Table 2

Comparison of model performance based on growth rate and SE block

Model	Growth Rate	Accuracy
Densenet 201	32	0.6763
Densenet 121	32	0.9623
DenseNet-SEGR	48	0.9822

5. 3. Integration of SE blocks in DenseNet architecture

The addition of Squeeze-and-Excitation (SE) Blocks to DenseNet-SEGR is performed after each Dense Block to strengthen the relevant features in tomato leaf disease classification. SE Blocks work through three main stages: Squeeze, which summarizes feature information using Global Average Pooling (GAP); Excitation, which recalibrates feature importance through Fully Connected Layers with ReLU and Sigmoid activations; and Reweighting, which reapplies optimal weights to initial features to increase focus on key disease characteristics. The integration of SE Blocks was shown to increase accuracy, precision, recall, and F1-score, improve model generalization, and optimize feature processing making it more effective in detecting tomato leaf disease patterns.

SE Blocks improve feature selectivity by adjusting the weight of each channel using Global Average Pooling (GAP) to summarize global information, and then passing it to Fully Connected Layers with Sigmoid activation. This process allows the model to amplify more relevant features and dampen less informative features, thus improving the accuracy of tomato leaf disease classification.

Model performance evaluation was conducted using the key metrics of Accuracy, Precision, Recall, and F1-Score to measure the effectiveness of DenseNet-SEGR in tomato leaf disease classification. The model was tested on the PlantVillage dataset, which consists of 10 disease classes, and compared with DenseNet-121, DenseNet-201, and MobileNetV2.

Accuracy Measures the percentage of correct predictions of tomato leaf disease classification results from all data. Accuracy provides an idea of how well the DenseNet-SEGR model performs overall:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}, \tag{5}$$

where T_P (true positive) is the number of correct predictions for the positive class, T_N (true negative) is the number of correct predictions for the negative class, F_P (false positive) is the number of false predictions for the positive class, F_N (false negative) is the number of false predictions for the negative class.

Precision shows how many positive predictions are correct from the total positive predictions, measuring the DenseNet-SEGR model's ability to avoid false positives:

$$\text{Pr ecision} = \frac{T_P}{T_P + F_P}, \tag{6}$$

where T_P (true positive) is the number of correct predictions for the positive class, F_P (false positive) is the number of false predictions for the positive class.

Recall measures how many positive cases the model detected from the total number of positive cases. A recall is important to ensure the model can detect all diseased tomato leaf plants so that no crucial cases are missed (false negatives):

$$\text{Recall} = \frac{T_P}{T_P + F_N}, \tag{7}$$

where T_P (True Positive) is the number of correct predictions for the positive class, F_N (False Negative) is the number of false predictions for the positive class.

F1-score is used to ensure that the DenseNet-SEGR model has a good balance between the ability to detect positive cases (recall) and avoid false positives (precision):

$$F1\_\text{Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{8}$$

where Recall measures how well the model captures all positive samples, Precision measures how accurate the positive predictions made by the model are.

Fig. 3 presents the classification report for the model's performance in detecting various tomato leaf diseases. The evaluation metrics include precision, recall, and F1-score for each class, along with overall accuracy.

Fig. 4 illustrates the training and validation loss over 50 epochs. The plot demonstrates the convergence of the model, showing a steady decline in both losses, which indicates effective learning and minimal overfitting.

The graph above shows the decreasing loss trend during the training and validation of the DenseNet-SEGR model. Training and validation loss dropped significantly in the first few epochs, indicating that the model quickly learned from the data. Around the 20th epoch, both losses level off at low levels, with only a tiny difference between the training and validation losses. It shows that the model doesn't overfit and works well in real life. This consistent reduction in loss shows that the training process has succeeded in improving model performance efficiently.

Fig. 5 presents the training and validation accuracy over 50 epochs. The graph indicates a steady increase in accuracy, with both training and validation accuracy converging towards a high-performance level, suggesting effective learning with minimal overfitting.

The graph above shows the increasing training and validation accuracy trend in the DenseNet-SEGR model over 50 epochs. Accuracy increases rapidly in the first few epochs and becomes more stable after around the 20th epoch. Training accuracy reached almost 100 %, while validation accuracy remained stable above 95 % with slight fluctuation, indicating the model's good generalization ability. This performance reflects that the model can learn effectively without experiencing significant overfitting.

The Confusion Matrix displayed results from evaluating the DenseNet-SEGR model with Squeeze-and-Excitation (SE) Blocks in classifying tomato leaf diseases.

Fig. 6 presents the confusion matrix for the classification model, illustrating the performance across different tomato leaf disease categories. The matrix shows the number of correctly and incorrectly classified instances, providing insight into the model's strengths and areas for potential improvement.

Classification Report:

	precision	recall	f1-score	support
Tomato_Bacterial_spot	0.9836	0.9883	0.9859	426
Tomato_Early_blight	0.9836	0.9000	0.9399	200
Tomato_Late_blight	0.9644	0.9921	0.9781	382
Tomato_Leaf_Mold	0.9742	0.9947	0.9844	190
Tomato_Septoria_leaf_spot	0.9590	0.9915	0.9750	354
Tomato_Spider_mites_Two_spotted_spider_mite	0.9794	0.9910	0.9852	335
Tomato__Target_Spot	0.9927	0.9680	0.9802	281
Tomato__Tomato_YellowLeaf__Curl_Virus	0.9969	0.9953	0.9961	642
Tomato__Tomato_mosaic_virus	0.9867	1.0000	0.9933	74
Tomato_healthy	0.9969	0.9694	0.9829	327
accuracy			0.9822	3211
macro avg	0.9817	0.9790	0.9801	3211
weighted avg	0.9825	0.9822	0.9822	3211

Fig. 3. Evaluation metrics

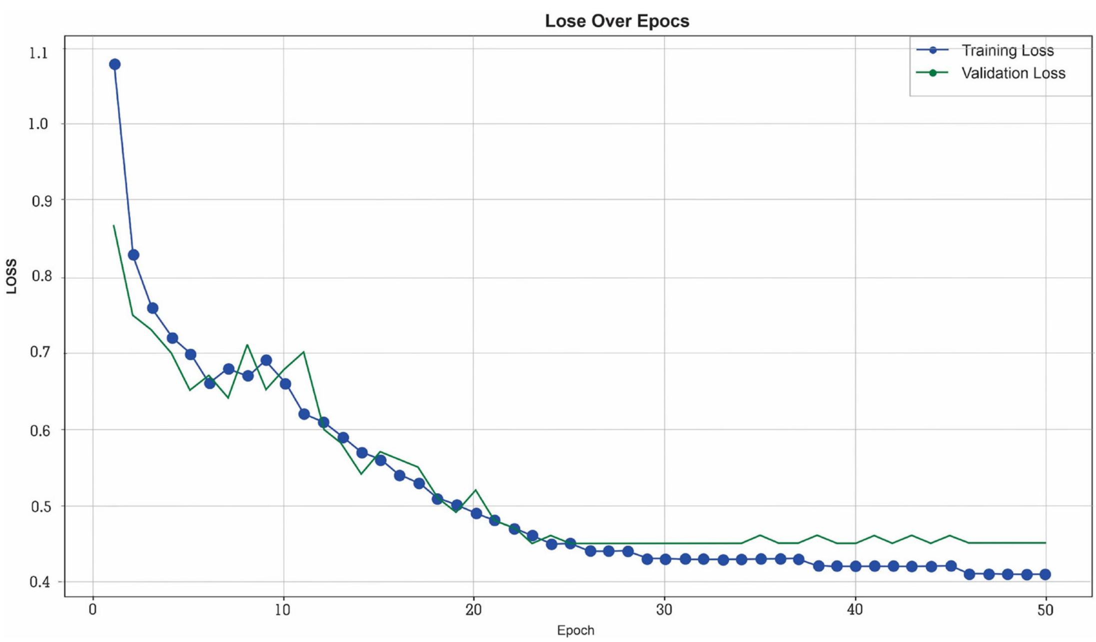

Fig. 4. Lost over epochs

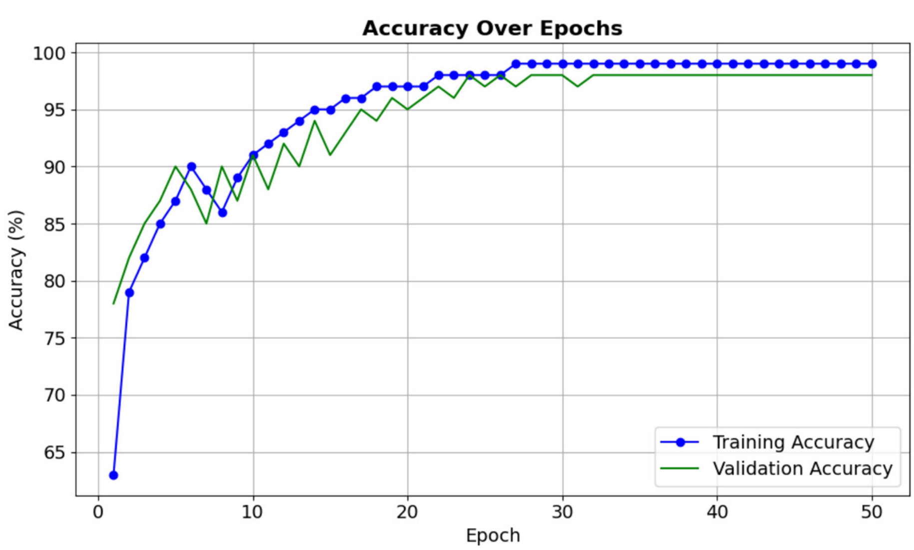

Fig. 5. Accuracy over epochs

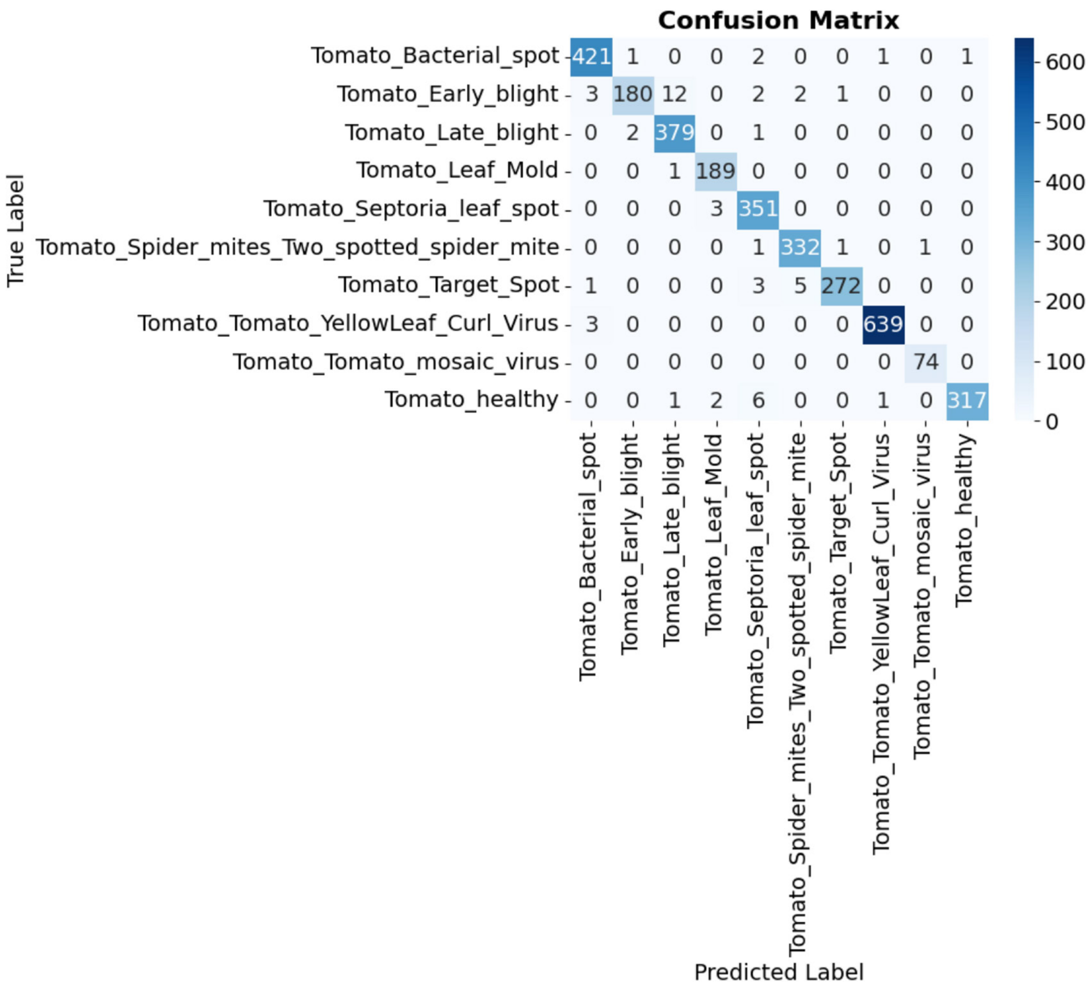

Fig. 6. Confusion matrix

This confusion matrix shows that the model has high accuracy in classifying tomato leaf diseases, with only a few minor errors in differentiating between Early Blight and Septoria Leaf Spot, as well as Late Blight and Early Blight.

Based on Table 3, which presents the comparison between the various models in terms of accuracy, precision, recall, and F1 score, it can be concluded that the DenseNet-SEGR model has the best performance in all evaluation metrics.

Table 3

Comparison of accuracy, precision, recall and F1-score

Model	Accuracy	Precision	Recall	F1 Score
Densenet 201	0.6763	0.4943	0.4985	0.4935
Densenet 121	0.9623	0.8689	0.8690	0.8688
MobileNetV2	0.9289	0.9295	0.9289	0.9287
DenseNet-SEGR	0.9822	0.9825	0.9822	0.9822

This comparison confirms the effectiveness of the DenseNet-SEGR model for improving classification accuracy compared to other models.

It is possible to see how well the DenseNet-SEGR, DenseNet201, and DenseNet121 models did at validating 50 times in the graph above. The DenseNet-SEGR model (blue line) achieves the highest validation accuracy, close to 100 %, and maintains good stability after about 20 epochs, showing superior performance in classification. DenseNet121 (green line) shows a consistent but steady increase in accuracy at a lower rate than DenseNet-SEGR, with a final accuracy of around 80 %. In contrast, DenseNet201 (orange line) performs the lowest, with validation accuracy stagnating at around 60 % after 20 epochs.

Based on Fig. 7 the graph shown, it can be seen the comparison of validation accuracy of the four models tested, namely DenseNet-SEGR, DenseNet-121, DenseNet-201, and MobileNetV2 for 50 epochs.

Fig. 7 illustrates the comparison of validation accuracy across different deep learning models: DenseNet-SEGR, DenseNet-121, DenseNet-201, and MobileNetV2 over 50 epochs. DenseNet-SEGR achieves the highest validation accuracy, stabilizing around 98–100 %, demonstrating the effectiveness of Squeeze-and-Excitation (SE) Blocks in enhancing feature selection and generalization. DenseNet-121 also performs well, reaching approximately 90–95 %, though it falls short of DenseNet-SEGR. In contrast, DenseNet-201 struggles to attain high accuracy, plateauing at 60-65 %, likely due to optimization challenges from its deeper architecture. MobileNetV2, while computationally efficient, exhibits the lowest validation accuracy at around 75–80 %, indicating its limitations in feature extraction compared to DenseNet models. Overall, these results confirm that DenseNet-SEGR is the optimal choice for plant disease classification, offering a balance of high accuracy and efficiency, making it highly suitable for AI-driven agricultural applications.

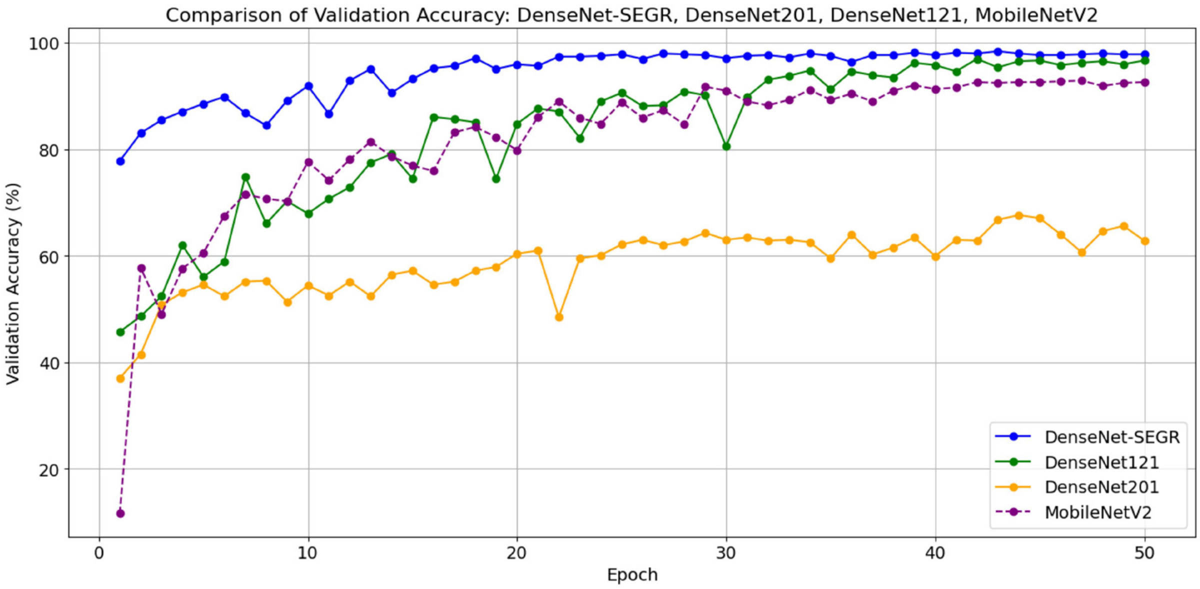

Fig. 7. Comparison of validation accuracy across DenseNet models

6. Discussion of results based on squeeze and excitation

The results of this study demonstrate that integrating the Squeeze-and-Excitation (SE) Block into the DenseNet-SEGR architecture significantly improves classification performance for tomato leaf diseases. Compared to conventional DenseNet models, the proposed architecture achieves superior accuracy (98.22 %), outperforming DenseNet-121 (96.23 %) and DenseNet-201 (67.63 %). These improvements are attributed to the enhanced feature selection capability of SE Blocks, which dynamically recalibrate feature importance across channels.

Unlike previous works such as [10], where DenseNet-201 with transfer learning was used for tomato classification but exhibited lower accuracy due to its inability to emphasize critical disease-related features, our proposed method effectively enhances feature selectivity. The integration of SE Blocks ensures that features such as color, texture, and disease-specific patterns receive higher importance, leading to better classification outcomes.

In contrast to [5], where SE Blocks were utilized in a hierarchical neural network for EEG classification, this research applies SE Blocks specifically to tomato leaf disease detection, demonstrating its adaptability to agricultural datasets. Additionally, while [19] explored SE Networks in histopathological image classification, our work extends SE integration to plant disease classification, addressing domain-specific challenges such as variations in lighting, leaf positioning, and disease symptom similarity.

The proposed DenseNet-SEGR model enhances classification performance through three key improvements: (1) Optimized Growth Rate (48), which improves feature propagation and representation compared to conventional DenseNet architectures; (2) Adaptive Feature Reweighting with SE Blocks, enabling dynamic feature selection through Global Average Pooling (GAP) and fully connected layers with ReLU and Sigmoid activation to emphasize disease-specific patterns; and (3) Robust Generalization via Data Augmentation, incorporating rotation, flipping, zooming, and brightness adjustments to mitigate overfitting and improve adaptability to diverse dataset variations.

This study effectively addresses the main limitations by (1) Addressing database imbalance through extensive data augmentation, improving generalization; (2) Improving computational efficiency, as DenseNet-SEGR remains optimized and reduces computational overload compared to deeper architectures such as DenseNet-201; and (3) Improving feature extraction, where SE Block increases sensitivity to disease-relevant features, enabling better classification for visually similar diseases [10, 23].

As shown in Table 1 and Fig. 2, the model with preprocessing and augmentation achieved an accuracy of 98.22 %, while the model without these enhancements reached 98.51 %. Although the raw accuracy and validation loss (0.5449) appear slightly better in the model without augmentation, the model with preprocessing and augmentation showed more stable validation performance across epochs, indicating better generalization capability in varied data conditions.

As shown in Table 2 and Fig. 7, the proposed DenseNet-SEGR model achieved a classification accuracy of 98.22 %, outperforming DenseNet-121 (96.23 %) and DenseNet-201 (67.63 %). This improvement can be attributed to more effective feature reuse and propagation enabled by the higher growth rate, which allows the network to capture more detailed information without excessive model complexity.

The results in Table 3 and Fig. 3 indicate that the proposed DenseNet-SEGR model achieved the highest performance across all key metrics-precision (98.25 %), recall (98.22 %), and F1-score (98.22 %) compared to baseline models. The SE blocks contributed to this improvement by recalibrating channel-wise feature importance, enabling the model to focus more effectively on regions of the leaf affected by disease.

In summary, the discussion confirms that each component of the proposed DenseNet-SEGR model-namely preprocessing and augmentation, architectural modification through increased growth rate, and integration of SE blocks-contributes to improved model performance, stability, and generalization. These outcomes support the overall goal of building an accurate and practical tomato leaf disease classification system suitable for deployment in real-world agricultural environments.

Despite its strengths, this research has several limitations, including dependence on datasets, high computational requirements, and lack of real-world testing on agricultural devices. Future improvements could focus on (1) model optimization to reduce complexity without sacrificing accuracy, (2) domain adaptation for better generalization under real-world conditions, (3) integration of edge computing for real-time disease detection via mobile devices or drones, and (4) hybrid approaches, such as combining DenseNet-SEGR with transformer models to improve feature extraction.

The integration of Squeeze-and-Excitation Blocks into the DenseNet-SEGR architecture has proven to be highly effective for plant disease classification. The improvements in feature selection, growth rate optimization, and data augmentation techniques contribute to superior accuracy and robustness. Future research directions should focus on real-world deployment, model compression, and cross-domain adaptation to maximize the practical impact of this approach in precision agriculture.

7. Conclusions

1. The model with preprocessing and augmentation achieved an accuracy of 98.22 %, slightly lower than the 98.51 % of the model without these steps. However, the augmented model demonstrated more stable validation trends and better generalization across epochs.

2. Modifying the DenseNet architecture by increasing the growth rate to 48 has proven effective in enhancing the model's ability to propagate and diversify features across layers, which in turn leads to improved classification performance. The proposed DenseNet-SEGR model achieved an accuracy of 98.22 %, surpassing DenseNet-121 (96.23 %) and DenseNet-201 (67.63 %). This performance gain can be attributed to the richer feature representations made possible by the increased growth rate, allowing the network to capture more detailed and discriminative patterns without significantly increasing computational complexity. Compared to conventional DenseNet variants, this architectural modification addresses key limitations, such as feature redundancy and insufficient deep feature learning.

3. The integration of Squeeze-and-Excitation (SE) blocks after each Dense Block has proven effective in enhancing the model's sensitivity to disease-relevant features by adaptively recalibrating the importance of each feature channel. As shown in Fig. 3 and Table 3, the SE-enhanced model outperformed baseline architectures—namely DenseNet-121, DenseNet-201, and MobileNetV2 – by achieving a precision of 98.25 %, a recall of 98.22 %, and an F1-score of 98.22 %. These improvements are primarily attributed to the SE blocks' capability to emphasize features associated with disease symptoms while suppressing irrelevant or redundant information. In contrast to conventional models that process all feature channels uniformly, the SE mechanism enables the network to focus on the most informative patterns, thereby improving classification accuracy. This result highlights the advantage of incorporating SE blocks into the architecture, particularly in enhancing attention to critical features for more reliable disease classification.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has data included as electronic supplementary material.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Bakr, M., Abdel-Gaber, S., Nasr, M., Hazman, M. (2022). Tomato disease detection model based on densenet and transfer learning. Applied Computer Science, 18 (2), 56–70. https://doi.org/10.35784/acs-2022-13
2. Ahmed, S., Hasan, Md. B., Ahmed, T., Sony, Md. R. K., Kabir, Md. H. (2022). Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification. IEEE Access, 10, 68868–68884. https://doi.org/10.1109/access.2022.3187203
3. Khan, I., Sohail, S. S., Madsen, D. Ø., Khare, B. K. (2024). Deep transfer learning for fine-grained maize leaf disease classification. Journal of Agriculture and Food Research, 16, 101148. https://doi.org/10.1016/j.jafr.2024.101148
4. Arulananth, T. S., Prakash, S. W., Ayyasamy, R. K., Kavitha, V. P., Kuppusamy, P. G., Chinnasamy, P. (2024). Classification of Paediatric Pneumonia Using Modified DenseNet-121 Deep-Learning Model. IEEE Access, 12, 35716–35727. https://doi.org/10.1109/access.2024.3371151
5. Zhu, H., Wang, L., Shen, N., Wu, Y., Feng, S., Xu, Y. et al. (2023). MS-HNN: Multi-Scale Hierarchical Neural Network With Squeeze and Excitation Block for Neonatal Sleep Staging Using a Single-Channel EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 31, 2195–2204. https://doi.org/10.1109/tnsre.2023.3266876
6. Yuan, Z., Li, X., Hao, Z., Tang, Z., Yao, X., Wu, T. (2024). Intelligent prediction of Alzheimer's disease via improved multifeature squeeze-and-excitation-dilated residual network. Scientific Reports, 14 (1). https://doi.org/10.1038/s41598-024-62712-w
7. Zhu, C., Wang, L., Zhao, W., Lian, H. (2024). Image classification based on tensor network DenseNet model. Applied Intelligence, 54 (8), 6624–6636. https://doi.org/10.1007/s10489-024-05472-4

8. Cui, X., Yu, Z., Li, J., Jiang, B., Li, S., Liu, J. (2023). Research on RFID Localization Algorithm of Refrigerator Foods Based on Improved DenseNet Model. IEEE Networking Letters, 5 (2), 135–139. https://doi.org/10.1109/lnet.2023.3258458
9. Mugume, E., Tumwesigye, A., Muhangi, A. (2021). A spatio-temporal sleep mode approach to improve energy efficiency in small cell DenseNets. SAIEE Africa Research Journal, 112 (3), 134–141. https://doi.org/10.23919/saiee.2021.9513627
10. Lu, T., Han, B., Chen, L., Yu, F., Xue, C. (2021). A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning. Scientific Reports, 11 (1). https://doi.org/10.1038/s41598-021-95218-w
11. Asker, M. E., Güngör, M. (2024). A hybrid approach consisting of 3D depthwise separable convolution and depthwise squeeze-and-excitation network for hyperspectral image classification. Earth Science Informatics, 17 (6), 5795–5821. https://doi.org/10.1007/s12145-024-01469-2
12. Roy, K., Chaudhuri, S. S., Frnda, J., Bandopadhyay, S., Ray, I. J., Banerjee, S., Nedoma, J. (2023). Detection of Tomato Leaf Diseases for Agro-Based Industries Using Novel PCA DeepNet. IEEE Access, 11, 14983–15001. https://doi.org/10.1109/access.2023.3244499
13. Zhang, Y., Zhang, X., Li, Z., Li, X., Wang, Z. (2024). Hierarchical intelligent lithology recognition for thin section images using enhanced DenseNet. Earth Science Informatics, 18 (1). https://doi.org/10.1007/s12145-024-01663-2
14. Bouni, M., Hssina, B., Douzi, K., Douzi, S. (2024). Synergistic use of handcrafted and deep learning features for tomato leaf disease classification. Scientific Reports, 14 (1). https://doi.org/10.1038/s41598-024-71225-5
15. Umar, M., Altaf, S., Ahmad, S., Mahmoud, H., Mohamed, A. S. N., Ayub, R. (2024). Precision Agriculture Through Deep Learning: Tomato Plant Multiple Diseases Recognition With CNN and Improved YOLOv7. IEEE Access, 12, 49167–49183. https://doi.org/10.1109/access.2024.3383154
16. Cheng, Y.-H., Kuo, C.-N., Lin, Y.-D. (2024). An Artificial IoT-Enabled Smart Production Line for 360° Visual Defect Detection and Classification of Cherry Tomatoes. IEEE Internet of Things Journal, 11 (10), 18082–18098. https://doi.org/10.1109/jiot.2024.3360715
17. Liu, H., Liu, S., Wen, C., Wong, W. E. (2022). TBEM: Testing-Based GPU-Memory Consumption Estimation for Deep Learning. IEEE Access, 10, 39674–39680. https://doi.org/10.1109/access.2022.3164510
18. Mao, Y., Kim, J., Podina, L., Kohandel, M. (2025). Dilated SE-DenseNet for brain tumor MRI classification. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-86752-y
19. Devassy, B. R., Antony, J. K. (2023). Histopathological image classification using CNN with squeeze and excitation networks based on hybrid squeezing. Signal, Image and Video Processing, 17 (7), 3613–3621. https://doi.org/10.1007/s11760-023-02587-y
20. Al-Waeli, M., Shahmohammadi, N., Tavakoli, S., Dizadji, A., Kvarnheden, A. (2024). Infection of tomato in Iraq with tomato leaf curl Palampur virus and multiple variants of tomato yellow leaf curl virus. Journal of Plant Pathology, 106 (3), 1283–1294. https://doi.org/10.1007/s42161-024-01682-4
21. Romero-Rodríguez, B., Petek, M., Jiao, C., Križnik, M., Zagorščak, M., Fei, Z. et al. (2023). Transcriptional and epigenetic changes during tomato yellow leaf curl virus infection in tomato. BMC Plant Biology, 23 (1). https://doi.org/10.1186/s12870-023-04534-y
22. Khan, M., Hossni, Y. (2025). A comparative analysis of LSTM models aided with attention and squeeze and excitation blocks for activity recognition. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-88378-6
23. Zhou, L.-Q., Sun, P., Li, D., Piao, J.-C. (2022). A Novel Object Detection Method in City Aerial Image Based on Deformable Convolutional Networks. IEEE Access, 10, 31455–31465. https://doi.org/10.1109/access.2022.3156953
24. Xu, J., Pan, S., Sun, P. Z. H., Hyeong Park, S., Guo, K. (2023). Human-Factors-in-Driving-Loop: Driver Identification and Verification via a Deep Learning Approach using Psychological Behavioral Data. IEEE Transactions on Intelligent Transportation Systems, 24 (3), 3383–3394. https://doi.org/10.1109/tits.2022.3225782
25. Sun, M., Yu, M., Lv, P., Li, A., Wang, H., Zhang, X. et al. (2021). Man-Made Threat Event Recognition Based on Distributed Optical Fiber Vibration Sensing and SE-WaveNet. IEEE Transactions on Instrumentation and Measurement, 70, 1–11. https://doi.org/10.1109/tim.2021.3081178
26. Al Jbaar, M. A., Dawwd, S. A. (2023). SIMD implementation of deep CNNs for myopia detection on a single-board computer system. Eastern-European Journal of Enterprise Technologies, 5 (9 (125)), 98–108. https://doi.org/10.15587/1729-4061.2023.289007