

PAPER • OPEN ACCESS

## Identification of Grape Leaf Diseases Using Convolutional Neural Network

To cite this article: Moh. Arie Hasan *et al* 2020 *J. Phys.: Conf. Ser.* **1641** 012007

View the [article online](#) for updates and enhancements.



**240th ECS Meeting** ORLANDO, FL

Orange County Convention Center Oct 10-14, 2021

Abstract submission deadline extended: April 23rd

**SUBMIT NOW**

# Identification of Grape Leaf Diseases Using Convolutional Neural Network

Moh. Arie Hasan<sup>1\*</sup>, Dwiza Riana<sup>2\*</sup>, Sigit Swasono<sup>3</sup>, Ade Priyatna<sup>4</sup>,  
Eni Pudjiarti<sup>5</sup>, Lusa Indah Prahartiwi<sup>6</sup>

Program Studi Ilmu Komputer STMIK Nusa Mandiri<sup>1-6</sup>

E-mail: 14002250@nusamandiri.ac.id, dwiza@nusamandiri.ac.id

**Abstract.** The presence of leaf diseases in grapes can reduce the productivity of grapes and result in losses for farmers. Leaf diseases are mainly caused by bacteria, fungi, virus etc. A proper diagnosis of disease in plants is needed in order to take appropriate control measures. This paper aims to assist in the identification and classification of grape leaf diseases Convolutional Neural Network (CNN). CNN is basically an artificial neural network architecture that requires repeated training processes to get good accuracy. CNN consists of 3 stages, namely Data Input, Feature Learning, and Classification. The implementation of CNN in this study uses Keras libraries that use the python programming language. Keras is a framework created to facilitate learning of computers. The CNN training process using 0.0001 learning rate obtained results with an accuracy rate of 91,37%

## 1 Introduction

Grapes are fruits originating from Europe and Western Asia. This fruit can be made into juice, dried into raisins or fermented into grapes and brandy. Besides grapes can also cure many diseases because it has medicinal properties. Grapes are an important fruit crop. The presence of diseases in grapes can result in losses for farmers. Identification of grape leaf disease is needed by farmers so that solutions can be found to avoid greater losses during harvest. Diseases of grape leaves such as powdery mildew and downy mildew can cause great losses[?]. Diseases in plants are found in the leaves, fruits and on the stems of plants. Early detection of leaf diseases is a major challenge in agriculture [2].

Deep Learning has evolved in recent years and achieved remarkable results in computer vision. Wen et al. [3]–[5] succeeded in significantly improving face recognition by using the discriminative feature learning approach. Zhang et al. [5] use deep learning for remote sensing image understanding. Guo et al. [6] presents a comprehensive review of deep learning and summarizes about the design and training of deep neural networks. Lu et al. [6] using various operations with convolutional neural network (CNN) model to identify 10 rice diseases with an accuracy of 95.48%. Dechant et al. [7] identified diseases in northern corn leaves using CNN.

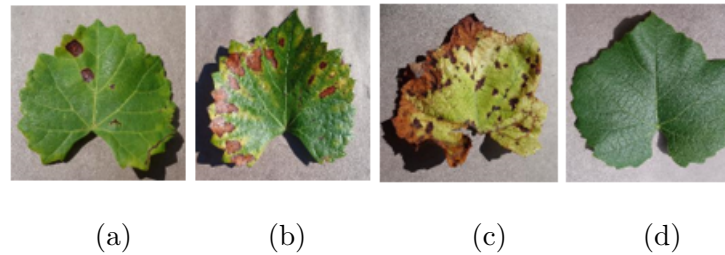
This study aims to identify grape leaf disease making it easier for farmers to identify vines and to deal with grape disease more quickly. Other studies applied image analysis and backpropagation neural network (BPNN) to identify grape diseases [8], SVM classifier to detect grape leaf disease [2], and using machine learning techniques to identify grape leaf diseases [9]. By using CNN it is expected to know the level of accuracy that results from.



## 2 Material and Method

### 2.1 Dataset

The dataset used in this study was 1000 grape leaf images consisting of 800 images for training data and 200 images for testing data are collected from Kaggle [10]. This dataset is needed starting from the object recognition stage, training, and testing. The dataset is divided into four classes, three classes represent infected grape leaves namely Black Rot, Esca, Leaf Blight and one class represents healthy leaves. Four types of grape leaves are shown in Fig. 1 :



**Figure 1.** Black Rot (a), Esca (b), Leaf Blight (c), and Health (d)

**Table 1.** Image Dataset of Grape Leaf

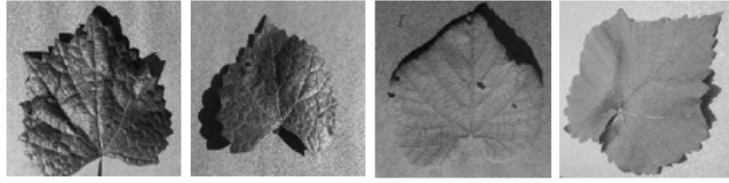
Class	Number of Images	Training Images	Testing Images
Black Rot	250	200	50
Esca	250	200	50
Leaf Blight	250	200	50
Health	250	200	50

### 2.2 Image Preprocessing And Labelling

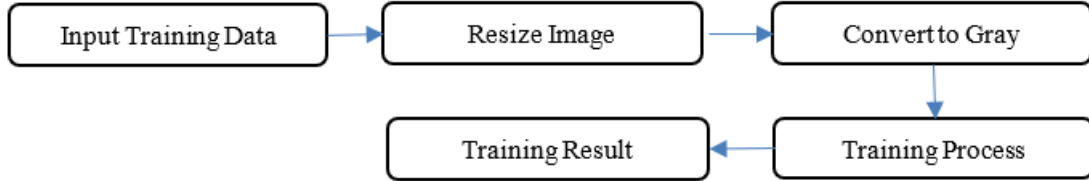
Before the model is trained, image preprocessing is done to improve image consistency in the dataset for the CNN classifier to be processed. The first step to take is to normalize the image size. Images were resized to  $200 \times 200$  pixels using a python script based on the Python Imaging Library (PIL). Then the image is changed to grayscale. The next step is to group the grape leaf images according to their categories and then label all images with the corresponding disease acronym. At this stage, four classes in the dataset in the training set and test set are marked.

### 2.3 Training Dataset

Dataset training is an initial step that aims to process the available dataset. In this training process, image data input will go through the training process using the Convolutional Neural Network that will form a model the performance will be tested later.



**Figure 2.** Four grape leaf image after normalization



**Figure 3.** Training process flow

#### 2.4 Convolutional Neural Network

Convolutional Neural Network is one type of neural network that is usually used in the image data processing. Convolution or commonly referred to as convolution is a matrix that has a function filter the image [11]. Convolutional Neural Network has some layers that are used to filter every process. The process is called the training process. In the training process, there are 3 stages namely the convolutional layer, the pooling layer, and the fully connected layer.

##### 2.4.1 Convolutional Layer

All data that touches the layer convolutional will experience the process convolution. the layer will convert each filter to all parts of the input data and resulting in an activation map or 2D feature map. The filter contained in Convolutional Layer has length, height (pixels), and thickness according to the channel input data. Each filter will experience a shift and "dot" operations between data input and value from the filter. Layer convolutional significantly experienced the complexity of the model through optimization of the output. This is optimized through three parameters, depth, stride, and zero settings padding [12]. On CNN, the most important operation is convolution. Convolution can perform calculations from two-dimensional images that will be mapped to a continuous sliding convolution window to get the corresponding convolution value. Each map feature is convoluted by several graphical input features. For input  $x$  from the  $i$ -th convolutional layer, calculated as follows :

$$h_{ic} = f(W_i * x), \quad (1)$$

where  $f$  is an activation function,  $*$  is a convolution operation, and  $W$  is the convolution kernel at the layer.  $W_i = [W_{i1}, W_{i2}, \dots, W_{iK}]$ ,  $K$  is the number of kernel convolution at the layer. Each  $W_{iK}$  kernel is a weight matrix  $M \times M \times N$ .  $M$  as the window size and  $N$  is the number of input channels [13].

##### 2.4.2 Pooling Layer

The activation map contains many features that can cause overfitting and computing loads. The pooling layer is often implemented to reduce dimensions and can help get invariant for translation. Pooling methods commonly used are max pooling (MP) and average pooling (AP) [14]. Max pooling is to choose the highest value from the pooling area while the average pooling is to calculate the average value of elements in each pooling area. For example region  $R$  contains pixel max pooling, and average pooling can be defined as follows :

$$MP : \{y_j = \max_{i \in R_j} x_i\} \quad (2)$$

$$AP : \{y_j = x_i / \sum_{i \in R_j} x_i\} \quad (3)$$

At kernel size equal to 2 and step equals 2. The maximum pool finally displays the maximum value of the four quadrants, while average pooling produces an average value.

#### 2.4.3 Fully Connected Layer

At the fully connected layer, each node is connected to each node in the previous and subsequent layers directly. Each node in the last frame in the pooling layer is connected as a vector to the first layer of the fully connected layer. This is the parameter most widely used with CNN in these layers but takes a long time in training because it requires complex computing [15]. This is the main weakness of the fully connected layer. Therefore, to overcome these weaknesses can be done by eliminating the number of nodes and connections. Deleted nodes and connections can be satisfied by using the dropout technique.

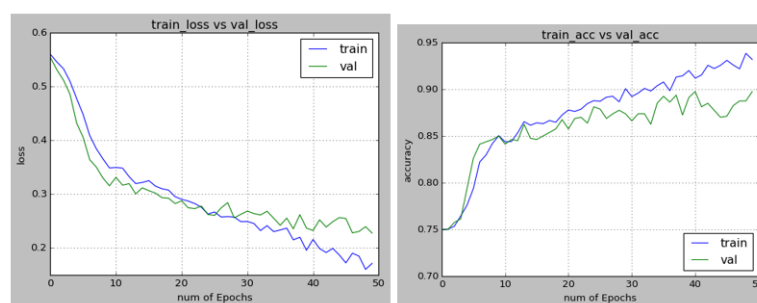
#### 2.4.4 Dropout

Dropout is a business to prevent overfitting as well as accelerate the learning process [16]. Overfitting is a condition where almost all data is has gone through the training process reached a good percentage, but it happens a discrepancy in the prediction process. In the system works, Dropout removes while a neuron in the form of hidden layers and visible layers are located in the network

### 3 Result and Discussion

#### 3.1 Testing the Number of Epochs

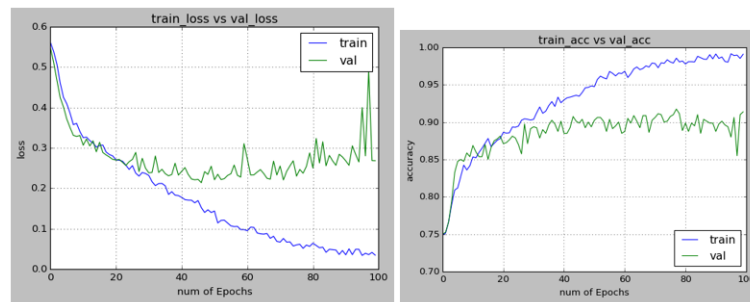
Tests carried out to determine the effect of the amount epochs used on system performance. An epoch is one complete presentation of the data set to be erudite to a learning machine. For comparison, this test uses the number of epoch 50 and epoch 100, whereas learning rates 0,0001. The test results can be seen in Figure 4.



**Figure 4.** Training Loss and Accuracy with Epoch 50 and Learning rate 0,0001

Based on Figure 4, it can be seen that using the training step 50 epoch and the learning rate of 0.0001 produces an accuracy rate of 89.75%.

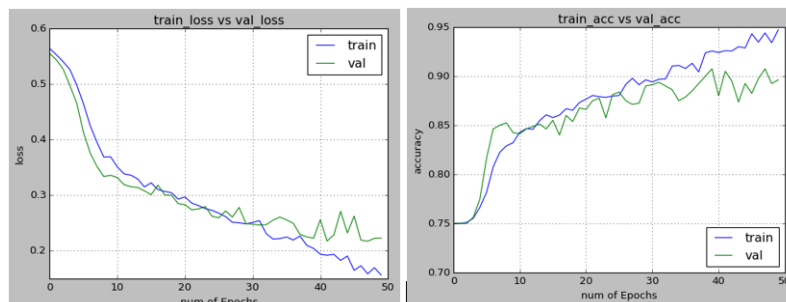
Based on Figure 5, it can be seen that using the training step 100 epoch and the learning rate of 0.0001 produces an accuracy rate of 91.37%. It can be concluded that based on the testing process carried out it can be analyzed that with more number of epochs can produce a better percentage of data accuracy. However, it is getting longer the number of epochs, the more time needed for the training process.



**Figure 5.** Training Loss and Accuracy with Epoch 100 and Learning rate 0,0001

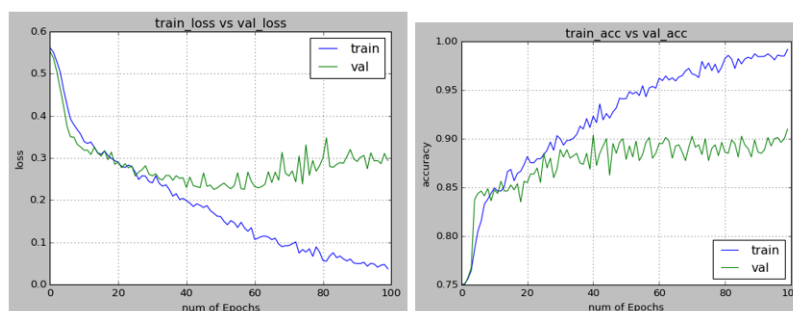
### 3.2 Testing the Number of Learning Rate

Tests carried out to determine the effect of the amount learning rate used on system performance. Learning rate is one of the training parameters to calculate the weight correction value during the training process, the learning rate is in the range between zero (0) to one (1). For comparison, this test uses the number of epoch 50 and epoch 100, whereas learning rates 0,001 and 0.01. The test results can be seen in Figure 6.



**Figure 6.** Training Loss and Accuracy with Epoch 50 and Learning rate 0,001

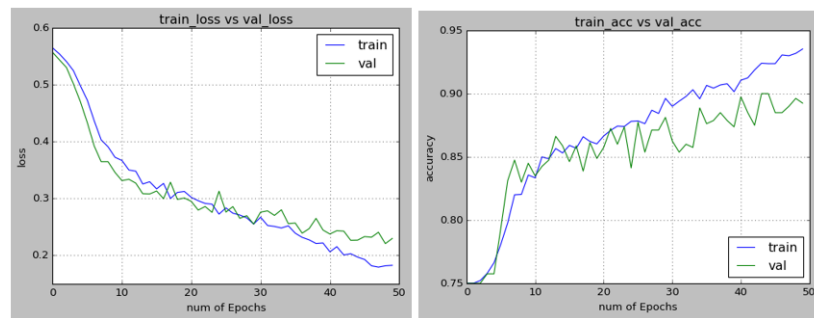
Based on Figure 6, it can be seen that using the training step 50 epoch and the learning rate of 0.001 produces an accuracy rate of 89.62%.



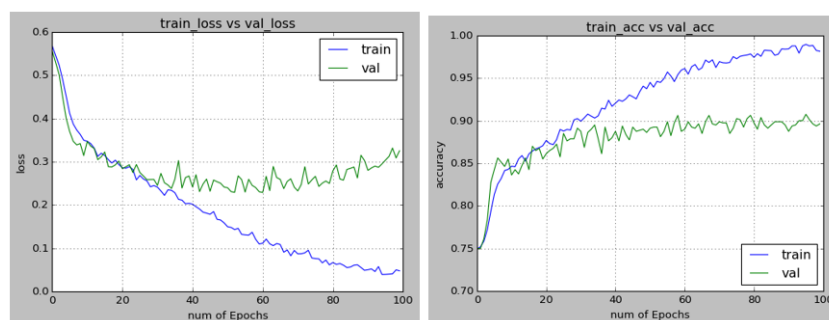
**Figure 7.** Training Loss and Accuracy with Epoch 100 and Learning rate 0,001

Based on Figure 7, it can be seen that using the training step 100 epoch and the learning rate of 0.001 produces an accuracy rate of 91.00%.

Based on Figure 8, it can be seen that using the training step 50 epoch and the learning rate of 0.01 produces an accuracy rate of 89.25%



**Figure 8.** Training Loss and Accuracy with Epoch 50 and Learning rate 0,01



**Figure 9.** Training Loss and Accuracy with Epoch 100 and Learning rate 0,01

Based on Figure 9, it can be seen that using the training step 100 epoch and the learning rate of 0.01 produces an accuracy rate of 89.62%. It can be concluded that based on the testing process carried out it can be analyzed that with more number of learning rate can produce a better percentage of data accuracy.

**Table 2.** Test Result

Dataset Amount	Image Size	Epoch	Learning Rate	Accuracy (%)
1000	200x200 px	50	0.0001	89.75
		50	0.001	89.62
		50	0.01	89.25
		100	0.0001	91.37
		100	0.001	91
		100	0.01	89.62

Based on the system test results table, it can be seen that the dataset is 1000 in size 200x200 px uses epoch 50 during the training process using a learning rate of 0.0001 produces an accuracy

rate of 89.75%, using a learning rate of 0.001 produces a rate accuracy is 89.62% while with a learning rate of 0.01 produces an accuracy rate of 89.25%. Testing with epoch 100 using a learning rate of 0,0001 produces 91.37% accuracy, with a learning rate of 0.001 produces an accuracy of 91.00%, and by using a learning rate 0.01 also produces an accuracy of 89.62%. From this comparison, it can be seen that the amount epoch and learning rate are very influential on the level of accuracy. The greater the number of epochs and learning rates than the better the level of accuracy. The greater the number of epochs then the better the level of accuracy, whereas if the greater the learning rate the worse the results of its accuracy and in the training process are unstable.

#### 4 Conclusion

In this research, four types of the grape leaves have been identified. Convolutional neural network models successfully implementing deep learning with Keras libraries yielding an accuracy of 91.37%. By using 80% images for training, and 20% images for testing, the classification algorithm used in this study allows the system to get a variety of samples. The results showed that to improve identification accuracy can be done by increasing the number of epochs and using a smaller learning rate. For further research, it is recommended to use more types of grape leaf disease and use other algorithms and other deep learning structures.

#### References

- [1] S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, and P. Kulkarni, "Diagnosis and classification of grape leaf diseases using neural networks," 2013 4th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2013, no. September 2015, pp. 2–7, 2013, doi: 10.1109/ICCCNT.2013.6726616.
- [2] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," Conf. Adv. Signal Process. CASP 2016, pp. 175–179, 2016, doi: 10.1109/CASP.2016.7746160.
- [3] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in European conference on computer vision, 2016, pp. 499–515.
- [4] L. Zhang, G.-S. Xia, T. Wu, L. Lin, and X. C. Tai, "Deep learning for remote sensing image understanding," J. Sensors, vol. 2016, 2016.
- [5] 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, 2016.
- [6] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," Neurocomputing, vol. 267, pp. 378–384, 2017.
- [7] C. DeChant et al., "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," Phytopathology, vol. 107, no. 11, pp. 1426–1432, 2017.
- [8] J. Zhu and A. Wu, "Identification of grape diseases using image analysis and BP neural networks," 2019.
- [9] S. M. Jaisakthi, P. Mirunalini, and D. Thenmozhi, "Grape Leaf Disease Identification using Machine Learning Techniques," 2019 Int. Conf. Comput. Intell. Data Sci., no. January 2020, pp. 1–6, 2019, doi: 10.1109/ICCIDS.2019.8862084.
- [10] "New Plant Diseases Dataset — Kaggle." 2018, Accessed: Jan. 16, 2020. [Online]. Available: <https://www.kaggle.com/vipooooo/new-plant-diseases-dataset>.
- [11] J. Ludwig, "Image Convolution," Portl. State Univ., pp. 1–8, 2013.
- [12] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv Prepr. arXiv1511.08458, 2015.
- [13] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," Comput. Intell. Neurosci., vol. 2017, 2017.
- [14] S.-H. Wang et al., "Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling," Front. Neurosci., vol. 12, p. 818, 2018.
- [15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.
- [16] H. Abhirawa, J. Jondri, and A. Arifianto, "Face Recognition Using Convolutional Neural Network," eProceedings Eng., vol. 4, no. 3, 2017.