

Original Article

# Equilibrium Optimizer with Deep Convolutional Neural Network-based SqueezeNet Model for Grape Leaf Disease Classification in IoT Environment

P. Sindhu<sup>1</sup>, G. Indirani<sup>2</sup>

<sup>1</sup>Department of Computer Science, Mahatma Gandhi Govt. Arts College, Chalakkara, P.O New Mahe, Puducherry(UT), India

<sup>2</sup>Department of Computer Science & Engineering, Govt. College of Engineering, Sengipatti, Thanjavur, Tamil Nadu, India.

<sup>1</sup>psindumano1@gmail.com

Received: 04 February 2022

Revised: 17 March 2022

Accepted: 16 April 2022

Published: 19 May 2022

**Abstract** - Internet of Things (IoT) plays a vital role in enhancing crop quality and productivity in the agricultural sector. Accurate and earlier detection of grape leaf diseases is important to control the spreading of diseases and safeguard the healthier growth of grape productivity. Since the traditional way of visual inspection is a difficult and laborious process, automated tools using computer vision and artificial intelligence (AI) approaches are essential. At the same time, the effective selection of hyperparameter values results in improved classification results. This study introduces a novel Equilibrium Optimizer with a Deep Convolutional Neural Network-based SqueezeNet Model for Grape Leaf Disease Classification (EOSN-GLDC) model in an IoT environment. The proposed EOSN-GLDC model focuses on recognizing and classifying grape leaf diseases. The presented EOSN-GLDC model initially employs the median filtering (MF) approach to remove noise. Followed by the EO algorithm with the SqueezeNet model is utilized as a feature extractor where the hyperparameters involved are adjusted by utilizing the EO algorithm. Moreover, an extreme learning machine (ELM) classifier is applied for allocating proper class labels to the input images. To demonstrate the improved performance of the EOSN-GLDC model, a comprehensive experimental analysis is made using a benchmark dataset, and the results indicate the betterment of the EOSN-GLDC model.

**Keywords** - Computer vision, Deep learning, Grape leaf diseases, Metaheuristics, Plant diseases.

## 1. Introduction

In agriculture, the Internet of Things (IoT) permits gadgets across a ranch to gauge a wide range of information from a distance and give this data to the landowner continuously [1]. IOT gadgets can accumulate data like soil dampness, fertilizer application, dam levels, domesticated animals' wellbeing, and screen wall vehicles and weather. Indian Economy is profoundly subject to the agrarian efficiency of the country. Grape is an exceptionally business product of India. It can undoubtedly be filled in tropical, sub-tropical, and calm climatic districts. India has various kinds of environment and soil in various pieces of the country [2]. This makes grapevines a significant vegetative proliferated crop with high financial significance. The grape plant will cause unfortunate yield and development when impacted by diseases. The diseases are expected to the viral, microscopic organisms, and growths contaminations brought about by insects, rust and nematodes, and so forth [3, 4]. The ranchers decide on these diseases through their experience or with the assistance of specialists through unaided eye perception,

which isn't a precise and tedious interaction [5]. Fig. 1 demonstrates the process of IoT in agriculture.

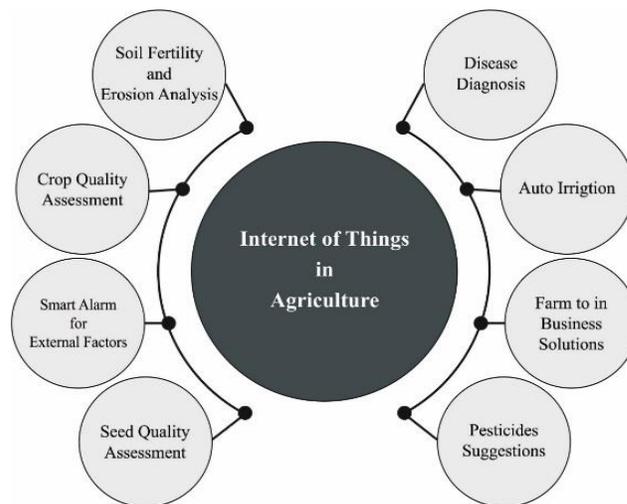


Fig. 1 Role of IoT in agriculture



Early discovery of disease is then a lot of required in the agribusiness and agriculture field to build the yield of the harvests [6]. A framework is proposed that can recognize and distinguish diseases in the leaves of the grape plants. Customarily, disease identification was done physically with the assistance of human specialists. For the most part, this approach is tedious, less practical, and more inclined to blunder [7, 8]. With the development of shrewd agribusiness and cultivating, more creative arrangements are expected to help before the location and grouping of assorted grape diseases and peats and the exact spreading of pesticides. Ongoing advancements in computerized reasoning and PC vision have tracked down different applications in savvy farming and grapevine [9]. There is a solid interest in wise disease discovery and grouping in future savvy agribusiness [10]. Computerized reasoning methods, particularly deep learning, empower the natural location and order of assorted grape diseases and vermin with a higher exactness rate.

Liu et al. [11] proposed an effective grape leaf disease classification model using an improved CNN model. The Inception network is utilized to enhance the results of the feature extractor, and a dense connectivity policy is presented to reinforce the exhibition of complex feature extraction. In [12], the mixture of many CNNs permits the projected model to extract extra complex features. The delegate capability of the United Model has, in this way, been upgraded. The United Model was assessed on the Plant Village dataset and contrasted with other best-in-class CNN models. In [13], a grape leaf disease detection network (GLDDN) is presented, using a dual attention mechanism to evaluate, detect, and classify features. The experimental process is carried out during the validation process using a benchmark dataset to confirm the GLDDN model is effectively suitable compared to other models as it detects the affected areas. In [14], a united CNN model depending upon the DL concepts is presented. The presented CNN model can differentiate leaves with general grape leaf illnesses. The inclusion of many CNN allows the presented model to extract distinct features.

Here a novel Equilibrium Optimizer with Deep Convolutional Neural Network is used based on SqueezeNet Model for Grape Leaf Disease Classification (EOSN-GLDC) model in an IoT environment. This model employs the median filtering (MF) approach to remove noise. The EO algorithm with the SqueezeNet model is utilized as a feature extractor where the hyperparameters are adjusted by utilizing of EO approach. Moreover, an extreme learning machine (ELM) classifier is applied for allocating proper class labels to the input images. To demonstrate the improved performance of the EOSN-GLDC technique, a comprehensive experimental analysis is made using a benchmark dataset, and the outcomes indicate the betterment of the EOSN-GLDC model.

## 2. The Proposed Model

In this study, a new EOSN-GLDC technique was developed to detect and classify grape leaf diseases in an IoT environment. The presented EOSN-GLDC model exploited the SqueezeNet model to extract the features where its hyperparameters are adjusted. Finally, an ELM classifier is applied to allocate proper class labels to the input images.

### 2.1. Feature Extraction using SqueezeNet

Once the input image is pre-processed, the SqueezeNet model is exploited to produce deep features. SqueezeNet is a convolution network that executes optimal efficacy than AlexNet with 50x lesser parameters [15]. It contains 15 layers with 5 various layers as 8 fire layers, 1 output layer softmax, 2 convolution layers, 3 max-pooling layers, and 1 global average pooling layer. In  $K \times K$ , notations suggest the receptive field size of filters, 's' refers to the stride size, and l denotes the length of feature maps (FM), respectively. An input of network contains  $227 \times 227$  dimensional with RGB channels. Fig. 2 illustrates the framework of the SqueezeNet Model.

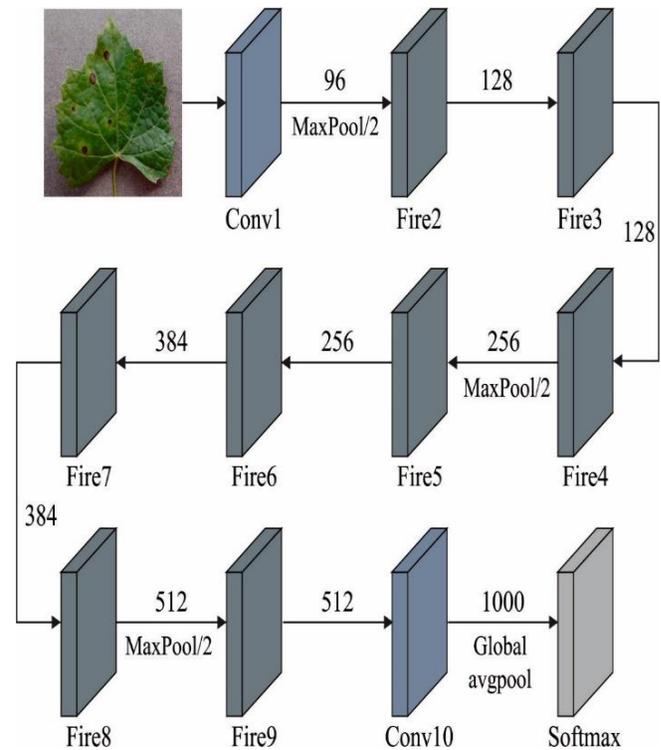


Fig. 2 Architecture of SqueezeNet Model

The generalized input image with convolution as well as max-pooling layers is implemented. The convolution layer obtained convoluted among weight and lesser region from the input volume, with  $3 \times 3$  kernels. Every convolution layer applies an elementwise activation drive as a positive part of its argument. The resultant tensor scale and input of fires are dependent upon everyone. The squeeze phase employs the filtering size of  $1 \times 1$ . However, the progress

employs the filtering size  $1 \times 1$  and  $3 \times 3$ . Primarily, an input tensor  $H \times W \times C$  allows the squeeze, and the count of convolution layers corresponds to  $C/4$  of the amount of input tensor channels.

Finally, the  $f\{y\}$  of squeeze function with kernel  $w$  is expressed as:

$$f\{y\} = \sum_{f=1}^{FM} \sum_{c=1}^c w_c^f x_c^{fm1} \quad (1)$$

### 2.2. EO based Hyperparameter Optimization

Then, the EO algorithm is applied to fine-tune the hyperparameters involved in the SqueezeNet model. EO studies an active mass balance procedure that controls volume. The arbitrary population was initialized by the standard distribution depending on the dimensional and amount of particles as follows [16, 17]:

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min})i = 1, 2, \dots, n \quad (2)$$

Whereas  $C_{min}$  and  $C_{max}$  indicates the lower and upper limits,  $C_i^{initial}$  characterizes the vector  $i$ th particle  $n$  describes the size of the population, and  $rand_i$  indicates the arbitrary value within  $[0,1]$ . The exponential term  $F$  demonstrates that it assists EO in attaining the suitable balance between intensification and diversification.  $\lambda$  is an arbitrary number within  $[0,1]$  to control the turn-over speed from the real control volume.

$$\vec{F} = e^{-\lambda(t-t_0)} \quad (3)$$

In which  $t$  is represented by the function of the quantity of iteration ( $Iter$ ) as follows:

$$t = \left(1 - \frac{Iter}{Max\_iter}\right) \left(a_2 \frac{Iter}{Max\_iter}\right) \quad (4)$$

Now,  $Iter = current\ iteration$ ,  $Max\_iter = maximal\ iteration$ , and variable  $a_2$  were applied to control the exploitation ability of  $EO$ . To ensure convergence and improve local along with the global searching ability of method, expressed in the following:

$$\vec{t}_0 = \frac{1}{\lambda} \ln \left( -a_1 \text{sign}(\vec{r} - 0.5) \left[ 1 - e^{-\lambda t} \right] \right) + t \quad (5)$$

Whereas  $a_1$  and  $a_2$  are employed to manage local along with the global searching ability of the EO method. The term

$\text{sign}(\vec{r} - 0.5)$  is accountable near the explorations route and exploitations. In  $EO$ , the  $a_1$  and  $a_2$  values are 2 and 1. By replacing Eq. (5) in Eq. (3), the term is transformed into:

$$\vec{F} = a_1 \text{sign}(\vec{r} - 0.5) \left[ e^{-\lambda t} - 1 \right] \quad (6)$$

The generation rate in the EO method was employed to improve exploitation, i.e., applied to the function of time. The 1<sup>st</sup> order exponential decay process from the technique of generation rate of the multi-purpose method is defined by the following equation:

$$\vec{G} = \vec{G}_0 e^{-k(t-t_0)} \quad (7)$$

In which  $G_0 = primary\ value$ ,  $k = decay\ variables$ .

At last, the generation rate appearance considered  $k = \lambda$  was defined in the following:

$$\vec{G} = \vec{G}_0 e^{-\lambda(t-t_0)} = \vec{G}_0 \vec{F}_0 \quad (8)$$

Here,  $G_0$  is evaluated by the following equation:

$$\vec{G}_0 = G \vec{C} P (\vec{C}_{eq} - \lambda \vec{C}) \quad (9)$$

$$G \vec{C} P = \begin{cases} 0.5r_1, & r_2 \geq 0 \\ 0, & r_2 < 0 \end{cases} \quad (10)$$

In which  $r_1$  and  $r_2$  represent the random number within  $[0,1]$ ,  $GCP$  represents the variable control generation rate. With the formula mentioned above, the upgrade equation of concentration is described by the following:

$$\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}) \vec{F} + \frac{\vec{G}}{\lambda V} (1 - \vec{F}) \quad (11)$$

The upgraded formula comprises: the initial term is equilibrium concentration; the next term is applied for global searching. The last term is answerable for local searching to achieve an accurate solution.

### 2.3. ELM based classification

In the final stage, the ELM model was utilized to assign proper class labels [18]. ELM is developed to prevent the iteration training process and increase the performance generality. The ELM encompasses  $n$ ,  $l$ , and  $m$  input, hidden, and  $m$  output layer neurons. Initially, consider the trained instance  $\{X, Y\} = \{x_i, y_i\}$  ( $i = 1, 2, \dots, Q$ ), and input feature  $X = [x_{i1} x_{i2} \dots x_{iQ}]$  and outcome matrix  $y = [y_{j1} y_{j2} \dots y_{jQ}]$  comprises the trained samples, while the matrices  $X$  and  $Y$  can be determined as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1Q} \\ x_{21} & x_{22} & \cdots & x_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nQ} \end{bmatrix}, \quad (12)$$

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{mQ} \\ y_{21} & y_{22} & \cdots & y_{mQ} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mQ} \end{bmatrix},$$

The parameters  $n$  and  $m$  represent the dimension of the input and output matrices. Then, the ELM fixed the weight among the hidden and input layers.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{l1} & w_{l2} & \cdots & w_{ln} \end{bmatrix}, \quad (13)$$

Now  $w_{ij}$  signifies the weights among the  $j$ th and  $i$ th input and hidden layers. Next, the ELM considered the weights amongst the hidden and output layers as follows

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{lm} \end{bmatrix}, \quad (14)$$

Here  $\beta_{jk}$  indicates the weights amongst the  $j$ th and  $k$ th hidden and output layers. Subsequently, the ELM executes the bias of hidden layers as follows:

$$B = [b_1 \ b_2 \ \dots \ b_n]^T \quad (15)$$

Next, the ELM chooses the network activation function  $g(x)$ , and the resulting matrix  $T$  is associated with the following.

$$T = [t_1, t_2, \dots, t_Q]_{m \times Q}. \quad (16)$$

The column vector of the resulting matrix  $T$  is associated by the following:

$$t_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{mj} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^l \beta_{i1} g(w_i x_j + b_i) \\ \sum_{i=1}^l \beta_{i2} g(w_i x_j + b_i) \\ \vdots \\ \sum_{i=1}^l \beta_{im} g(w_i x_j + b_i) \end{bmatrix} \quad (j = 1, 2, 3, \dots, Q). \quad (17)$$

Considering the above two equations, it turns out to be

$$H\beta = T', \quad (18)$$

Now  $T'$  denotes the transposition of  $T$  and  $H$  represents the outcomes of the hidden state.

### 3. Performance Validation

This section presents a comprehensive simulation analysis of the proposed model using a benchmark Plant village dataset from the Kaggle repository [19]. The dataset encompasses four classes of black rot (BR), black measles (BMEA), leaf blight (LB), and normal, with 300 images each. A few sample images are illustrated in Fig. 3.

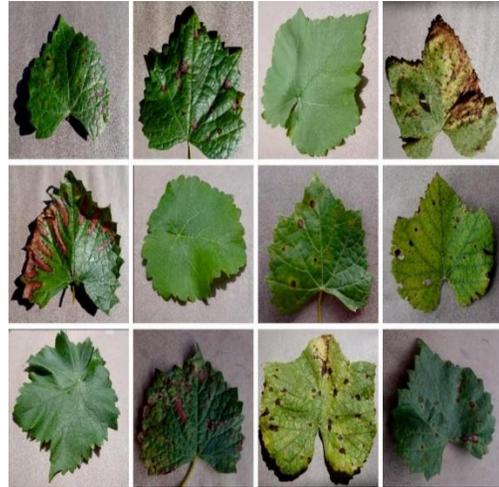


Fig. 3 Sample Images

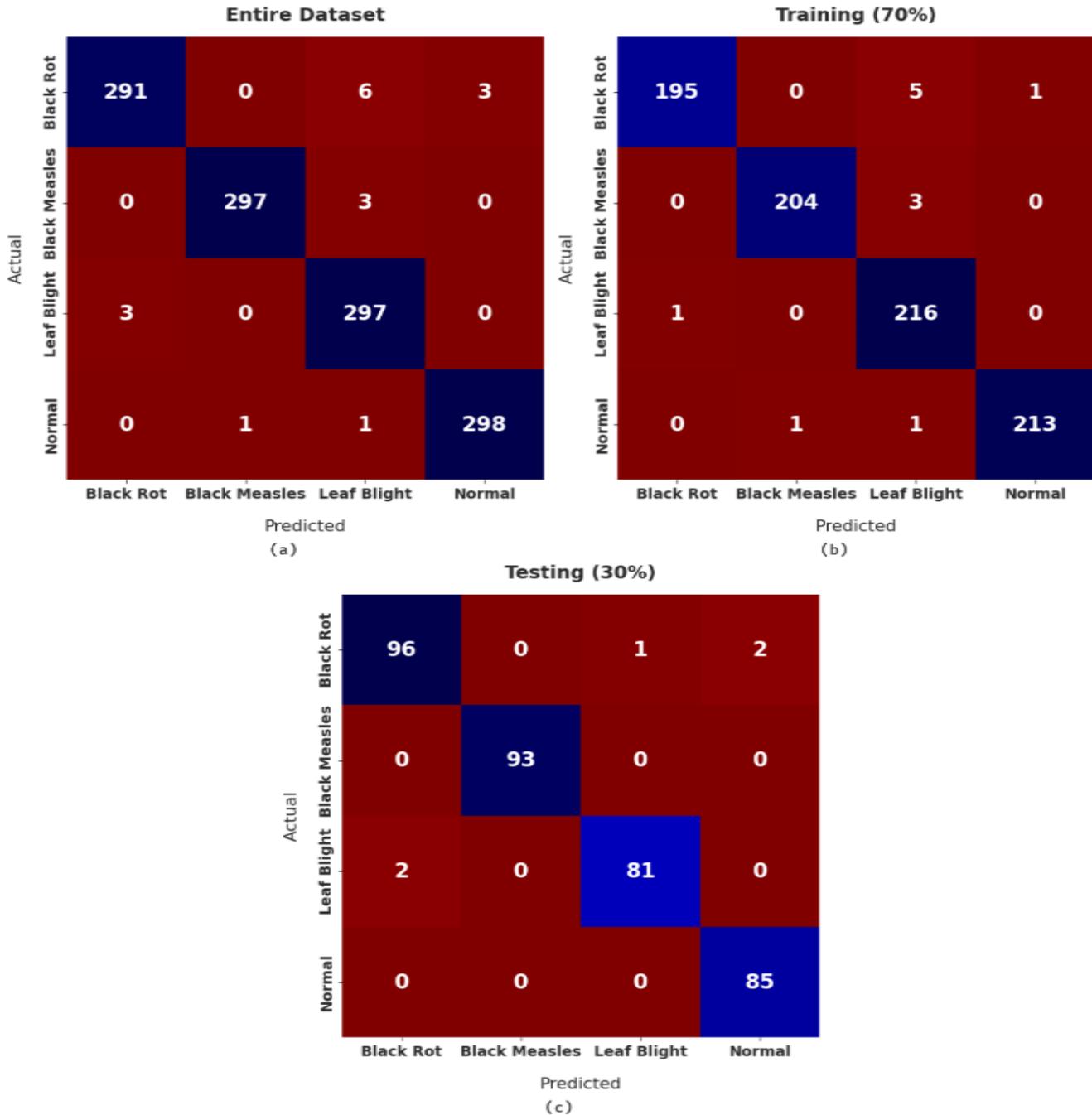


Fig. 4 Confusion matrix of EOSN-GLDC technique

Fig. 4 implies a set of three confusion matrices generated by the EOSN-GLDC model on the test dataset. On the entire dataset, the EOSN-GLDC model has recognized 291 images under BR, 297 images under BMEA, 297 images under LB, and 298 images under Normal. Following 70% of the training dataset, the EOSN-GLDC approach has recognized 195 images under BR, 204 images under BMEA, 216 images under LB, and 213 images under Normal. In addition, on the 30% of the testing dataset, the EOSN-GLDC methodology has recognized 96 images under BR, 93 images

under BMEA, 81 images under LB, and 85 images under Normal.

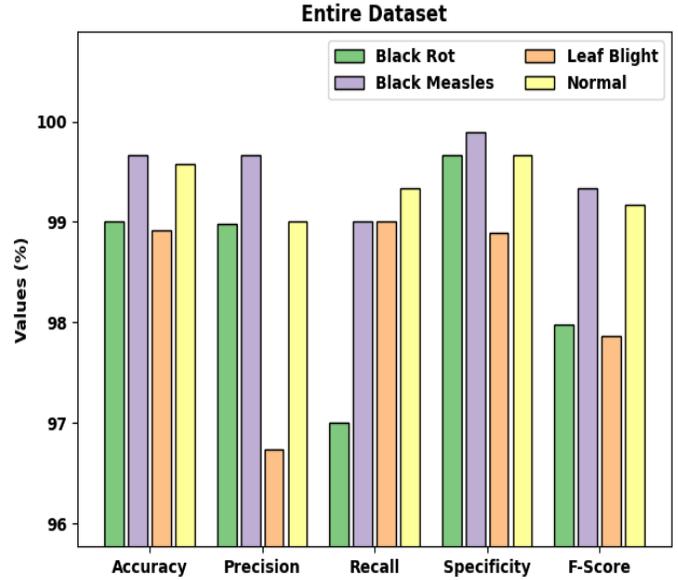
Table 1 exhibits detailed classifier results of the EOSN-GLDC model on the test dataset under distinct class labels. The experimental values in the table implied that the EOSN-GLDC model had resulted in ineffectual outcomes under every class.

**Table 1. Result analysis of EOSN-GLDC technique with distinct measures and datasets**

Class Labels	Accuracy	Precision	Recall	Specificity	F-Score
<b>Entire Dataset</b>					
Black Rot	99.00	98.98	97.00	99.67	97.98
Black Measles	99.67	99.66	99.00	99.89	99.33
Leaf Blight	98.92	96.74	99.00	98.89	97.86
Normal	99.58	99.00	99.33	99.67	99.17
<b>Average</b>	<b>99.29</b>	<b>98.60</b>	<b>98.58</b>	<b>99.53</b>	<b>98.58</b>
<b>Training (70%)</b>					
Black Rot	99.17	99.49	97.01	99.84	98.24
Black Measles	99.52	99.51	98.55	99.84	99.03
Leaf Blight	98.81	96.00	99.54	98.56	97.74
Normal	99.64	99.53	99.07	99.84	99.30
<b>Average</b>	<b>99.29</b>	<b>98.63</b>	<b>98.54</b>	<b>99.52</b>	<b>98.58</b>
<b>Testing (30%)</b>					
Black Rot	98.61	97.96	96.97	99.23	97.46
Black Measles	100.00	100.00	100.00	100.00	100.00
Leaf Blight	99.17	98.78	97.59	99.64	98.18
Normal	99.44	97.70	100.00	99.27	98.84
<b>Average</b>	<b>99.31</b>	<b>98.61</b>	<b>98.64</b>	<b>99.54</b>	<b>98.62</b>

Fig. 5 inspects comprehensive grape leaf disease detection outcomes of the EOSN-GLDC model on the entire dataset. The proposed EOSN-GLDC model has gained effective performance under each class. For instance, the proposed EOSN-GLDC model has identified BR class with  $accu_{racy}$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 99.00%, 98.98%, 97%, 99.67%, and 97.98%, respectively. The proposed EOSN-GLDC algorithm has identified BMEA class with  $accu_{racy}$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 99.67%, 99.66%, 99%, 99.89%, and 99.33%, correspondingly. Moreover, the proposed EOSN-GLDC

technique has identified the LB class with  $accu_{racy}$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 98.92%, 96.74%, 99%, 98.89%, and 97.86%, respectively. Eventually, the proposed EOSN-GLDC system has identified Normal classes with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$ s of 99.58%, 99%, 99.33%, 99.67%, and 99.17%, correspondingly.



**Fig. 5 Result analysis of the EOSN-GLDC technique on the entire dataset**

Fig. 6 demonstrates comprehensive grape leaf disease detection outcomes of the EOSN-GLDC technique on 70% of the training dataset. The proposed EOSN-GLDC approach has gained effective performance under each class. For instance, the proposed EOSN-GLDC model has identified BR class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 99.17%, 99.49%, 97.01%, 99.84%, and 98.24% correspondingly. Likewise, the proposed EOSN-GLDC technique has identified BMEA class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 99.52%, 99.51%, 98.55%, 99.84%, and 99.03%, correspondingly. Additionally, the proposed EOSN-GLDC model has identified the LB class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$ s of 98.81%, 96%, 99.54%, 98.56%, and 97.74% correspondingly. At last, the proposed EOSN-GLDC approach has identified the Normal class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 99.64%, 99.53%, 99.07%, 99.84%, and 99.30%, respectively.

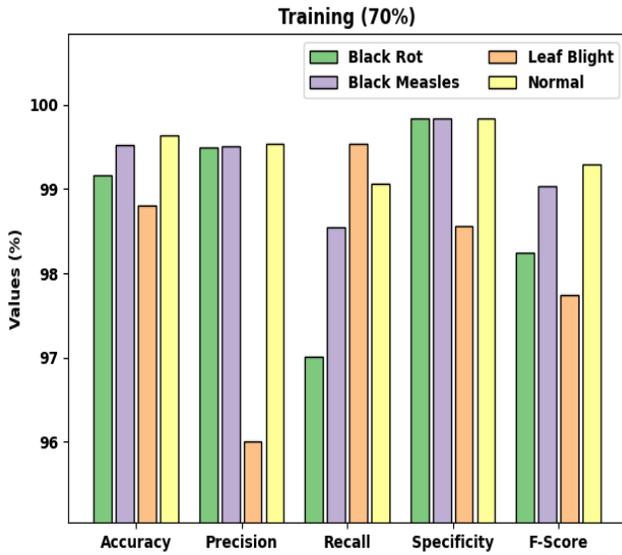


Fig. 6 Result analysis of ESN-GLDC technique on 70% of the training dataset

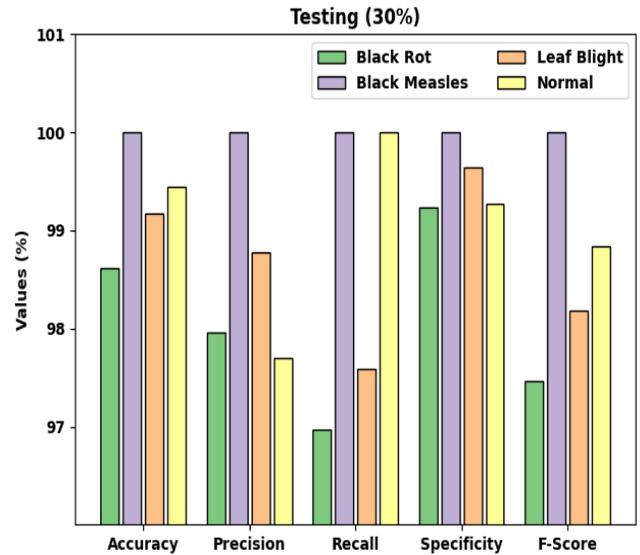


Fig. 7 Result analysis of ESN-GLDC technique on 30% testing dataset

Fig. 7 examines comprehensive grape leaf disease detection outcomes of the ESN-GLDC model on 30% of the testing dataset. The proposed ESN-GLDC model has gained effective performance under each class. For instance, the proposed ESN-GLDC model has identified BR class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 98.61%, 97.96%, 96.97%, 99.23%, and 97.46% correspondingly. Also, the proposed ESN-GLDC algorithm has identified BMEA class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$  of 100%, 100%, 100%, 100%, and 100% correspondingly. Furthermore, the proposed ESN-GLDC method has identified the LB class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$ s of 99.17%, 98.78%, 97.59%, 99.64%, and 98.18%, respectively. Finally, the proposed ESN-GLDC approach has identified the Normal class with  $accu_y$ ,  $prec_n$ ,  $reca_l$ ,  $spec_y$ , and  $F_{-score}$ s of 99.44%, 97.70%, 100%, 99.27%, and 98.84%, correspondingly.

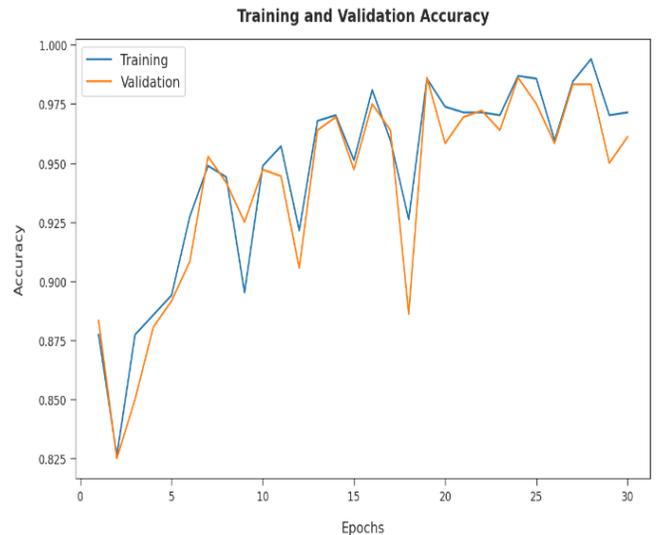


Fig. 8 Accuracy graph analysis of ESN-GLDC technique

Fig. 8 illustrates the training and validation accuracy inspection of the ESN-GLDC approach on the applied dataset. The figure conveyed that the ESN-GLDC model has offered maximum training/validation accuracy in the classification process.

Next, Fig. 9 exemplifies the training and validation loss inspection of the ESN-GLDC model on the applied dataset. The figure revealed that the ESN-GLDC model had offered reduced training/accuracy loss in the classification process of test data.

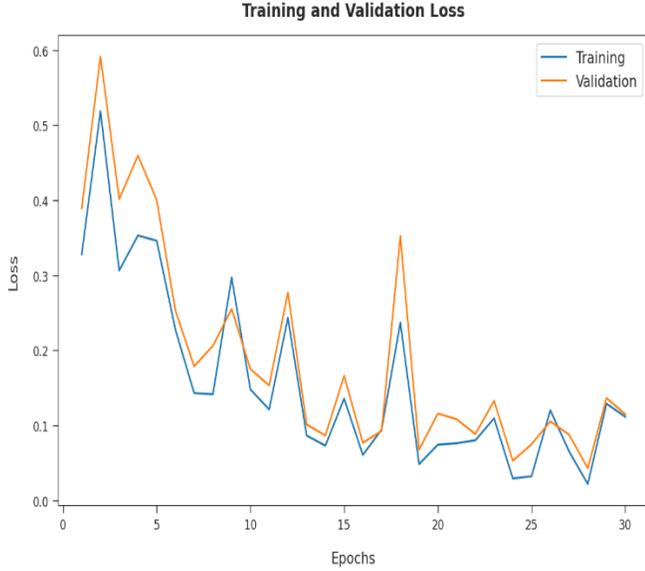


Fig. 9 Loss graph analysis of EOSN-GLDC technique

To exhibit the better outcomes of the EOSN-GLDC model, a comparative examination with recent methods is made in Table 2 and Fig. 10 [20]. The experimental values indicated that the GA-SVM, GA-BPN, and GA-Fuzzy models had showcased the least performance with  $a_n$  accuracy of 81.16%, 81.38%, and 82.57%, respectively. The ResNet18 model has offered reasonable performance with  $a_n$  accuracy of 98.90%,  $prec_n$  of 97.25%, and  $reca_l$  of 98.12%. In line with this, the VGG16 model has accomplished considerable outcomes with  $the\ accuracy$  of 98.50%,  $prec_n$  of 98.52%, and  $reca_l$  of 98.21%. However, the EOSN-GLDC model has achieved maximum  $accuracy$  of 99.31%,  $prec_n$  of 98.61%, and  $reca_l$  of 98.64%. From the detailed results and discussion, it can be clear that the EOSN-GLDC model has resulted in maximum performance over the other methods.

Table 2. Comparative analysis of EOSN-GLDC technique with existing algorithms

Methods	Accuracy	Precision	Recall
GA-SVM	81.16	81.07	80.74
GA-BPN	81.38	80.95	80.10
GA-Fuzzy	82.57	83.98	82.89
ResNet18	98.90	97.25	98.12
VGG16	98.50	98.52	98.21
EOSN-GLDC	99.31	98.61	98.64

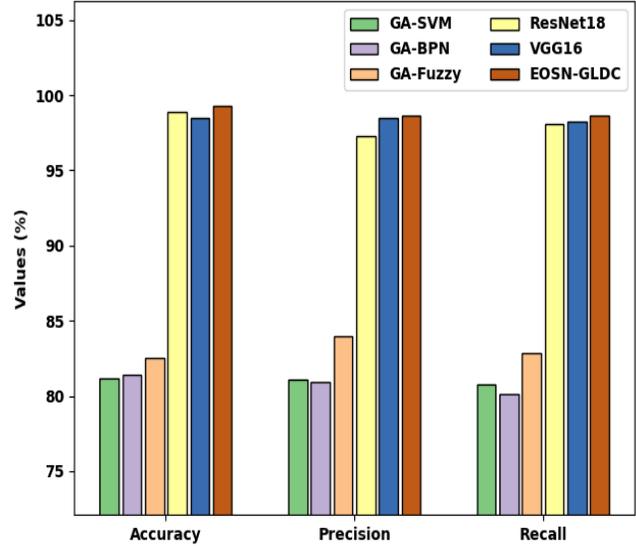


Fig. 10 Comparative analysis of EOSN-GLDC technique with existing algorithms

Therefore, the EOSN-GLDC technique appears as a novel method for grape leaf disease detection and classification.

#### 4. Conclusion

This study developed a novel EOSN-GLDC approach to detect and classify grape leaf diseases in an IoT environment. The presented EOSN-GLDC model has enabled IoT devices to collect grape leaf images and pre-process them. Then, it exploited the SqueezeNet model as a feature extractor where the hyperparameters involved are adjusted. Finally, an ELM classifier is applied to allocate proper class labels to the input images. To demonstrate the improved performance of the EOSN-GLDC model, a comprehensive experimental analysis is made using a benchmark dataset, and the outcomes indicate the betterment of the EOSN-GLDC model. In the future, the performance of the EOSN-GLDC model can be tuned by the hybrid DL classifiers.

#### Conflicts of Interest

This has no conflict of interest regarding the publication of this paper.

#### Funding Statement

There is no fund received for this research.

## References

- [1] S. M. Jaisakthi, P. Mirunalini, and D. Thenmozhi, Grape Leaf Disease Identification Using Machine Learning Techniques, In International Conference on Computational Intelligence in Data Science (ICCIDS), IEEE. (2019) 1-6.
- [2] L. Falaschetti, L. Manoni, C. F. Rivera, D. Pau, G. Romanazzi, O. Tomaselli, and C. Turchetti, A Low-Cost, Low-Power and Real-Time Image Detector for Grape Leaf Esca Disease Based on a Compressed CNN, IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 11(3) (2021) 468-481.
- [3] C. Zhou, Z. Zhang, S. Zhou, S. Xing, J., Q. Wu and J. Song, Grape Leaf Spot Identification Under Limited Samples by Fine Grained-GAN, IEEE Access. 9 (2021) 100480-100489.
- [4] L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, Machine Learning for Plant Leaf Disease Detection and Classification—A Review, In International Conference on Communication and Signal Processing (ICCSP), IEEE. (2019) 0538-0542.
- [5] A. D. Andrushia, and A. T. Patricia, Artificial Bee Colony Optimization (ABC) for Grape Leave Disease Detection, Evolving Systems. 11(1) (2020) 105-117.
- [6] S. Barburiceanu S, R. Terebes, and S. Meza, Grape Leaf Disease Classification Using LBP-Derived Texture Operators and Color, In IEEE International Conference on Automation, Quality, and Testing, Robotics (AQTR), IEEE. (2020) 1-6.
- [7] B. Liu, C. Li, S. He, and H. Wang, A Data Augmentation Method Based on Generative Adversarial Networks for Grape Leaf Disease Identification, IEEE Access. 8 (2020) 102188-102198.
- [8] X. Xie, Y. Liu, B. He, J. Li, and H. Wang, A Deep-Learning-Based Real-Time Detector for Grape Leaf Diseases Using Improved Convolutional Neural Networks, Frontiers in Plant Science. 11 (2020) 751.
- [9] X. Xie, Y. Ma, B. J. He, S. Li, and H. Wang, A Deep-Learning-Based Real-Time Detector for Grape Leaf Diseases using Improved Convolutional Neural Networks, Frontiers in Plant Science. 11 (2020) 751.
- [10] K. P. Ferentinos, Deep Learning Models for Plant Disease Detection and Diagnosis, Comput. Electron. Agric. 145 (2018) 311–318. Doi: 10.1016/j.compag.2018.01.009.
- [11] B. Liu, Z. Ding, L. Tian, D. He, S. Li, and H. Wang, Using Improved Deep Convolutional Neural Networks, Grape Leaf Disease Identification, Frontiers in Plant Science. 11 (2020) 1082.
- [12] P. Amudala, An Efficient Approach for Detecting Grape Leaf Disease Detection, International Journal of Circuit, Computing, and Networking. 1(2) (2020).
- [13] R Dwivedi R, Dey S, Chakraborty C, and Tiwari S, Grape Disease Detection Network Based on Multi-Task Learning and Attention Features, IEEE Sensors Journal. 21(16) (2021) 17573-17580.
- [14] M. Ji, M., Zhang, L., and Q. Wu, Automatic Grape Leaf Disease Identification Via Unitedmodel Based on Multiple Convolutional Neural Networks, Information Processing in Agriculture. 7(3) (2020) 418-426.
- [15] B. Koonce, SqueezeNet, In Convolutional Neural Networks with Swift for Tensorflow, Apress, Berkeley, CA. (2021) 73-85.
- [16] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, Equilibrium Optimizer: A Novel Optimization Algorithm, Knowledge-Based Systems. 191 (2020) 105190.
- [17] M. Abdel-Basset, R. Mohamed, S. Mirjalili, R. K. Chakraborty, and M. J. Ryan, Solar Photovoltaic Parameter Estimation Using an Improved Equilibrium Optimizer, Solar Energy. 209 (2020) 694-708.
- [18] T. S. Chy, and M. A. Rahaman, To Detect Sickle Cell Anemia, A Comparative Analysis by KNN, SVM & ELM Classification, In International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), IEEE. (2019) 455-459.
- [19] [Online]. Available: <https://www.kaggle.com/abdallahalidev/plantvillage-dataset>.
- [20] W. Harshal, R. Kokare, and Y. Dandawate, Detection and Classification of Diseases of the Grape Plant Using Opposite Color Local Binary Pattern Feature and Machine Learning for the Automated Decision Support System, In 3rd International Conference on Signal Processing and Integrated Networks (SPIN), IEEE. (2016) 513-518.