# Early and Late Fusion of Deep Convolutional Neural Networks and Evolutionary feature optimization for Plant leaf illness recognition

**Jalal Sadoon Hameed Al-bayati [1*], Burak Berk Üstündağ[2*]**
**[1,2] Department of Computer Engineering; Faculty of Computer and Informatics Engineering; Istanbul Technical University; Maslak, Ayazaga, 34469 Istanbul, Turkey**

***Corresponding Author:* Jalal Sadoon Hameed Al-bayati**
**E-mail:** al-bayati@itu.edu.tr ; jalal.hameed@uobaghdad.edu.iq; jalal_albayati@yahoo.com

**ABSTRACT**: Plant leaf illness recognition is an effective research topic in the last decade. Related to this interest, deep learning architectures are showing great era in various image processing and computer vision fields including image classification, feature detection, and pattern recognition in images. In this study, we investigate many aspects of convolutional neural networks for image pattern recognition, we examine the early and late fusion of multiple classifiers of pattern recognition using several plants leaves Commonly, plenty of time had been taken in consideration for disease discovering with the available technologies of diagnosis, and in normal cases, Planters normally fail to find the best period of illness prevention. The detection of Plant leaf diseases is a considerable research issue, and one if their goals are uncovering an effective way for leaf image illness identification. The article has made a potential effort to find a procedure that should be able to expose the illness of plant leaf using early and late fusion of two classifiers: Modified Optimized Deep Neural Network (MODNN) with evolutionary grasshopper feature optimization (GOA), Speeded Up Robust Features (SURF) and Modified Convolutional Neural network (MCNN) which could support the system to obtain excellent discover and high classification accuracy. The Classification model parameters, such as Precision, Recall, F-measure, Error, and Accuracy is computed, and an analysis has been done to describe the validation of the model.

**Keywords:** Fusion, Classifiers, accuracy, leaf illness; MCNN, MODNN, GOA, SURF.

## 1. Introduction

Disease in plant leaves is one of the main concerns because it affects the production and results in economy losses. Plant disease detection is developing area as agriculture is an important sector in Economy and Social life. Unscientific methods were used previously. Slowly with technical and scientific advancement, more responsible methods through moderate turnaround time are designed and proposed for advanced detection of plant disease. Such systems are extensively used and proved helpful to farmers as identification of plant disease is possible with less time span and improving actions are carried out at the appropriate time [1]. The plant leaves normally captured and uploaded to computers to be an image. These images are two dimensional signal, which can be defined by using a function $f(x,y)$, here x is the horizontal coordinate whereas y is the vertical coordinate. This function is used to provide the pixel value of the image at that coordinate [2]. The scanned or digital image captured by any digital equipment is two dimensional array of numbers varying from 0 to 255. In this research work, an early and late fusion of multiple classifiers for plant leaf illness recognition system is developed by using feature optimization and classification algorithm. The process of finding the leaf illness goes through various stages like pre-processing, segmentation, extraction, optimization and classification [3]. The article organized as follows. Section 2 describes the literature review and related work, followed by the experiment architecture in Section 3. Section 4 describes the system methodology, Section 5 analyses the results and discussion. Lastly, the conclusion is concluded with the future aspects in the last section following the references.

## 2. Related Work

Classification of the plant diseases is the key to preventing the losses in the amount of the farming product. The studies of the plant diseases signify the studies of visually observable patterns seen on the plant leaf. Strength monitoring and disease detection on plant using their leaf is very critical for sustainable agriculture. *Chouhan et al.*. have used some image processing schemes were utilized which monitors the images automatically to discover the pattern of the rotten region. *Kalaivani et al.* have used some validation algorithms and similarity formulas have been further used to boost the results. *Zhang et al.* have suggested a new version of segmentation based on a hybrid clustering. The model approach additionally directed through the segmentation to speed up the convergence of the expectation maximization algorithm. The experimental analysis have shown that the approach is effective. *Salazar-Reque et al.* have suggested also a novel crop protection scheme in the farming activity. On the other hand, the scheme was a time-consuming. The model have utilized the Simple Linear Iterative Clustering to differentiate the same regions as called superpixels.By using the artificial neural networks, they trained the model. The Model consisted of the properties of color pixels from the superpixels regions for the classification as healthy or not healthy. *Golhani et al.* have proposed a advanced approach appears to be an efficient machine learning model to detect the illness using the Spectral Disease Index. *Liu Bin et al.* have built a new structure based on AlexNet's convolutional neural network to detect apple leaf illness was created. The study demonstrated that the proposed deep neural network model provides a better solution for disease control of apple leaf disease. *Sladojevic et al.* have created a database with deep convolutional network to recognize 13 different types of plant illness out of healthy leaves. *Mohanty et al.* used a public data of PlantVillage of more than 50,000 images of broken and healthy plant leaves. *Zhang Chuanlei et al*. have presented an image processing scheme for an apple leaf disease identification using Support Victor Machine classifier. The model has presented 90% classification accuracy on apple leaves and has shown that the model is quality and practical.

## 3. Experiment Architecture

This section briefs the proposed work. Figure 1 represents the framework for Early and Late fusion of using Convolutional Neural Networks and Neural Networks with evolutionary feature optimization and feature extraction for the Plant Illness Recognition Fusion System (PIRFS). The PIRFS uses two classifiers: the first one is four models of Convolutional Neural Networks (AlexNet, GoogleNet,Resnet, and Inception), the second classifier is the SURF external feature extraction method, which supports the reduction of recognition complexity with highest accuracy performance, besides, the system is using Image Intensity Control approach to adjust the image enhancement, for the optimization of extracted features, evolutionary Grasshopper Optimization Algorithm (GOA) is utilized as feature optimization. Training, Testing, and validation of datasets are classified using the early fusion and late fusion of Convolutional Neural Network and Grasshopper optimization of Deep Neural Network, the framework of the system is given in Figure 3.

The system architecture starts with uploading the images into the system as data acquisition stage, then pre-processing using color intensity functions, and segmentation using K-means algorithm with morphological operations, the output images are processed through two classifiers, the training sets are passed through early fusion stage for the classification, or passed through late fusion by simple fusion formula. Finally, the recognized matching performance results are obtained via performance confusion matrix.

In general, The Stages of CNN architecture are given as:

*i.* ***Convolution Layers – the kernel:*** the convolution layer can carry an input data image, select the learnable weights and biases to multiple patterns inside the image and can distinguish an object from the other. The pre-processing needed in a CNN is much lower as compared to many classification methods. The learnable layers assist to carry out the required feature datasets. Each layer is utilized to the pixel values of the plant leaf image. The CNN layers work as augmentation filters, when we augment the 265x265x3 image into a 180x180x3 image and then apply the 3x3x1 kernel, the created convolved matrix will have dimensions of 265x265x3, which is called *Same Padding*. If we applied the kernel without padding, it's called *Valid Padding*.

***ii.*** ***Pooling:*** The layer of pooling is required for feature learning aspect, by using the pooling layer, we will get the extracted and optimized features from the image. Normally, two kinds of pooling are used, the Max pooling or the Average pooling, the Max pooling layers return the maximum value from the part of the image that covered by the kernel, where the average pooling outputs the average values from the squared part of the image wrapped by the kernel. Most of the Realtime applications, the Max Pooling performs a lot better than the Average Pooling because of discarding of noisy activations with dimensionality reduction. On the other hand, the Average Pooling runs the reduction as a noise deletion. The output of max-pooling layer is crossed to next stage which is the feature classification.

iii. ***Feature Classification:*** feature classification stage is a fully connected layer is a inexpensive method of learning non-linear components of the high-dimension features as symbolized by the output of the kernel. From a series of iterations, the model can determine the important features in image and predict the classification of certain group of images using the *Softmax function* approach. There are many different numbers of layers and different activation functions which is occasionally called configurations, every changes of CNN architectures will create a new type of CNN architectures consider each one has good classicisation results for various applications related to computer vision field. The whole Convolutional Neutral Network architecture is shown in Figure 1.

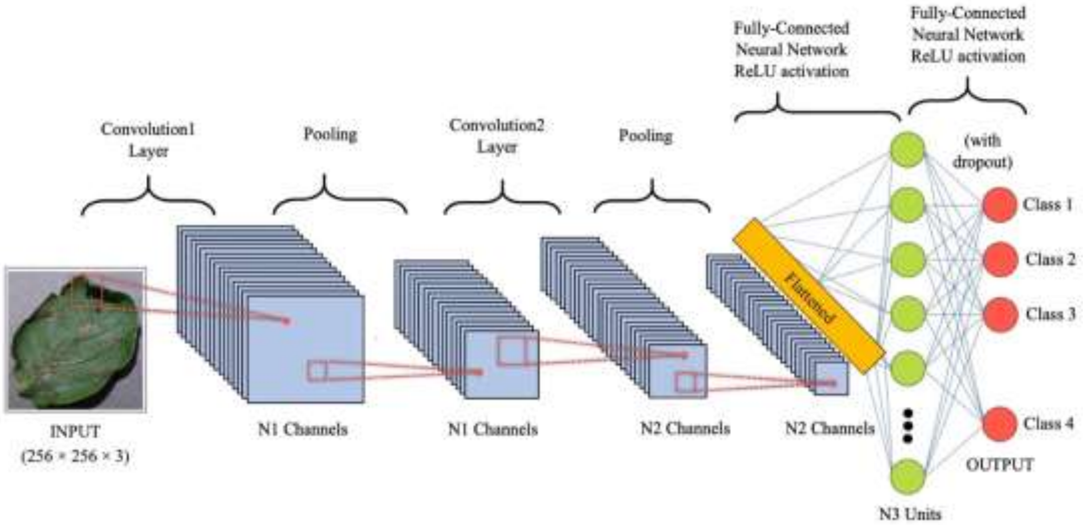


Fig.1. PIRFS CNN Architecture

The layers of Neural Network (Shown in Figure 3) are given as:

**i.** *Input Layer*: the input layer is the nodes of input image of the plant leaf, the inputs are determined through 1 to N. N is the number of nodes of input layer.

**ii.** *Hidden Layers*: the hidden layer is very important stage in artificial neural networks, it consists of a layer of neurons, the output of the hidden layer is connected to the inputs of the next hidden layer. It is not visible to the network output so it's called hidden. Normally, the first hidden layer finds the pixels of while and black colors as they the white and black pixels of the image, the second hidden layer creates the identification of shapes and edges, the third hidden layer identifies more complicated edges and shapes for the input image and so on it identifies more complex spots of the image to the output layer.

**iii.** *Output Layer*: the output layer is the last stage of the neural network that passed to the classification model, which is used as a maximum activation value and creates a structure of the model. The grasshopper optimization method had been used separately to increase the possibility of discovering the feature distinction from the image.

**iv.** *Classification Model*: The model is a fully connected layers, the fully connected layers have multiple neurons, each neuron is entirely connected to all learned optimized feature maps taken from the previous layer. They depend on the activation formula to calculate the class label probabilities.

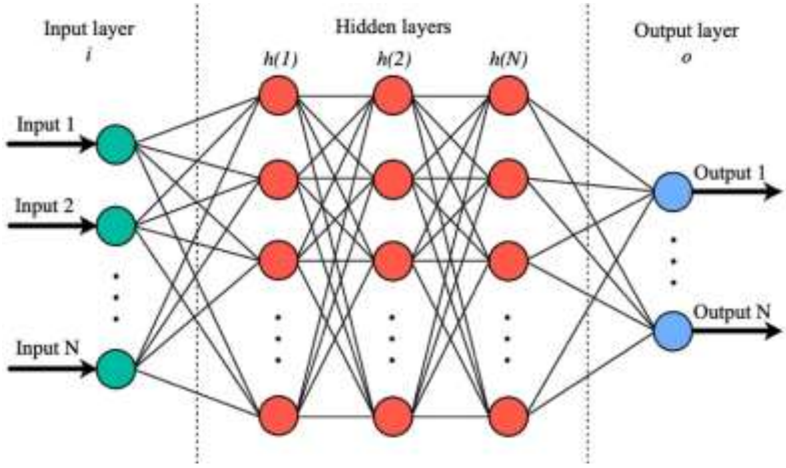


Fig.2. PIRFS DNN Architecture

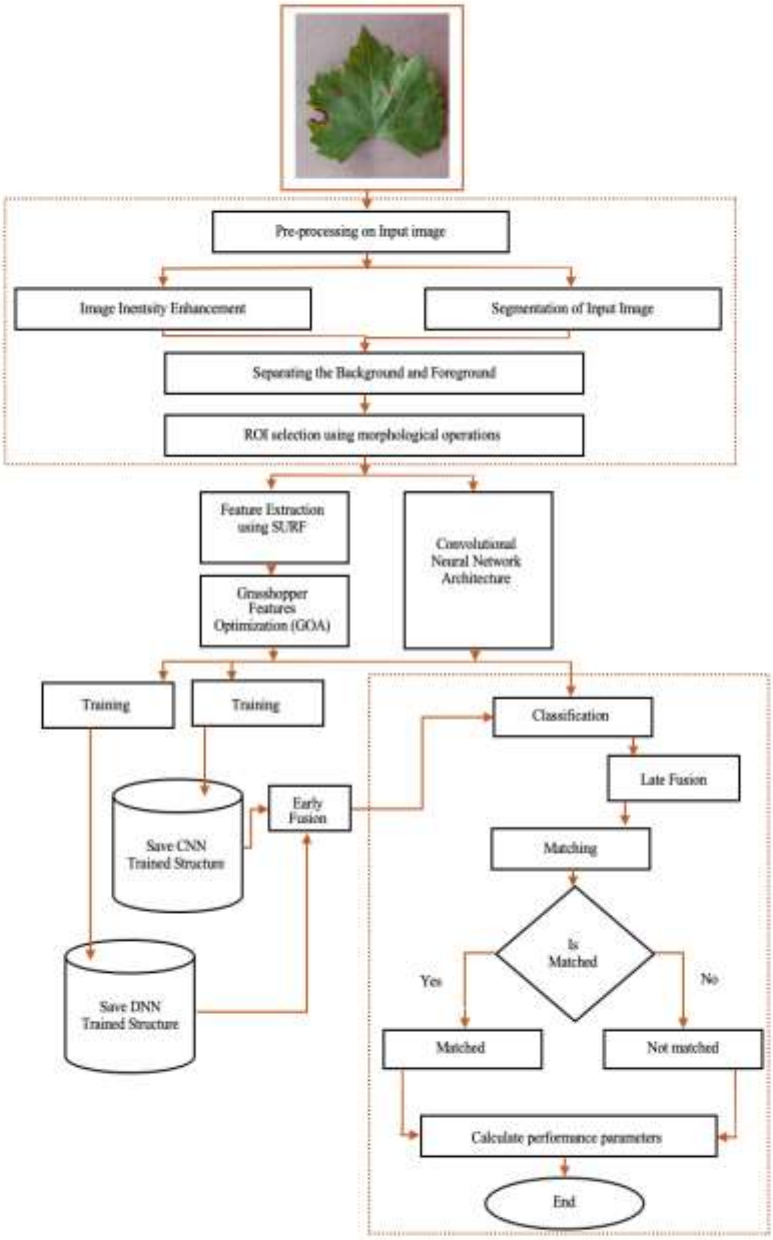


Fig.3. Framework of PIRFS

**4. System Methodology**

This section characterize the utilized approaches for plant leaf illness recognition from the plant leaf data images. The datasets are obtained from the dataset named as *Plant Village Datasets Master* [10]. In these datasets, Several types of plant illness from which apple, corn, grape, peach, pepper, potato, strawberry leaves are given. The types of illness for each plant category are shown in figure 4.

Fig. 4. Plant Leaf Images Dataset Sample, Right to Left (Apple black rot, Apple scab, Apple cedar-rust, Apple healthy, Corn cercospo, Corn common rust, Corn northern blight, Corn healthy, Grape black rot, Grape measles, Grape leaf blight, Grape healthy, Peach bacterial spot1,2,3, Peach healthy, Pepper bacterial spot1,2,3, Pepper spot, Potato early blight, Potato late blight, Potato healthy1,2, Strawberry scorch1,2,3, Strawberry healthy.

The Methodology that are followed to layout an perfect plant leaf illness recognition and classification model are demonstrated in details below [11]:

***a. Image Contrast Enhancement***

Image enhancement approaches are utilized to get an accurate plant illness image, it's known that for grey level images. Normally, the lowest and highest borders of pixel color values are 0 and 255. The borders by using the image contrast enhancement are defined as 'Lo" and "Hi". This normalization peruse the uploaded plant leaf image to find these pixel color values founded in the plant leaf image, which required contrast enhancement. The perused pixels are namely $L_p$ and $H_p$. All pixels ($P_I$) of plant leaf image is enhanced using the formula:

$$P_{ICE} = (P_I - L_p) \times \left(\frac{Hi - Lo}{H_p - L_p}\right) + Lo, \qquad (1)$$

As shown in equation (1), $P_{ICE}$ is the pixel of image contrast enhanced, and $P_I$ is original plant leaf image. The next Figure presents the results of image contrast enhancement technique [13].

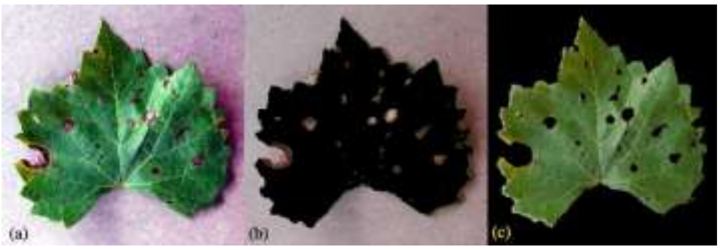

Fig.5. (a) Enhanced Contrast image (b) Background image (c) Foreground image

The following formula is utilized for the enhancement pixels as the output of the plant leaf image pre-processing as shown in figure 5.

$$P_{IFL} = P_{frame} - P_{average}, \qquad (2)$$

Where $P_{IFL}$ : Pixels Frame Limit which is utilized to modify the limit of contrast through the image enhancement. The next stage after the enhancement of the image, the k-means clustering is utilized.

### *b. Segmentation*

By applying K-means, the foreground is extracted and then morphological operations are utilized to segment ROI[14]. Segmentation of ROI is utilized to grant more accuracy of illness classification, the figure 6 represents the result of segmentation. Clusters of foreground and background are 2, clustering the two regions using Replicates and Squared Euclidean distance as each centroid is the mean of pixels in that region for decreasing convergence time.

Fig.6. Segmented Image ( grape leaf)

### *c. Feature extraction*

By running the SURF scheme as a feature extraction method on the segmented image [15], yields to extract only the region of the leaf, which is affected by the illness, the extracted features of the segmented image are emphasized as they were shown in Figure 7.

Fig. 7. Extracted SURF Features after segmentation

### *d. Feature optimization method*

GOA is used to optimize SURF features and to remove the unwanted feature sets by using the different objective fitness function.[16].

$$\text{Fitness function: } f(\text{fit}) = \begin{cases} 1, & \text{fs} < \text{ft} \\ 0, & \text{fs} \geq \text{ft} \end{cases} \quad (3)$$

Grasshopper optimization algorithm is inspired by the nourishment behavior of grasshoppers. The grasshopper's lifecycle has two main phases: the larval phase and adulthood phase. The distinguishing feature in the larval phase is slow movement and small step movement, while the adulthood phase is bigger and unexpected movement is the vital feature of the swarm in adulthood phase. The food source search process has two directions: exploration and exploitation. For the exploration, grasshoppers tend to move quickly. on the other hand, they are willing to move locally in the exploitation phase. These two operations and locating a food source, are done by grasshoppers in a natural manner. By using mathematics, the swarming or grouping actions of grasshoppers are modelled in the following equation:

$$X_i = S_i + G_i + A_i \quad (4)$$

Where $X_i$: Position of the grasshopper i, $S_i$: Social interaction, $A_i$: Wind advection

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\widehat{d_{ij}} \quad (5)$$

Where $d_{ij}$: distance between ith and jth grasshoppers, $d_{ij} = |x_j - x_i|$. $s$: Strength of the social force, $\widehat{d_{ij}}$: The unit vector is calculated using $\widehat{d_{ij}} = \frac{x_j - x_i}{d_{ij}}$

$$S(r) = f e^{\frac{r}{l}} - e^{-r} \quad (6)$$

Where $f$ is the intensity of attraction and $l$ indicates the attractive length scale. The $G_i$ component in Equation 4 is calculated as:

$$G_i = -g\widehat{e_g} \quad (7)$$

$g$: gravitational constant, $\widehat{e_g}$: unity vector toward the center of earth. The $A_i$ component in Equation 4 is calculated as:

$$A_i = u\widehat{e_w} \quad (8)$$

$u$: Constant drift, $\widehat{e_w}$: unity vector in the direction of the wind, Substituting in Equation 4 results:

$$X_i = \sum_{j=1, j\neq i}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w}) \quad (9)$$

In the optimization algorithm, the equation (9) is not used, as it averts the optimization algorithm from exploring and exploiting the search space nearby a solution. Actually, this nymph grasshopper model is designed for the grasshopper swarm which resides in free space. Moreover, this mathematical model was not employed directly to solve optimization problems, as the grasshoppers rapidly achieve the comfort zone and the swarm does not converge to a specified point. A modified version of Eq. (9) is employed to solve optimization problems:

$$X_i^d = c_1 \left( \sum_{j=1, j\neq i}^{N} c_2 (\left|\frac{ub_d - lb_d}{2}\right|) s(|x_j^d - x_i^d| \frac{x_j - x_i}{d_{ij}} \right) + \widehat{T_d} \quad (10)$$

$ub_d - lb_d$: represents the upper and lower bounds, $T_d$: The target value and best solution, $c_1, c_2$: coefficients used to shrink the rest zone, repulsion zone and attraction zone. The gravity component is not used and its assumed that the wind direction is always toward a target $T_d$.To calculate the next position of the grasshopper, the position of the target, the current position and the position of all other grasshoppers are used. The parameter $c_1$ is used for reducing the movements of grasshoppers around the target which means that $c_2$ balances the exploration and exploitation of the whole swarm around the target. The parameter $c_2$ is used to decrease the space to lead the grasshoppers to find the optimal solution in the search space. Both parameters ($c_1$ and $c_2$) can be considered as a single parameter and it is modified using this equation:

$$c = c_{max} - I\frac{c_{max}-c_{min}}{N} \quad (11)$$

Where $I$: number of current iterations, $N$: maximum number of iterations, $c_{max}, c_{min}$: maximum and minimum value of $c$. The algorithm pseudocode used for feature optimization given below:

Initialize Objective function $f(x), x = (x_1, x_2, \dots\dots x_{dim})$, dim = no. of dimensions
Generate initial population of n grasshoppers $x_i, i = 1,2, \dots.. n)$
Calculate fitness of each grasshopper.
T = the best search agent
While stopping criteria not met do
Update c1 using $c = c_{max} - I\frac{c_{max}-c_{min}}{N}$
Update c2 using $c = c_{max} - I\frac{c_{max}-c_{min}}{N}$
for each grasshopper gh in population do
Normalize the distances between grasshoppers
Update the position of the gh by Eq. (10)
If needed, update the bounds of gh
end
If there is a better solution, update T

Feature optimization is also known as attribute selection in any pattern recognition. It's like dimensionality reduction, both schemes try to minimize the number of features. However, the main difference is dimensionality reduction creates new combinations of features. On the other hand, feature optimization excludes the features without made any changes[17]. The next stage after feature optimization, the optimized features are passed through the fully connected neural network.

### *e. Modified Optimized Deep Neural Network classifier (MODNN)*

The fully connected neural network layer is created using algorithmic architecture which it's pseudocode is proposed as follows.

```
Initialize NN with parameters
E = No. of Iterations
N = Neurons as carrier
Class = class of each category of plants
Training parameters for performance evaluation
Training used: Scaled Conjugate Gradient
TD Division: Random / [0.7,0.3,0.3]
For each set of Ts
If Training Dataset (TDs) belongs to Class1 of feature set
Group (1) = class N of TDs
Else if Training Data (TDs) belongs to Class2 of feature set
Group (2) = class N of TDs
Else if Training Data (TDs) belongs to Class3 of feature set
```

```
Group (3) = class N of TDs
End
Initialized NN in model using Training dataset TDs and their Target Tar
Network = patternnet (N)
training parameters = set
Network = Train (Network, TDs, Class)
Classification scores = simulation (Network, Test optimized features)
If Classification scores = True
Run Classification scores of the plant leaf
Calculate parameters of performance
End
Returns: Results
End
```

### *f. Modified Convolutional Neural Network Models (MCNN)*

The essential architecture in the convolutional neural network initiates with various convolutional layers and pooling layers, followed by fully connected neural network, for an input neuron *a* of *n*th convolutional neural layer, it assigns:

$$a_n = ReLU(W_n * a) \quad (12)$$

Where $W$represents the convolutional kernel. $W_n = [W_n^1, W_n^2, \ldots., W_n^m)$, $m$ is the kernel size of the convolutional layer. Each kernel is three dimensional weight matrix $I \times I \times J$. $I$is the window size and $J$ is the number of input neurons. * represents convolution operation, ReLU is the rectified linear function $ReLU(x) = \max(0, x)$ which is utilized as activation function for the CNN models represented by high resolution images is sometimes impossible. For the neural network training aims to calculate the values of $W$ that minimize the loss error function $Er$. Gradient descent approach is utilized where $W$ is updated through iterations as:

$$W_c = W_{c-1} - \alpha \frac{\partial Er(W)}{\partial W}, \quad (13)$$

Where $\alpha$ is the learning rate, which is very essential parameter that defines the step size of learning, this parameter should be very carefully selected because it will be one the reasons of underfitting and overfitting. For the CCN models, we have utilized four kinds of standard configurations, In AlexNet, the configuration was significantly revealed that the experimental tests outperformed the old ones, it had very similar architecture of LeNet, however, it had more layers meaning it was deeper, and with stack convolutional layers. It contained 11×11, 5×5,3×3, convolution layers, max pooling, dropout, ReLU activation and Stochastic Gradient Descent with momentum. For the GoogLeNet, the winner of ImageNet Large Scale Visual Recognition Competition in 2014, It achieved very high-level accuracy performance and excellent evaluation, this architecture consisted of a twenty-two deep CNN layers, however, the number of parameters processed from the system was reduced from 60 million parameters as AlexNet architecture to only 4 million parameters, it used convolution, pooling, SoftMax activation. In VGG19, from its name, it contains 19 convolutional layers and its similar to AlexNet, only 3×3 convolutions, however it had lots of filters. For the last one, the ResNet or Residual Neural Network (RNN), it used the skip technique to skip some connections which is also known as gated recurrent units and have a strong similarity to the latest successful elements that applied in these networks. The RNN, by using the skip units, were able to train 152 layers with lower complexity that older architectures with fewer layers. The Inception, also an important and very famous method in the world of deep learning CNN models which has a depth of 48 layers, below is the pseudocode of the Model.

```
ConV_model = CNN_ model
N= number of images
miniBatchSize = 10
 For m = 1 to N
Images = Folders of images including subfolders and Labels
Split size = (Trainingset = 0.7,Testset = 0.3, Validationset = 0.3)
    For m = 1 to N(Rezise images according to the best fit)
augmentation Trainingset = augmentedImages([256 256 3]
augmentation Validationset = augmentedImages([256 256 3]
    End
 End
 For m = 1 to N
training_option = stochastic gradient descent with momentum learning
ConV Trainingset = Images,network, training_option
Y_Predicitons = Classification of Trainingset,Testset,Validationset
 end
accuracy = accuracy of Trainingset,Testingset,Validationset
Plot confusion matrix = confusion chart[Trainingset,Testingset,Validationset]
Calculate Performance
```

The following table summarize the CNN models and their parameters;

Table 1.CNN Models parameters

| Model | No. of Layers | Convolution Layers | Pooling | Activation/Prediction |
|---|---|---|---|---|
| AlexNet | 25 | 5 | 5 (Max) | RelU/Softmax |
| GoogLeNet | 144 | 57 | 13(Max) | RelU/Softmax |
| ResNet18 | 71 | 20 | 17(Max) | RelU/Softmax |
| Inceptionv3 | 315 | 94 | 4(Max)-9(Average) | RelU/Softmax |

### *g. Early and late fusion*

Many early and late fusion strategies have been created, early fusion have some advantages over late fusion strategies given that they are conducted through the training sets properly. Various feature training sets show different characteristics of same pattern and combining those features retain active discriminant information while completely remove the redundant information. In our approach, we have early fusion that separates the training datasets into two segments, each segment is passed through training and then exceeded to the classification. In contrary to early fusion, late fusion approaches train separate classifiers for each of the image channels present in our dataset. If we have $\boldsymbol{n}$ different classifiers, then it yields various learning modalities. thereafter, late fusion approaches merge those $\boldsymbol{n}$ different classifier scores into one classification value [23]. Late fusion approaches have several obvious obstacles compared to early fusion approaches. First of all, late fusion approach increases the computational time required due to the higher number of classifiers to be trained. Second obstacle is that classifier models are not symbolized with information from different models, so correlations between those models are not considered in the classifier outputs. Despite these clear drawbacks, late fusion approaches are stated to acquire comparable or even better results to early fusion approaches in some applications. Yang et al. refer this to heterogeneous and independent nature of multimedia features as hierarchical regression for multimedia analysis [35]. The late fusion function contains the function that combines the prediction outputs of both our system and CNN outputs by taking the arithmetic mean of them as a late fusion method [37].

$$\boldsymbol{Pred^c = argmax\left(\frac{Pred^c_{OMDNN}+Pred^c_{MCNN}}{2}\right)} \tag{14}$$

Where $\boldsymbol{Pred^c}$ is the prediction of late fusion classifiers classification score, $\boldsymbol{Pred^c_{OMDNN}}$ is the prediction of classification of using optimized neural networks classifier, $\boldsymbol{Pred^c_{MCNN}}$ is the prediction of classification of using modified convolutional neural network classifier.

### *h. Performance Evaluation*

Performance evaluations must be fairly considered to evaluate the models we have suggested and created by utilizing standard evaluation form for each model. The performance of the models are done using the parameters of confusion matrix, accuracy, precision, recall, F-Score, and Error rate.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \times \mathbf{100}, \tag{15}$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}, \tag{16}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}, \tag{17}$$

$$\boldsymbol{F-Score} = \mathbf{2} \times \frac{\boldsymbol{Precision\times Recall}}{\boldsymbol{Precision+Recall}}, \tag{18}$$

$$\boldsymbol{Error} = \frac{\boldsymbol{FN+FP}}{\boldsymbol{TP+TN+FP+FN}} \times \mathbf{100}, \tag{19}$$

Where, $\boldsymbol{TP}$: True Positives, $\boldsymbol{TN}$: True Negatives, $\boldsymbol{FP}$: False Positives, $\boldsymbol{FN}$: False Negatives.

**5. Results and Evaluation**

This section explains the result and comparative analysis obtained after the evaluation of the article work. For the evaluation, QoS parameters, such as Precision, Recall, F-measure, Error, and Accuracy, is considered, the dataset consists of more than 20,000 images,7 categories, 14 illness classes, and 7 healthy classes. In the preceding, the comparative analysis using early and late fusion is performed to check the uniqueness of the article work.

Table 2. Early fusion Precision scores.

| *Plant Leaf Category* | *MODNN with Alexnet* | *MODNN with GoogleNet* | *MODNN with Resnet* | *MODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9468 | 0.9635 | 0.9662 | 0.9596 |
| *Corn* | 0.9304 | 0.9504 | 0.944 | 0.9495 |
| *Grape* | 0.9823 | 0.9738 | 0.9812 | 0.9699 |
| *Peach* | 0.9874 | 0.9919 | 0.9810 | 0.9766 |
| *Pepper* | 0.9869 | 0.9799 | 0.9777 | 0.9649 |
| *Potato* | 0.9308 | 0.9698 | 0.9732 | 0.9506 |
| *Strawberry* | 0.9872 | 0.9926 | 0.9779 | 0.9724 |

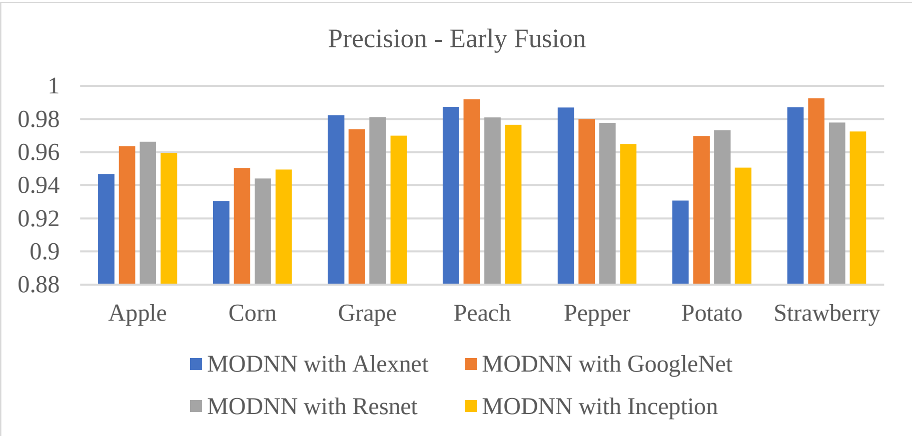


Fig. 8. Early fusion Precision scores chart

Figure 8 and Table 2 shows the precision computation of the early and late fusion approach. The Figure shows the comparison of Optimized Deep Neural Network with four CNN models: Alexnet, GoogleNet,Resnet, and Inception. Precision is the mean of correct values in the process of evaluation with respect to the total dataset. Using seven categories of plants with four Optimized CNN models using early fusion, the best result in Apple is MODNN-Resnet with 0.97. For Corn, the best result is MODNN-Googlenet with 0.95. For Grape, the best result is MODNN-Alexnet with 0.982. For Peach, is MODNN-Googlenet with 0.992. For Pepper, is MODNN-Alexnet with 0.986. For Potato, is MODNN-Resnet with 0.973. For Strawberry, is MODNN-Googlenet with 0.9926.

Table 3. Early fusion Recall scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9507 | 0.9234 | 0.9377 | 0.8833 |
| *Corn* | 0.9597 | 0.9702 | 0.9519 | 0.9558 |
| *Grape* | 0.9603 | 0.9541 | 0.9668 | 0.9460 |
| *Peach* | 0.9606 | 0.8427 | 0.8980 | 0.8348 |
| *Pepper* | 0.9605 | 0.8364 | 0.9001 | 0.8406 |
| *Potato* | 0.9154 | 0.8441 | 0.9073 | 0.8452 |
| *Strawberry* | 0.9522 | 0.8522 | 0.8879 | 0.8375 |

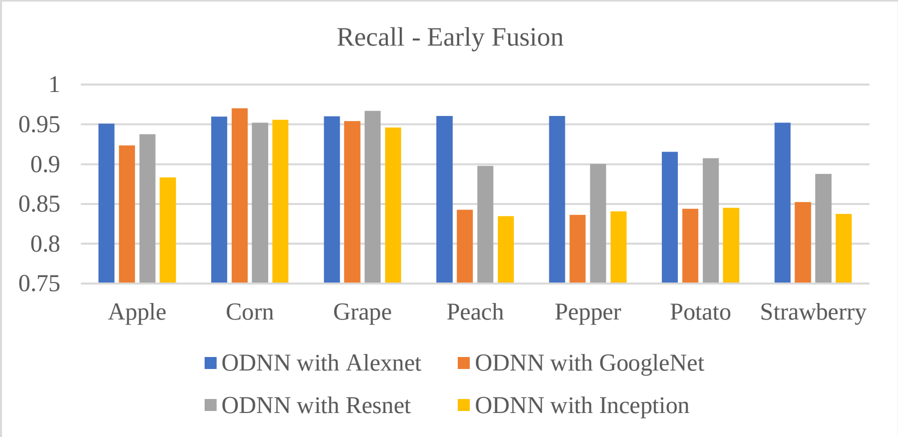


Fig.9. Early fusion Recall scores chart

Figure 9 and Table 3 shows the recall computation of the early and late fusion approach. The Figure shows the comparison of Optimized Deep Neural Network with four CNN models: Alexnet, GoogleNet,Resnet, and Inception. Recall is the mean of relevant instances in the process of evaluation among the retrieved instances. Using seven categories of plants with four Optimized CNN models using early fusion, the best recall result in Apple is MODNN-Googlenet with 0.95. For Corn, the best result is MODNN-Googlenet with 0.97. For Grape, the best result is MODNN-resnet with 0.97. For Peach, is MODNN-Alexnet with 0.96. For Pepper, is MODNN-Alexnet with 0.96. For Potato, is MODNN-Alexnet with 0.91. For Strawberry, is MODNN-Alexnet with 0.95.

Table 4. Early fusion F-measure scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9484 | 0.9431 | 0.9517 | 0.9157 |
| *Corn* | 0.9448 | 0.9602 | 0.9480 | 0.9526 |
| *Grape* | 0.9712 | 0.9638 | 0.9739 | 0.9579 |
| *Peach* | 0.9738 | 0.9113 | 0.9377 | 0.9002 |
| *Pepper* | 0.9735 | 0.9025 | 0.9373 | 0.8984 |
| *Potato* | 0.9230 | 0.9026 | 0.9391 | 0.8948 |
| *Strawberry* | 0.9694 | 0.9170 | 0.9037 | 0.8999 |

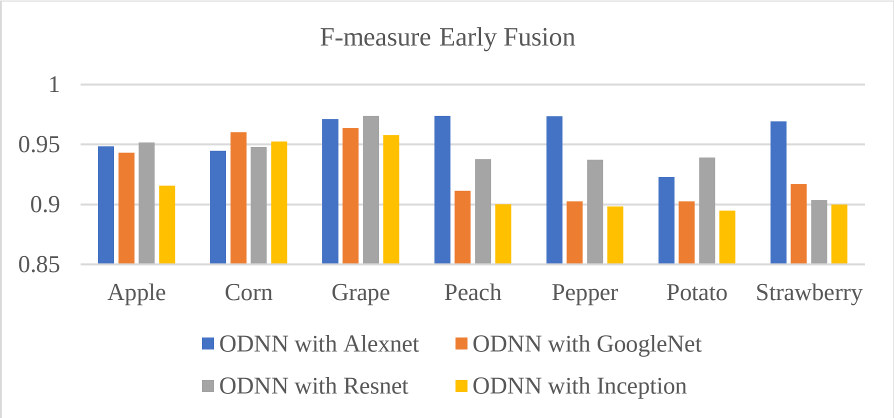


Fig.10. Early fusion F-measure scores chart

The computation of F-measure is shown in Figure 10. F-measure is the harmonic mean of precision and recall rate. The Figure depicts the comparison of fusion of optimized NN with CNN. The highest F-measure score is 0.9739 in grape plants using MODNN-Resnet.

Table 5. Early fusion Error scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 3.8988 | 3.3725 | 3.1619 | 5.5304 |
| *Corn* | 6.3068 | 4.1577 | 5.0227 | 4.8497 |
| *Grape* | 2.4342 | 3.1725 | 2.1881 | 3.6237 |
| *Peach* | 0.9954 | 0.8699 | 1.2463 | 1.9364 |
| *Pepper* | 2.1412 | 3.4889 | 2.7477 | 4.6345 |
| *Potato* | 6.2889 | 5.4375 | 3.4251 | 6.9081 |
| *Strawberry* | 1.805 | 0.1029 | 2.6561 | 2.9752 |

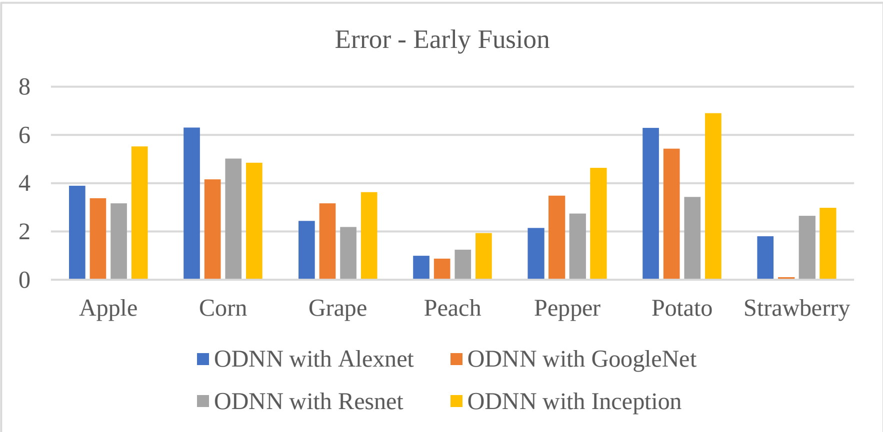


Fig. 11. Early fusion Error scores chart

The computation of Error is shown in Figure 11. The Figure explains the comparison of fusion. The lowest error found in Strawberry using MODNN-Googlenet. There is an enhancement of about 49.80% in the proposed model in Error.

Table 6. Early fusion Accuracy scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 96.1013 | 96.6276 | 96.8381 | 94.4695 |
| *Corn* | 93.5932 | 95.8423 | 94.9773 | 95.1503 |
| *Grape* | 97.5658 | 96.8275 | 97.8119 | 96.3763 |

| | | | | |
|---|---|---|---|---|
| *Peach* | 99.0047 | 99.1301 | 98.7537 | 98.0636 |
| *Pepper* | 97.8588 | 96.5111 | 97.2523 | 97.2523 |
| *Potato* | 93.7111 | 94.5626 | 96.5749 | 93.0919 |
| *Strawberry* | 98.1950 | 99.8971 | 97.3439 | 97.0248 |

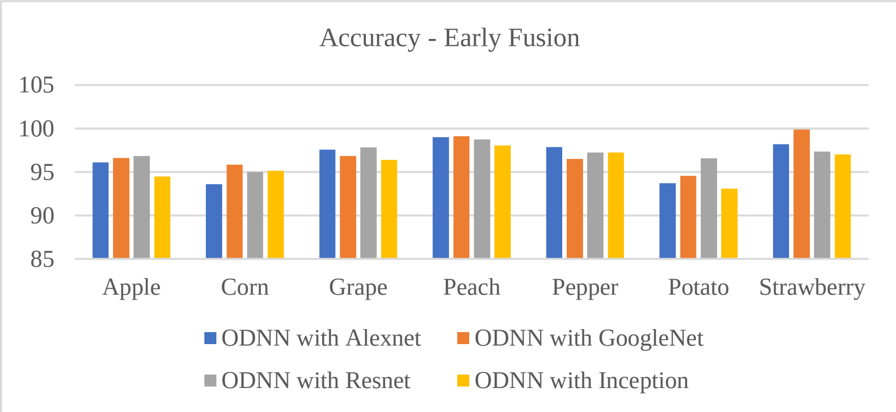


Fig. 12. Early fusion Accuracy scores chart

Figure 12 and Table 6 shows the Accuracy computation of the proposed approach. Accuracy is the main factor of the designed system.. The Figure shows the comparison of early fusion of Optimized DNN with CNN models. The x-axis in the Figure shows the samples, whereas Y-axis shows the Accuracy values. There is an enhancement of 2.22 % in the proposed model in accuracy than standard CNN model.

Table 7. Late fusion precision scores

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9273 | 0.9447 | 0.9473 | 0.9320 |
| *Corn* | 0.9123 | 0.9318 | 0.9258 | 0.9309 |
| *Grape* | 0.9631 | 0.9548 | 0.9620 | 0.9510 |
| *Peach* | 0.9681 | 0.9725 | 0.9618 | 0.9575 |
| *Pepper* | 0.9676 | 0.9608 | 0.9586 | 0.9461 |
| *Potato* | 0.9126 | 0.9509 | 0.9542 | 0.9321 |
| *Strawberry* | 0.9679 | 0.9732 | 0.9588 | 0.9534 |

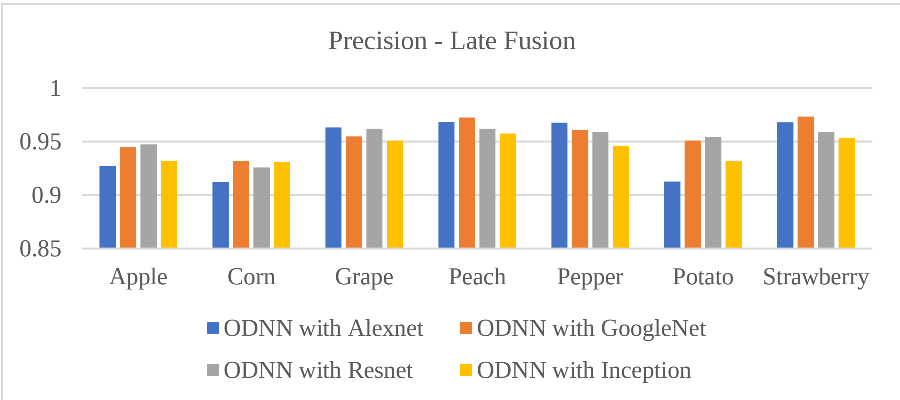


Fig. 13. Late fusion Precision scores chart

Figure 13 and Table 7 shows the precision computation of the late fusion approach. The Figure shows the comparison of late fusion of Optimized Neural Network with four CNN models: Alexnet, GoogleNet,Resnet, and Inception. The best result in Apple is MODNN-Resnet with 0.94. For Corn, the best result is MODNN-Googlenet with 0.93. For Grape, the best result is MODNN-Alexnet with 0.96. For Peach, is MODNN-

Googlenet with 0.97. For Pepper, is MODNN-Googlenet with 0.967. For Potato, is MODNN-Resnet with 0.954. For Strawberry, is MODNN-Googlenet with 0.973.

Table 8. Late fusion recall scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9332 | 0.9062 | 0.9201 | 0.8668 |
| *Corn* | 0.9417 | 0.9524 | 0.9341 | 0.9379 |
| *Grape* | 0.9423 | 0.9362 | 0.9487 | 0.9283 |
| *Peach* | 0.9426 | 0.8269 | 0.8812 | 0.8192 |
| *Pepper* | 0.9426 | 0.8207 | 0.8833 | 0.8248 |
| *Potato* | 0.8983 | 0.8283 | 0.8903 | 0.8293 |
| *Strawberry* | 0.9343 | 0.8362 | 0.8712 | 0.8218 |

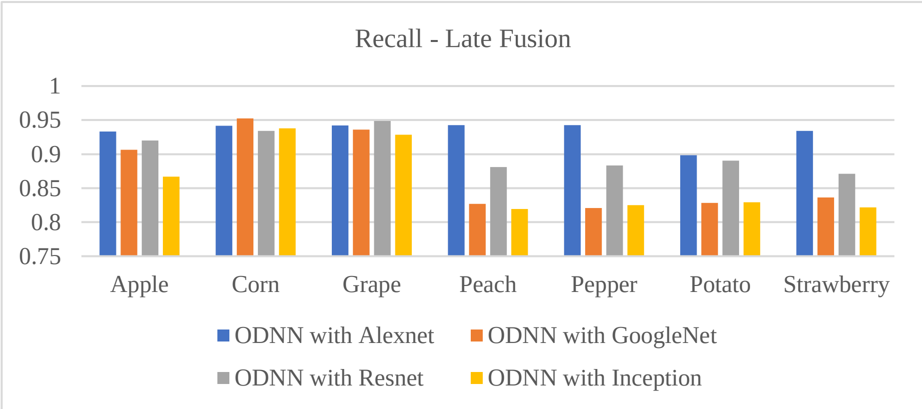


Fig. 14. Late fusion Recall scores chart

Figure 14 and Table 8 shows the recall computation of the proposed approach. The Best recall value in Corn plants using MODNN-Googlenet with 0.952.

Table 9. Late fusion F-measure scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 0.9302 | 0.9251 | 0.9335 | 0.8982 |
| *Corn* | 0.9268 | 0.9418 | 0.9299 | 0.9344 |
| *Grape* | 0.9526 | 0.9454 | 0.9554 | 0.9396 |
| *Peach* | 0.9552 | 0.8939 | 0.9198 | 0.8829 |
| *Pepper* | 0.9549 | 0.8853 | 0.9194 | 0.8813 |
| *Potato* | 0.9054 | 0.8854 | 0.9211 | 0.8777 |
| *Strawberry* | 0.9508 | 0.8995 | 0.9219 | 0.8827 |

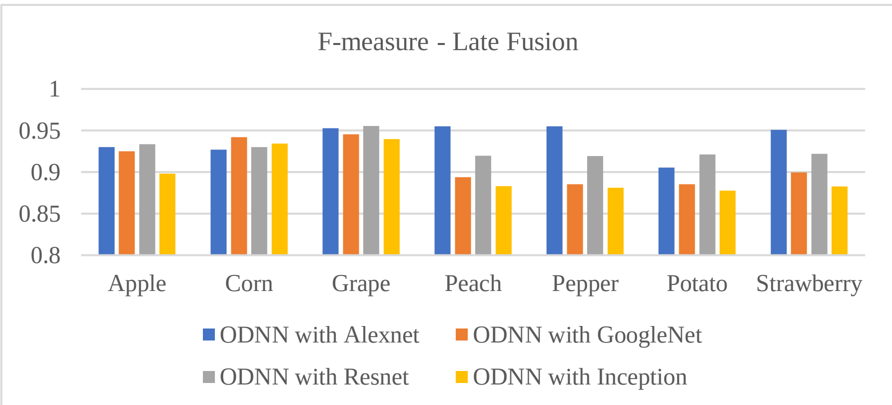


Fig. 15. Late fusion F-measure scores chart

Figure 15 and Table 9 shows the F-measure computation of the proposed approach. The Highest F-measure value in Grape plants with MODNN-Resnet with 0.9554.

Table 10. Late fusion Error scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 3.9808 | 3.4445 | 3.2339 | 5.6024 |
| *Corn* | 6.4788 | 4.2297 | 5.0948 | 4.9217 |
| *Grape* | 2.5062 | 3.2445 | 2.2601 | 3.6957 |
| *Peach* | 1.0674 | 0.9419 | 1.3183 | 2.0084 |
| *Pepper* | 2.2132 | 3.5609 | 2.8197 | 4.7065 |
| *Potato* | 6.3609 | 5.5095 | 3.4971 | 6.9801 |
| *Strawberry* | 1.877 | 0.1749 | 2.7281 | 3.0472 |

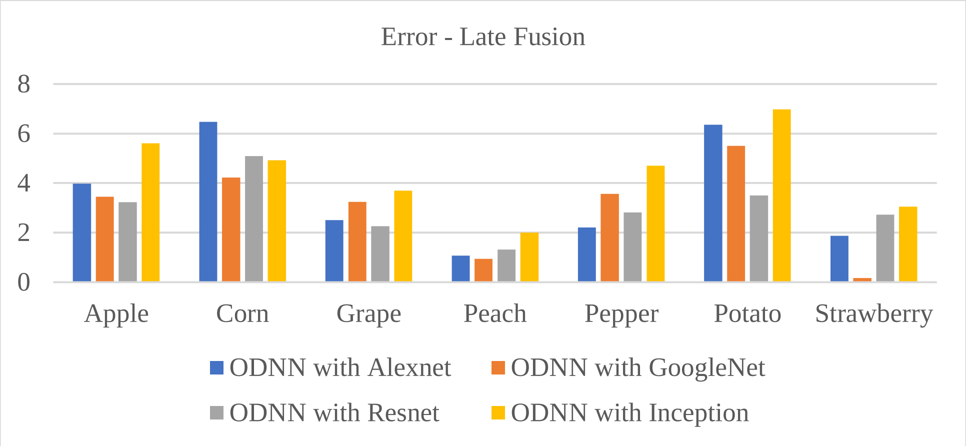


Fig. 16. Error of Late fusion scores chart

The computation of Error is shown in Figure 16. The Figure explains the comparison of fusion. The lowest error found in Strawberry using MODNN-Googlenet. There is an enhancement of about 46.1% in the proposed model in Error.

Table 11. Late fusion accuracy scores.

| *Plant Leaf Category* | *ODNN with Alexnet* | *ODNN with GoogleNet* | *ODNN with Resnet* | *ODNN with Inception* |
|---|---|---|---|---|
| *Apple* | 96.1013 | 96.5556 | 96.7661 | 94.3977 |
| *Corn* | 93.5212 | 95.7703 | 94.9052 | 95.0783 |
| *Grape* | 97.4938 | 96.7555 | 97.7399 | 96.3043 |
| *Peach* | 98.9327 | 99.0581 | 98.6817 | 97.9916 |
| *Pepper* | 97.7868 | 96.4391 | 97.1803 | 95.2936 |
| *Potato* | 93.6392 | 94.4906 | 96.5029 | 93.0199 |
| *Strawberry* | 98.1230 | 99.8251 | 97.2719 | 96.9528 |

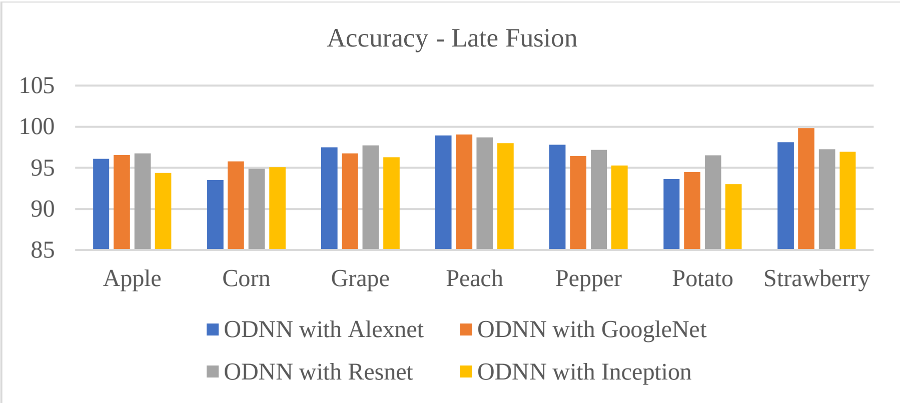


Fig. 17. Accuracy of Late fusion scores chart

Figure 17 and Table 11 shows the late fusion Accuracy computation of the proposed approach.The Figure shows the comparison of late fusion of Optimized DNN with CNN models. The x-axis in the Figure shows the samples, whereas Y-axis shows the Accuracy values. There is an enhancement of 2.14 % in the proposed model in accuracy than standard CNN model.

Table 12. Accuracy Comparative analysis using early and late fusion

| *Plant Leaf Category* | *AlexNet* | *ODNN - Alexnet Early fusion* | *ODNN - Alexnet Late fusion* | *GoogleNet* | *ODNN-Googlenet Early fusion* | *ODNN-Googlenet Late fusion* | *ResNet* | *ODNN - Resnet Early Fusion* | *ODNN - Resnet Late Fusion* | *Inception* | *ODNN - Inception Early Fusion* | *ODNN - Inception Late Fusion* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Apple* | 93.8 | 96.10 | 96.02 | 94.84 | 96.63 | 96.03 | 95.26 | 96.84 | 94.39 | 90.53 | 94.47 | 94.39 |
| *Corn* | 89.62 | 93.59 | 93.52 | 94.12 | 95.84 | 95.77 | 92.39 | 94.98 | 94.91 | 92.73 | 95.15 | 95.08 |
| *Grape* | 96.72 | 97.57 | 97.49 | 95.24 | 96.83 | 96.76 | 97.21 | 97.81 | 97.74 | 94.34 | 96.38 | 96.30 |
| *Peach* | 98.75 | 99.01 | 98.93 | 98.99 | 99.13 | 99.06 | 98.24 | 98.75 | 98.68 | 96.86 | 98.06 | 97.99 |
| *Pepper* | 97.30 | 97.87 | 97.79 | 94.61 | 96.51 | 96.44 | 96.09 | 97.25 | 97.18 | 92.32 | 95.37 | 95.29 |
| *Potato* | 89.01 | 93.711 | 93.64 | 90.71 | 94.56 | 94.49 | 94.74 | 96.57 | 96.50 | 87.77 | 93.09 | 93.02 |
| *Strawberry* | 96.38 | 98.19 | 98.12 | 99.79 | 99.89 | 99.83 | 94.68 | 97.34 | 97.27 | 94.04 | 97.02 | 96.95 |

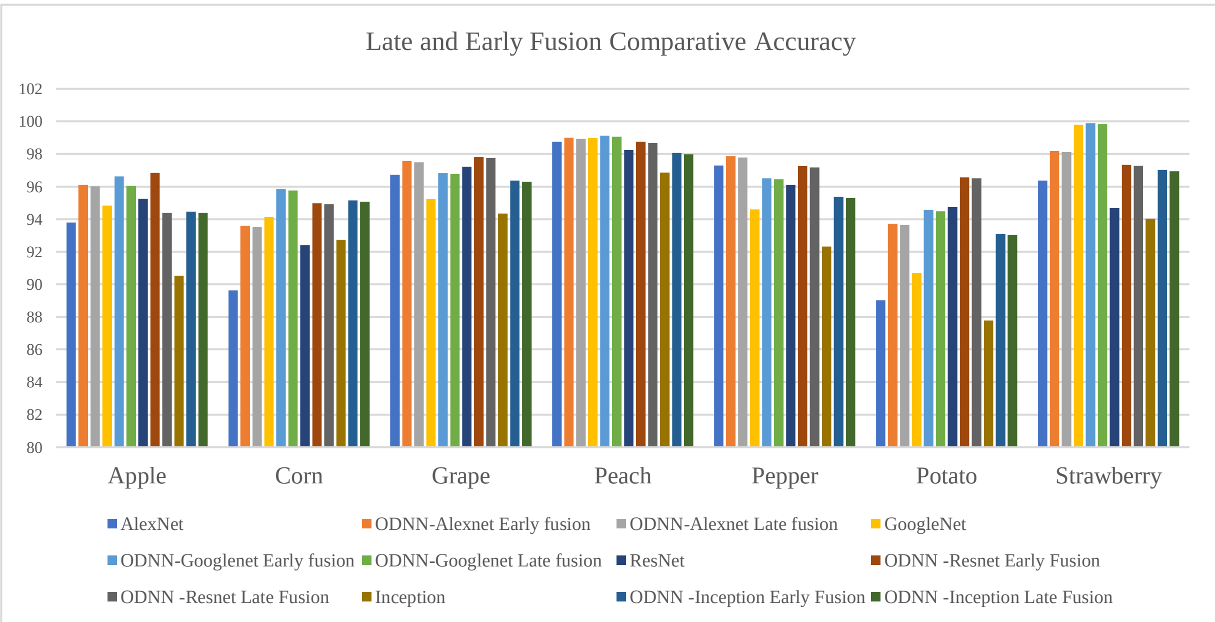


Figure 18. Accuracy analysis of using early and late fusion

Figure 18 and Table 12 is the representation of the comparison of accuracy parameters for proposed article, the standard CNN models and early-late fusion of MODNN with CNN models. The early fusion shows slightly better results the late fusion, there is also an enhancement about 2.18% for both early and late fusion of accuracy than traditional convolutional neural network models.

## 6. Conclusion

In this paper, effective experimental analysis is created. The recognition of plant illness in the early phase is a mandatory for research due to the fact of agricultural troubles to detect the infectious disease and other epidemics. In this article, we have created many models based on the concepts of artificial intelligent to create the best fusion for plant illness using their leaf characteristics. The proposed System has been carried out in several procedures before applying the ODNN integration with CNN as enhancement and segmentation. Using the early and late fusion of classifiers, the accuracy of the proposed System for Plant leaves has developed and enhanced. From the experiments results, it has seen using early fusion is better than late fusion due to early integration of training instances while using the early fusion. Also, it is concluded that the accuracy of the proposed model with ODNN as an optimized DNN increases by approximately 2.18% more from CNN techniques with accurate classification.

Within a short time, we will examine and estimate our system models in other plants, fields and environments with more leaves of several plants and farms trying to build a recognition model based on the multiple classifiers system scheme.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Acknowledgement:** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

**Author Contribution:** All authors contributed to the work. All authors read and approved the final manuscript.

## REFERENCES

1. Chuanlei Z., Shanwen, Z., Jucheng Y.,Yancui S., Jia C., Apple leaf disease identification using the genetic algorithm and correlation-based feature selection method. International Journal of Agricultural and Biological Engineering (2017), 10-2, 74-83.
2. Nachtigall, L. G., Araujo, R. M., Nachtigall, G. R., Classification of apple tree disorders using convolutional neural networks, IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI) (2016) November, 472-476.
3. Liu. B., Zhang, Y., He D., Li Y., Identification of apple leaf diseases based on deep convolutional neural networks, Symmetry (2017), 10-1, 11.
4. Singh, V., Misra, A.K., Detection of plant leaf diseases using image segmentation and soft computing techniques, Information processing in Agriculture (2017), 4-1, 41-49.
5. Sahoo R.K., Panda R., Barik R.C., Panda S.N., Automatic Dead zone detection in 2-D leaf image using clustering and segmentation technique, International Journal of Image, Graphics and Signal Processing (2018), 10-10, 11.
6. Kaur S., Pandey S, Goel S., Plants disease identification and classification through leaf images: A survey, Archives of Computational Methods in Engineering (2019), 26-2, 507-530.
7. Zhang S., Wang H., Huang W., You Z., Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG,, Optik (2018),157, 866-872.
8. Barbedo J.G.A., A new automatic method for disease symptom segmentation in digital photographs of plant leaves, European journal of plant pathology (2017),147-2, 349-364.
9. J. G. A. Barbedo, A review on the main challenges in automatic plant disease identification based on visible range images, Biosystems Engineering (2016), vol. 144, 52-60.

10. V. Pooja, R. Das, V. Kanchana, Identification of plant leaf diseases using image processing techniques" in Technological Innovations in ICT for Agriculture and Rural Development (TIAR), IEEE (2017), 130-133.
11. Kalaivani S., Shantharajah S.P, Padma T., Agricultural leaf blight disease segmentation using indices based histogram intensity segmentation approach, Multimedia Tools and Applications (2019), 1-15.
12. Chouhan S.S., Singh U.P., Jain S., Applications of Computer Vision in Plant Pathology: A Survey, Archives of Computational Methods in Engineering (2019), 1-22.
13. Golhani K., Balasundram S.K., Vadamalai G., Pradhan, B., A review of neural networks in plant disease detection using hyperspectral data, Information Processing in Agriculture (2018), 5(3), 354-371.
14. Salazar Reque I.F., Huamán S.G., Kemper G. Telles J., Diaz D., An Algorithm for Plant Disease Visual Symptom Detection in Digital Images Based on Superpixels, International Journal on Advanced Science, Engineering and Information Technology (2019), 9(1), 194-203.
15. Zhang S., You Z., Wu X., Plant disease leaf image segmentation based on superpixel clustering and EM algorithm, Neural Computing and Applications (2019), 31(2), 1225-1232.
16. Camargo A., Smith J.S., An image-processing based algorithm to automatically identify plant disease visual symptoms, Biosystems engineering (2009), 102(1), 9-21.
17. Shilian Zheng, Peihan Qi, Shichuan Chen, Xiaoniu Yang, Fusion Methods for CNN-Based Automatic Modulation Classification, IEEE Access (2019), volume 7, 66496 - 66504
18. Aizhong Mi, Lei Wang, and Junyan Qi, A Multiple Classifier Fusion Algorithm Using Weighted Decision Templates, Scientific Programming, Hindawi (2016).
19. V. G. Giannoglou and J. B. Theocharis, Decision fusion of multiple classifiers for coronary plaque characterization from IVUS images, International Journal on Artificial Intelligence Tools, vol. 23, no. 3, Article ID 1460005, 2014.
20. S.-Y. Liang, D.-Q. Han, and C.-Z. Han, A novel diversity measure based on geometric relationship and its application to design of multiple classifier systems, Acta Automatica Sinica, vol. 40, no. 3, pp. 449–458, 2014.
21. Kuan Liu, Yanen Li,Ning Xu, Learn to Combine Modalities in Multimodal Deep Learning, MICCAI (2019).
22. Hilal Ergun, Yusuf Caglar Akyuz, Mustafa Sert,Jianquan Liu, Early and Late Level Fusion of Deep Convolutional Neural Networks for Visual Concept Recognition, International Journal of Semantic Computing (2016), Vol. 10, No. 03, 379-397.
23. Jian Guo and Stephen Gould, Deep CNN ensemble with data augmentation for object detection, CoRR (2015), abs/1506.07224.
24. Y-Lan Boureau, Jean Ponce and Yann LeCun, A theoretical analysis of feature pooling in visual recognition, in Proceedings of the 27th International Conference on Machine Learning (2010), pp. 111–118.
25. Herbert Bay, Tinne Tuytelaars and Luc Van Gool, Surf: Speeded up robust features, in Computer Vision–ECCV 2006, 2006, pp. 404–417.
26. Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus and Yann LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, arXiv preprint arXiv:1312.6229, 2013.
27. Alina Roitberg, Tim Pollert, Monica Haurilet, Manuel Martin, Rainer Stiefelhagen, Analysis of Deep Fusion Strategies for Multi-modal Gesture Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop 2019.
28. Yuan Dong, Shan Gao, Kun Tao, Jiqing Liu, Haila Wang, Performance evaluation of early and late fusion methods for generic semantics indexing, Journal of Pattern Analysis and Applications,Springer (2014), Vol. 17(1), 37-50.
29. Juan-Manuel Preez-Rua, Valentin Vielzeuf,Stephane Pateux,Moez Baccouche,Frederic Jurie, MFAS: Multimodal Fusion Architecture Search, IEEE (CVPR 2019).
30. P.K.Atrey,M.A.Hossain,A.ElSaddik,andM.S.Kankan- halli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010.
31. X. Zhou and B. Bhanu. Feature fusion of side face and gait for video-based human identification. *Pattern Recognition*, 2008.
32. Sumair Aziz, Muhammad Awais, Tallha Akram, Umar Khan, Musaed Alhussein, Khursheed Aurangzeb, Automatic Scene Recognition through Acoustic Classification for Behavioral Robotics, Electronics, MDPI (2019).
33. Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, Frédéric Jurie, CentralNet: a Multilayer Approach for Multimodal Fusion, HAL (2018), Id: hal-01858560.
34. Yi Yang, Jingkuan Song, Zi Huang, Zhigang Ma, Nicu Sebe,Alexander G. Hauptmann, Multi-feature fusion via hierarchical regression for multimedia analysis, IEEE Trans. Multimedia (2013),15(3), 572–581.

35. Zhen-zhong Lan, Lei Bao, Shoou-I Yu, Wei Liu, Alexander G. Hauptmann, Double Fusion for Multimedia Event Detection, International Conference on Multimedia Modeling, MMM 2012: Advances in Multimedia Modeling, Springer (2012), 173-185.
36. Yuan Dong, Shan Gao, Kun Tao, Jiqing Liu, Haila Wang, Performance evaluation of early and late fusion methods for generic semantics indexing, Pattern Analysis and Applications, Springer (2014), v.17, 37–50.
37. Eduardo Fonseca, Rong Gong, Dmitry Bogdanov, Olga Slizovskaia, Emilia Gomez, Xavier Serra, Acoustic Scene classification by ensembling gradient boosting machine and convolutional neural network, Detection and Classification of Acoustic Scenes and Events, Germany (2017).
38. Shahrzad Saremi,Seyedehzahra Mirjalili, Seyedali Mirjalili, Jin Song Dong, Grasshopper Optimization Algorithm: Theory, Literature Review, and Application in Hand Posture Estimation, Nature-Inspired Optimizers, Springer (2019), Nature-Inspired Optimizers, Studies in Computational Intelligence, v.811, 107-122.