

## Diagnosis of Leaf Disease Using Enhanced Convolutional Neural Network

Dr. A.Sumaiya Begum<sup>1</sup>, S.Savitha<sup>2</sup>, S.Shahila<sup>3</sup>, S.Sharmila<sup>4</sup>

Department of Electronics and Communication Engineering,

R.M.D Engineering College, Tamil Nadu

Email :asb.ece@rmd.ac.in<sup>1</sup>,savitha18shankar@gmail.com<sup>2</sup>,shahilaselvakumar@gmail.com<sup>3</sup>,  
nishithacharulatha.18@gmail.com<sup>4</sup>

**Abstract** – Despite enormous remote sensing technique available, leaf disease detection continues to be a real concern. This paper Proposes a method that uses computer vision based deep learning techniques to detect 15 categories of Leaf diseases for three types of leaves namely tomato, potato and pepper bell. For the proposed method different deep learning techniques such Modified Mobilenet, Xception, Inception\_Resnet\_v2 has been implemented for feature extraction, training, testing and appropriate detection of the leaf disease. Of all the above methods Enhanced mobilenet proves to be best in the detection of all 15 categories of diseases. The parameters obtained from mobilenet are more promising when compared to Xception, Inception\_Resnet\_v2. Using the state-of-the-art architecture considerable efficiency up to 97 percentage has be achieved.

**Keywords** - Convolutional Neural Network, Machine Learning, Deep learning, Leaf Disease, Disease Diagnosis.

### I. INTRODUCTION

In this paper, enhanced network architecture is proposed for the effective diagnosis of leaf diseases. Effective yield and management of crops in farming fields involves many process like check over plants water level, time to time irrigation, usage of fertilizers and pesticides, continuous monitoring for identifying disease in the early stage. It is found arduous in identifying early plant disease as the farmers lack in knowledge on disease identification and remedial measure to terminate it. Thus, there is a huge need for diagnosing the plant disease at the early stage, such that timely remedial actions can be taken by the farmers to exterminate the disease from further damaging and affecting the other crops .Detection of Plant disease through the observation via naked involves a high degree of complexity. Plant pathologists often fail to positively diagnose specific diseases, and are consequently led to mistaken conclusions and treatments. Convolutional neural network models are used to perform the detection and diagnosis of plant disease using simple leaves images containing both healthy and diseased plants.The growth of world economy is highly dependent on the productivity rate from agriculture. Downfall in productivity is mainly due to global climatic changes and abnormalities. Usage of pesticides by farmers in all the parts of the country, also results in the quality degradation. Hence, a high quality agricultural practice is the need of the day to increase the productivity at a large scale. Also identification of plant disease at an starting stage will help the farmers in coping up the productivity loss in production.

At present the world's population has made a very huge demand and need for all the food products, so in order to meet the highly increasing growth of the world's demand; Farming methods have taken a new step, by using fertilizers for fast effective growth. Recent studies clearly show that the machine learning techniques can be utilized in plant disease identification.

Dheeb Al Bashish et.al used K-means segmentation algorithm for recognizing the disease infection in plant leaves with an accuracy of 93%. With advancement in the technology) and the extensive availability of deep learning frameworks and enhancement in computer vision aids in solving numerous challenges.

Convolutional neural network is similar to the exact biological network. Like the neurons in the neural system they pass the data from one neuron to another

### II. RELATED WORKS

There are enormous number of techniques available for the diagnosis of leaf diseases in plants. Machine Learning has plays an important role in it

An automated leaf disease detection system from visual appearance of the leaf is of great advantage. Several such techniques have been used and implemented for leaf disease detection[1]-[6]. Song et al utilized a SVM classifier to classify and detect a variety of maize leaf diseases.[1]. The SVM technique is suited for small database systems, while they fail to generate appreciable results for large database systems. Recently deep learning techniques have shown improvement to a greater extent in the detection accuracy [7]–[12].

Zhang et al., improved the accuracy level by combing quantum neural network along with the characteristic parameter[6]. Several image processing techniques were implemented for plant disease detection [13-17], Zhou et al achieved an accuracy Of 95% by feature extraction using Particle swarm optimization [18].Rumpf et al used support vector machine algorithms for diagnosis of sugarbeet diseases. However, depending upon the severity of disease a low accuracy level of 65 to 90% was achieved in these techniques [19]. An model for feature extraction and neural network was implemented by certain authors for leaf disease detection. This ensemble improves the learning ability of the neural network.[20-21].

An ANN architecture with 10 hidden layers and an output detecting five classes was implemented by Al- Hiary H et al., .A K-means clustering technique was proposed by the authors to generate an

accuracy of 94%. An immense growth in convolutional neural network architectures led to a wide range of research in plant leaf disease detection. Several researchers have implemented different convolutional neural network architectures as a remedial measure.[23-26].

### III. CONVOLUTIONAL NEURAL NETWORK ALGORITHM

There are three different network architecture used for the diagnosis of leaf diseases namely Xception, Inception ResNet V2, Mobilenet.

MobileNets architecture are based on streamlined architectures that uses depth wise separable convolutions to build a light weight and enhanced neural networks.

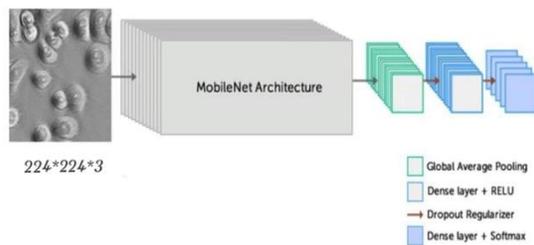


Fig.1. Mobilenet architecture

Table I Parameters of Mobilenet

Type/Stride	Filter Shape	Input Size
Conv/s2	3x3x3x32	224x224x3
Conv dw/s1	3x3x32dw	112x112x32
Conv/s1	1x1x32x64	112x112x32
Conv dw/s2	3x3x64dw	112x112x64
Conv/s1	1x1x64x128	56x56x64
Conv dw/s1	3x3x128dw	56x56x128
Conv/s1	1x1x128x128	56x56x128
Conv dw/s2	3x3x128dw	56x56x128
Conv/s1	1x1x128x256	28x28x128
Conv dw/s1	3x3x256dw	28x28x256
Conv/s1	1x1x256x256	28x28x256
Conv dw/s2	3x3x256dw	28x28x256
Conv/s1	1x1x256x512	14x14x256
5xConvdw/s1	3x3x512dw	14x14x512
Conv/s1	1x1x512x512	14x14x512
Conv dw/s2	3x3x512dw	14x14x512
Conv/s1	1x1x512x1024	7x7x512
Conv dw/s2	3x3x1024dw	7x7x1024
Conv/s1	1x1x1024x1024	7x7x1024
Avg Pool/s1	Pool 7 x 7	7x7x1024
FC/s1	1024 x 1000	1x1x1024
Softmax/s1	classifier	1x1x1000

Depth-wise separable improved convolution technique in which factorized convolution is performed. In Mobilenet architecture the depth-wise convolution provides one layer to each input channel. Actually, standard convolution combines both filters and input to obtain output. The DSC splits into couple of layers (The effect of factorization of standard convolution reduces computational time and model size).

Depth wise convolutional are made up of two layers

1. Depth-wise convolution
2. Point-wise convolution

As specified before depth wise convolution provides a one layer per input channel. Point wise convolution aids in creation of a linear combination of the output. MobileNets uses batch normalization and relu non- linearity function for both layers. Depth wise convolution is efficient related to standard convolution. It filters only input channel.. The joining of depth-wise convolution and point wise convolution is known as Depth-wise separable convolution.

Inception ResNet v2 architecture is a combination of inception and resnet structures. The building module of inception architecture is inception block. There are many filter dimension in each block and each type of Kernel is of various shape, output of the block joins the output from different kernels. This improves the diversity in terms of width. Therefore the model improved its recognition performance for objects with different sizes. The Inception architecture is greatly tunable it refers to changing the filters amount does not actually affect the network.

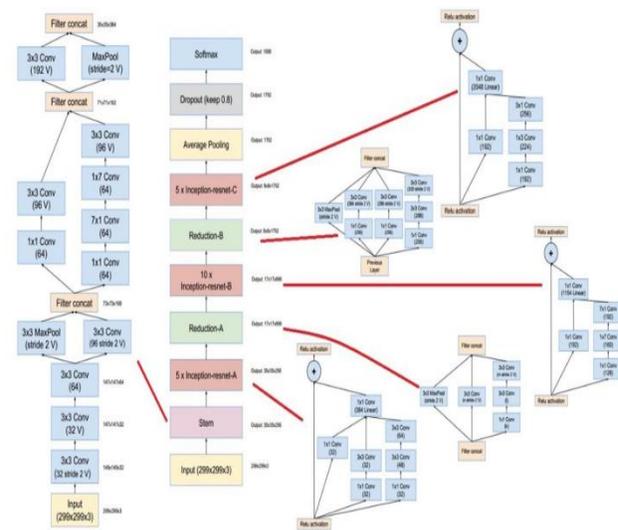


Fig.2. Inception-Resnet v2 architecture

Fig.4. Dataset

Xception is the enhanced outlook of Inception network. It is improved network architecture. The number of neural network connections is fewer and so the model is lighter. The Xception architecture is a linear collection of depth-wise separable convolution layers. This makes the architecture enhanced and highly effective. Depth-wise convolution is the channel-wise  $n \times n$  spatial convolution. If we have 5 channels, then we will have  $5 n \times n$  spatial convolutions. Point-wise convolution actually is the  $1 \times 1$  convolution to change the dimension. The architecture is clearly depicted below.

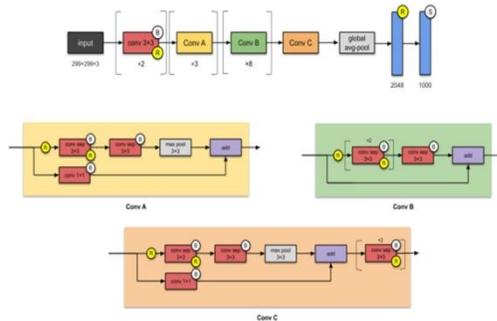


Fig .3 Xception architecture

#### IV. PROCESS IMPLEMENTATION

The whole method for detecting leaf disease using machine learning is described clearly. The entire method for detection is segmented as stages starting from dataset collection to detection of diseases. Plant disease diagnosis involves a series of steps right from the data collection from various sources. Then pre-processing it according to the input required for the network. In this methodology effective network architecture is used to improve the accuracy via transfer learning methodology. The image features are obtained from the image and then a classifier is used to detect the disease. In this project, we are using the dataset from the plant village consisting of 20,000 images which includes both healthy and diseased plant.

##### i. Leaf Dataset



The leaf data collection for our project is collected from Plant Village dataset. It is obtained from a site called Kaggle. The dataset comprises of different types of leaf images of healthy and diseased plants. The entire dataset is provided in Fig 4.

##### ii. Image preprocessing and labeling

Images downloaded from the internet were in various formats along with perfect quality. In order to obtain better feature vector all images must be used as dataset and so it achieves better consistency. Next stage of process is processing the images to obtain an ideal image. During phase of collecting the images for the dataset, images with poor resolution and smaller dimensions were not considered as ideal images for dataset. So, only the images with higher resolution and dimension are considered for the process. Images used for the dataset were resized as  $224 \times 224 \times 3$  to reduce training time, which was computed in python, using tensor flow framework. The images are resized to  $299 \times 299 \times 3$  for the Xception and Inception Resnet V2.

##### iii. Transfer learning:

Transfer learning is an effective machine learning technique where a model developed for a goal is reused as the starting point for a model on an another goal. In transfer learning, you can use knowledge such as features, weights from previously trained models for training newer models.

Deep learning systems and models are multi layered and advanced architectures that learn different features at different layers. It consumes much of the time of the training process and provides very effective results.

The benefits of Transfer Learning is has the ability to pace up the time it takes to develop and train a model by again reusing these modules of developed models. This helps accelerates and enhances the model training process.

##### iv. Feature extraction:

Feature Extraction is a process of translation of raw data into a form that particular machine learning algorithm needs. Features are extracted from the input of the image via the keras. The features must represent the information of data in a format that will best fit the needs of algorithm that is going to be used to solve the problems.

The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important information. Feature Extraction can also reduce the amount of redundant data for given analysis. Also the reduction of the data and the machine's efforts in building variable combinations features facilitate the speed of learning and generalization steps in machine learning process

v. *Training and classification:*

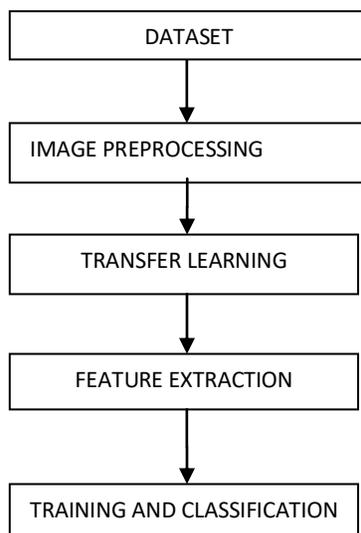
Classifier is made up of fully connected neural layers. The aim of the classifier is to classify images based on detected feature vectors. Logistic Regression is the machine learning classifier. Scikit-learn's Logistic Regression offers a number of techniques for training a logistic regression, called solvers.

```
model = LogisticRegression(max_iter=10,solver="sag")
```

Stochastic average gradient descent is technique that allows us to train a model much greater speed than other solvers when our data is very huge. However, it is also very sensitive to the factor of feature scaling so standardizing our features is particularly important. We can set our learning algorithm to use this particular solver by setting solver='sag'.

Logistic regression is most important supervised learning classification method. It is a fast, versatile extension of a generalized linear model. From the above code solver is to find the parameter weights to minimize a cost function is called solver function. Sag is stochastic average Gradient descent. A variation of gradient approaches uses a random sample of previous gradient values for fast big dataset. 'Sag' solver is used in logistic regression operator for large dataset to speed up pipeline evaluation.

After completing the training process via the logistic regression classifier, the Rank 1 and Rank 5 and the predictions are made by taking the probabilities of each label. Followed by that process is the conversion of accuracies into percentages via the coding. The model is then evaluated with the test data and it is also programmed to dump the classification report into a file (Document). Finally the confusion matrix is obtained for the support data. Several classification metrics' like precision, Recall, F1 score could be inferred from the confusion matrix.



**TABLE II:** Number of classes used for the model

Class no	Classes	Training data	Testing data
0	Pepper bell Bacterial spot	910	87
1	Pepper bell healthy	1351	127
2	Potato Early blight	895	105
3	Potato healthy	55	97
4	Potato Late blight	987	13
5	Tomato Target Spot	1187	217
6	Tomato Tomato mosaic virus	279	94
7	Tomato Tomato Yellow Leaf Curl Virus	3023	186
8	Tomato Bacterial spot	2032	95
9	Tomato Early blight	810	190
10	Tomato healthy	1451	162
11	Tomato Late blight	1769	140
12	Tomato Leaf Mold	603	349
13	Tomato Septoria leaf spot	1731	40
14	Tomato two spotted spider mite	1514	162
	Total	18597	2064

Table II cited above shows the total number of classes along with their data name and number of data (image) in each class .The table has both training and testing data .

V. EVALUATION METRICS

Confusion matrix:

Confusion matrix is a tabular visualization of the model predictions versus the actual value of the class. The vertical row with class name denotes the actual class whereas the horizontal row represents the predicted class by the model. The diagonal elements represent the number of places in which the predicted value is same as the actual value, while off-diagonal elements are those which represents the wrong predictions. Greater the diagonal values of the confusion matrix the better, indicates much correct prediction.

- Positive (P): Observation is positive (for example: Healthy).
- Negative (N): Observation is not positive (for example: diseased).
- True Positive (TP) — if the healthy class is predicted as healthy then it is true positive.
- True Negative (TN) — if a diseased class is predicted as healthy then it is true negative.
- False Positive (FP) — if the healthy class is predicted as a diseased then it is false positive.
- False Negative (FN) — if diseased class is predicted as a diseased then it is false negative.

Let us consider a confusion matrix for better clarity,

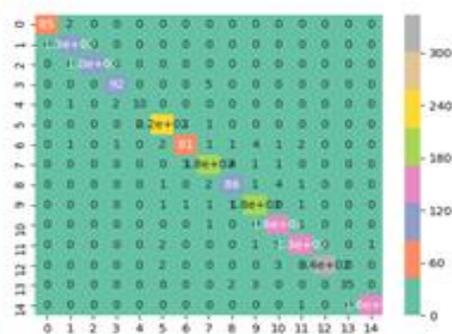


Fig. 5. Confusion matrix

Using the matrix above and considering the values in the vertical axis as the actual class, and the values in the horizontal axis the prediction. Then for the Class 0:

- **True Positive** = 85 -> samples of class 0, classified as class 0
- **False Positive** = 0 -> samples of classes 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 but classified as class 0
- **False Negative** = 2 -> samples of class 0, but classified as classes 1
- **True Negative** = The sum of all the values in the matrix except those in column 0 and row 0

**PRECISION:**

Precision is also called as positive prediction value. It quantifies the number of positive class predictions that actually belongs to the positive class. Precision also refers to the closeness of two or more measurements to each other.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{1}$$

**RECALL:**

Recall is also known as sensitivity. It is of estimate of correctly predicted positive entities to the all actual entities in the actual class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{2}$$

**F1-SCORE:**

F1-score is a weighted harmonic mean between precision and recall.

$$\text{F1-SCORE} = 2 \frac{(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad \text{3}$$

**SUPPORT:**

Support is the number of actual occurrences of the class in the specified dataset.

$$\text{Support} = \frac{(\text{Precision} + \text{Recall})}{2} \quad \text{3}$$

VI. RESULTS

i. XCEPTION:

The fig 6 depicts the confusion matrix for xception architecture. It was obtained during the testing of 10% of the images from each category of the entire dataset.

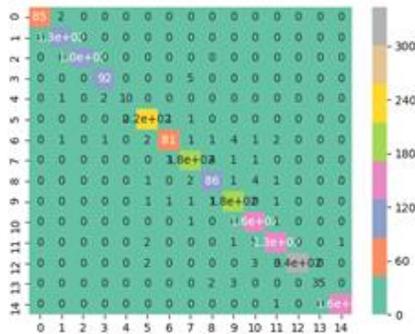


Fig .6. Confusion matrix for Xception

The table III depicts the result obtained when the dataset is tested for the feature extraction by Xception architecture.

**Table III Results for Xception**

<b>Rank 1 : 93.84%</b>				
<b>Rank 5 : 99.95%</b>				
Classes	Precision	Recall	F1-Score	Support
0	0.96	0.98	0.97	87
1	0.98	0.98	0.98	127
2	0.98	0.98	0.98	105
3	0.94	0.91	0.92	97
4	0.91	0.77	0.83	13
5	0.95	0.98	0.97	217
6	0.84	0.68	0.75	94
7	0.90	0.92	0.91	186
8	0.94	0.94	0.94	95
9	0.91	0.95	0.93	190
10	0.87	0.95	0.91	162
11	0.90	0.86	0.88	140
12	0.99	0.97	0.98	349
13	0.99	0.85	0.88	40
14	0.92	0.99	0.99	162
Accuracy			0.94	2064

ii. INCEPTION\_RESNET\_v2:

The fig 7 depicts the confusion matrix for Inception\_Resnet\_v2 architecture. It was obtained during the testing of 10% of the images from each category of the entire dataset.

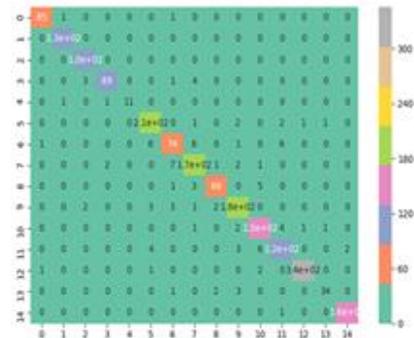


Fig. 7. Confusion matrix for Inception\_Resnet\_v2

The table IV depicts the result obtained when the dataset is tested for the feature extraction by Inception\_Resnet\_v2 architecture.

**Table IV Results for Inception\_Resnet\_v2**

<b>Rank 1 : 94.82%</b>				
<b>Rank 5 : 99.81%</b>				
Classes	Precision	Recall	F1-Score	Support
0	0.98	0.98	0.98	87
1	0.98	1.00	0.99	127
2	0.95	1.00	0.98	105
3	0.97	0.92	0.94	97
4	1.00	0.85	0.92	13
5	0.94	0.97	0.95	217
6	0.84	0.79	0.81	94
7	0.92	0.93	0.92	186
8	0.95	0.91	0.92	95
9	0.93	0.94	0.94	190
10	0.92	0.94	0.93	162
11	0.91	0.89	0.90	140
12	0.99	0.99	0.99	349
13	0.94	0.85	0.89	40
14	0.99	0.99	0.99	162
Average			0.95	2064

## i. MOBILENET :

The fig 8 depicts the confusion matrix for Mobilenet architecture. It was obtained during the testing of 10% of the images from each category of the entire dataset.

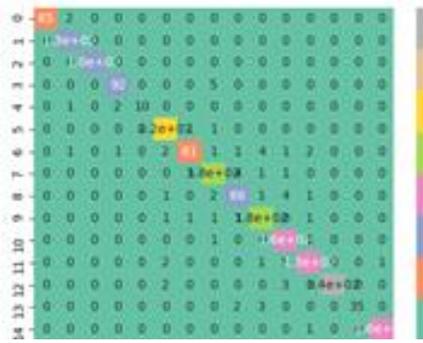


Fig. 8. Confusion matrix for Mobilenet

Table V Results for Mobilenet

Rank 1	96.61%			
Rank 5	99.95%			
S.NO	Precision	Recall	F1-score	Support
0	1.00	0.98	0.99	87
1	0.97	1.00	0.98	127
2	1.00	1.00	1.00	105
3	0.97	0.95	0.96	97
4	1.00	0.77	0.97	13
5	0.96	0.99	0.98	217
6	0.94	0.86	0.90	94
7	0.94	0.95	0.95	186
8	0.91	0.91	0.91	0.95
9	0.95	0.97	0.96	190
10	0.92	0.99	0.95	162
11	0.96	0.94	0.95	140
12	1.00	0.99	0.99	349
13	1.00	0.88	0.93	40
14	0.99	0.99	0.99	162
Accuracy			0.97	2064

The table V depicts the result obtained when the dataset is tested for the feature extraction by Inception\_Resnet\_v2 architecture. Then the results obtained from the three different classifiers are actually compared to find the effective model among all the models,

TABLE VI COMPARISON TABLE

Architecture	Rank 1	Rank 5	Average accuracy
Mobilenet	96.61%	99.95%	97%
Inception Resnet V2	94.82%	99.81%	95%
Xception	93.85%	99.95%	94%

In the above table VI, it depicts the performance analysis between the three different architectural models. It is observed that the Mobilenet network perform well than the other two convolutional neural. The reason is that, its architecture makes it extract more features resulting in better prediction results.

❖ **Mobilenet >Inception\_Resnet\_V2>Xception**

❖ **97% > 95% > 94%**

## VII.CONCLUSION

In this project, the performance of three different types of CNN image classifier, which is built around Mobilenet, Inception\_Resnet\_v2 and Xception model is proposed. It is observed that the Mobilenet model identifies and classifies the images more accurately than the Inception\_Resnet\_v2 model and Xception model, when tested on KAGGLE image library datasets. Mobilenet architecture is composed of several inception modules packed together, which makes it denser and informative. The feature extractor performs well with modules. This technique has outperformed the current accuracy obtained so far. In future, this model can further be modified to include several layers that would complement the output accuracy. It can also be trained on large datasets of more than 100 classes with some modification. The model would be very much beneficial in real time application especially in agricultural field to explore and classify various untouched field where there is a possibility of large amount of human error in evaluation and identification of various types of leaf diseases

## VIII. REFERENCES:

- [1] K. Song, X. Y. Sun, and J. W. Ji, "Corn leaf disease recognition based on support vector machine method," *Trans. Chin. Soc. Agricult. Eng.*, vol. 23, no. 1, pp. 155-157, Jan. 2007.
- [2] L. Chen and L. Y. Wang, "Research on application of probability neural network in maize leaf disease identification," *J. Agricult. Mech. Res.*, vol. 33, no. 6, pp. 145- 148, Jun. 2011.
- [3] L. F. Xu, X. B. Xu, and H. Min, "Corn leaf disease identification

- based on multiple classifiers fusion,” *Trans. Chin. Soc. Agricult. Eng.*, vol. 31, no. 14, pp. 194–201, 2015.
- [4] N. Wang, K. Wang, R. Xie, J. Lai, B. Ming, and S. Li, “Maize leaf disease identification based on fisher discrimination analysis,” *Scientia Agricultura Sinica*, vol. 42, no. 11, pp. 3836–3842, 2009.
- [5] Z. Qi et al., “Identification of maize leaf diseases based on image technology,” *J. Anhui Agricult. Univ.*, vol. 43, no. 2, pp. 325–330, Feb. 2016.
- [6] F. Zhang, “Recognition of corn leaf disease based on quantum neural network and combination characteristic parameter,” *J. Southern Agriculture*, vol. 44, no. 8, pp. 1286–1290, 2013.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.
- [8] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.
- [9] Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [10] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.
- [11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
- [12] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
- [13] Reddy P. R., Divya S. N., Vijayalakshmi R. Plant disease detection technique tool—a theoretical approach. *International Journal of Innovative Technology and Research*. 2015:91–93. [Google Scholar]
- [14] Mahlein A.-K., Rumpf T., Welke P., et al. Development of spectral indices for detecting and identifying plant diseases. *Remote Sensing of Environment*. 2013;128:21–30. doi: 10.1016/j.rse.2012.09.019.
- [15] Xiuqing W., Haiyan W., Shifeng Y. Plant disease detection based on near-field acoustic holography. *Transactions of the Chinese Society for Agricultural Machinery*. 2014;2.
- [16] Mahlein A.-K., Oerke E.-C., Steiner U., Dehne H.-W. Recent advances in sensing plant diseases for precision crop protection. *European Journal of Plant Pathology*. 2012;133(1):197–209. doi: 10.1007/s10658-011-9878-z
- [17]. Reed T. R., Dubuf J. M. H. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*. 1993;57(3):359–372. doi: 10.1006/ciun.1993.1024.
- [18]. Zhou C., Gao H. B., Gao L., Zhang W. G. Particle swarm optimization (PSO) algorithm. *Application Research of Computers*. 2003;12:7–11.
- [19]. Rumpf T., Mahlein A.-K., Steiner U., Oerke E.-C., Dehne H.-W., Plümer L. Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Computers and Electronics in Agriculture*. 2010;74(1):91–99. doi: 10.1016/j.compag.2010.06.009.
- [20]. Zhou Z. H., Chen S. F. Neural network ensemble. *Chinese Journal of Computers*. 2002;25(1):1–8. [Google Scholar]
- [21]. Karmokar B. C., Ullah M. S., Siddiquee Md. K., Alam K. Md. R. Tea leaf diseases recognition using neural network ensemble. *International Journal of Computer Applications*. 2015;114(17):27–30.
- [22]. Al-Hiary H., Bani-Ahmad S., Reyalat M., Braik M., AL-Rahamneh Z. Fast and accurate detection and classification of plant diseases. *Machine Learning*. 2011;14:p. 5.
- [23]. Lenz I., Lee H., Saxena A. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*. 2015;34(4-5):705–724. doi:10.1177/0278364914549607.
- [24]. Alipanahi B., DeLong A., Weirauch M. T., Frey B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*. 2015;33(8):831–838. doi: 10.1038/nbt.3300.
- [25]. Zhang L., Xia G.-S., Wu T., Lin L., Tai X. C. Deep learning for remote sensing image understanding. *Journal of Sensors*. 2016;2016:2. doi: 10.1155/2016/7954154.7954154
- [26]. Arevalo J., Gonzalez F. A., Ramos-Pollan R., Oliveira J. L., Lopez M. A. G. Convolutional neural networks for mammography mass lesion classification. *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medical and Biology Society (EMBC’15)*; August 2015; pp. 797–800.