# Comparing the Architecture of Convolutional Neural Network for Corn Leaves Diseases Image Classification

Aeri Rachmad(✉), Wahyudi Setiawan, and Eka Mala Sari Rochman

Faculty of Engineering, University of Trunojoyo Madura, Bangkalan, Indonesia
{aery_r,wsetiawan}@trunojoyo.ac.id

**Abstract.** This article discusses the comparison of the Convolutional Neural Network (CNN) architecture for image classification of corn leaves diseases. This study uses public data on from Plantvillages. The data be contained in four classes: gray leaf spot, common rust, leaf blight, and healthy. Each class consists of 1000 images, except for the gray leaf spot only 500 images. The experiment uses five CNN architectures: SqueezeNet, AlexNet, ResNet18, ResNet50, and ResNet101. SqueezeNet and AlexNet are CNN sequential models with deeper layer models. While Residual Network (ResNet) is a CNN architecture that utilizes additional output from the previous two layers to be used as input to the next layer. The CNN parameters used during the training step are learning rate 0.0001, epoch 1, batch-size 8, and adaptive moment estimation (Adam). The experiment was running in a single Central processing Unit (CPU). The percentage of training and testing data is 70:30. ResNet50 shows the best accuracy up to 95.59% with a computation time of 79 min. and 10 s. The experiment shows that the use of CNN architecture with the residual network model is better than the sequential network model.

**Keywords:** Corn Leaves Diseases · Image Classification · Convolutional Neural Network · Residual Network

## 1 Introduction

Diseases of corn can affect all parts of the plant, from the roots, stems, leaves to the cob. The cause of the disease needs to be known in order to maximize the treatment carried out. Corn diseases are easiest to notice from the leaves. The wide shape of the leaves makes it easy to make observations. Identification of corn plant diseases can be known from the symptoms and discoloration that appear in the corn leaf area. To identify a disease in a plant requires an expert. Usually in certain regions of Indonesia there are experts who are referred to as agricultural extension workers. However, the limited number of agricultural extension workers is a problem.

An alternative to having knowledge is to build a computerized system. Research on the classification of diseases on corn leaves has been extensively carried out. Disease classification in corn leaves is one way to determine the accuracy of disease diagnosis using symptoms or changes that exist in corn leaves. In 2018, Xihai Zhang, Yue


I. H. Agustin (Ed.): ICONNSMAL 2022, AISR 177, pp. 81–88, 2023.
https://doi.org/10.2991/978-94-6463-174-6_9

Qiao, Fanfeng Meng, Chengguo Fan, Mingming Zhang conducted research on disease identification in corn garden leaves. This study used two models from the improved Convolutional Neural Network (CNN), namely GoogleNet and Cifar10. The data used in this study was taken from several sources, namely Plant Village and Google Websites and was divided into 8 disease categories with a total data of 500 images. The GoogLeNet model achieved an average identification accuracy of 98.9%, and the Cifar10 model achieved an average accuracy of 98.8% [1].

In 2020, Valeria Maeda-Gutiérrez, Carlos E. Galván-Tejada, Laura A. Zanella-Calzada, José M. Celaya-Padilla, Jorge I. Galván-Tejada, Hamurabi Gamboa- Rosales, Huizilopoztli Luna-García, Rafael Magallanes-Quintanar, Carlos A. Guerrero, Méndez, Carlos A. Olvera-Olvera conducted research on CNN architectural comparisons to classify diseases of tomato plants. The study compared AlexNet, GoogleNet, InceptionV3, ResNet18, and ResNet50 architectures. The dataset used is divided into 9 classes with a total data of 18,160 images. The results of the study obtained the highest accuracy value from GoogleNet with an accuracy of 99.72% [2].

In 2019, Sachin B. Jadhav, Vishwanath R. Udupi, Sanjay B. Patil conducted a study on the classification of diseases on the leaves of soybean plants. This research used 5 CNN models, namely AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201. The dataset uses 1200 images of soybean leaf crops divided into four classes. The results of the study based on the proposed method using AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201 pre-trained networks achieved an accuracy of 95%, 96.4%, 96.4%, 92.1%, 93.6% [3].

In 2021, Minlan Jiang, Peilun Wu, Fei Li conducted research on the detection of dark spots on eggs using the GoogleNet model CNN. The dataset used in this study used 1200 images of eggs with bitnik and 8850 images of normal eggs with a total of 10,050 data. The results of this study showed that GoogleNet has the highest accuracy with a value of 98.19% [4].

In this article, the classification of diseases in corn images is implemented using a comparison of five CNN architectures, namely AlexNet, SqueezeNet, ResNet18, ResNet50, and ResNet101. we conduct an experiment using a single CPU with learning rate parameters of 0.0001, epoch 1, batch-size 8, and adaptive moment estimation (Adam).

## 2 Research Method

In this section, it is explained about the research stages consisting of input datasets, split data training and testing, cross-validation, CNN pre-trained network, and testing using the best model results from the training process. The full process of the research stages is shown in Fig. 1.

### 2.1 Input Dataset

The image that will be used is the image of the leaves of the corn plant that you get on the https://www.kaggle.com/datasets/arhasnaazzahra/cornleavediseases. The trial data comes from the plant village. The data totals 3. 500 images of corn leave and consists

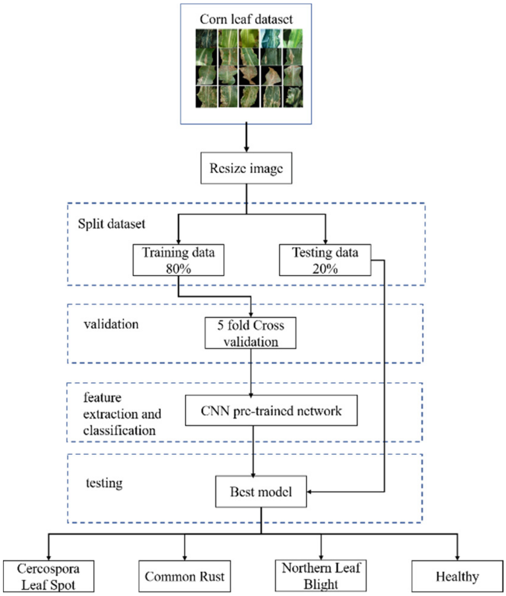


**Fig. 1.** The proposed system designs


**Fig. 2.** Diseases of corn leaves (source: plantvillage)

of 4 classes divided based on the condition of the corn leaves, where each class has a different number. All classes have 1000 images, except gray leaf spot has only 500 images. Corn leaf data is shown in Fig. 2.

### 2.2 Resize Image

The imagery on the dataset adjusts to the input size of the CNN. For CNN AlexNet and SqueezeNet using input size 227 × 227. As for CNN ResNet, it uses an input size of 224 × 224.

### 2.3 Split Dataset

Split dataset for training and testing 80:20. In the training data, a validation process is carried out to get the best model. Training data validation uses 5-fold cross validation, which is to divide the data into five parts. Four parts are used for training, one part is used for validation. Next compare the results of fold-1 to fold-5. The best results will be saved as the best model. Furthermore, the testing data uses the best model for the classification test process.

### 2.4 CNN Architecture

From testing the training data and testing data, it produces a confusion matrix as shown in Table I. This confusion matrix is used to measure performance in machine learning classification problems where the output can be in the form of two or more classes [11, 12]. From the confusion matrix table, accuracy, recall, precision, and F1 score will be calculated.

#### 2.4.1 AlexNet

The classification and detection of objects in images begins with the emergence of simple datasets such as CIFAR [5–7] and NORB[8–10]. Conventional machine learning can solve datasets that have tens of thousands of images. But along with real-world comparisons of having to classify and detect millions of images, the magnet I dataset [11–13] emerged. In 2010, a competition for classification and detection of objects on imagery was carried out through the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

The challenges of processing large amounts of data making conventional machine learning less optimal for solving them. Alexnet started competing in 2010, but won first in 2012 with the smallest error of 15.3%. ILSVRC classified with 1.2 million training data, 50 thousand validation data, and 150 thousand testing data.

Alexnet consists of eight layers with five convolution layers and three fully connected layers. In addition, Alexnet has a non-linear Rectified Linear Unit (ReLU) that can perform activation six times faster than tanh activation. Alexnet can also be run using multiple Graphical Processing Units (GPUs). The existence of overlapping pooling reduces errors by up to 0.5%. However, Alexnet has an overfitting problem because it has more than 60 million parameters. Generally this problem is mitigated by the use of data augmentation and dropouts [13–15].

#### 2.4.2 SqueezeNet

SqueezeNet is a CNN by Deep Scale, UC Berkeley and Stanford University. SqueezeNet has a smaller size than Alexnet. The SqueezeNet architectural design is as follows [16, 17]:

a. Replace the $3 \times 3$ filter to $1 \times 1$. The parameters on the SqueezeNet become nine times smaller than using a $3 \times 3$ filter
b. Resize channel input to filter $3 \times 3$
c. Down sampling of the final layers makes the SqueezeNet have a big map activation that can make classification accuracy higher
d. Numbers 1 and 2 reduce the number of parameters, while number 3 maximizes the accuracy value with limited parameters

e. Availability of fire modules. It consists of a squeeze and expand process. Squeeze has a $1 \times 1$ convolutional filter. While the expand consists of a 1x1 and 3x3 convolution filter. Both filters are followed by ReLU activation
f. Squeeze architecture consists of convolutional layers (conv 1). Followed by eight fire modules (fire 2 – 9). Next is the final convolutional layer (conv. 10). The number of modules per layer filters goes up from the beginning to the end of the layer (128,256, 384, 512). In addition, there is max-pooling with stride 2 on conv 1, fire 4, fire 8 and conv. 10
g. SqueezeNet reduces the model 50 times smaller than AlexNet.

### 2.4.3 ResNet

At first CNN just added a deeper layer. In the deep layers of the network, the more progressive the learning of complex features. At the beginning layer learn edges. On the second layer learn about shapes. The next layer learns about the object.

On CNN sequential, there is a saturation of the value of the activation result which is getting smaller and causes overfitting. The more layers, the result does not improve, but the performance decreases. This can be seen from the trials conducted by Kaiming He et al. using 20 layers compared to 56 layers. The performance of 56 layers is lower than 20 layers. These problems can be in the form of optimization functions, initialization of the network and vanishing gradient problems. This can be overcome by Residual Blocks on the ResNet. Residual blocks have a "skip connection" that does not have any parameters. It connects the output of the previous layer to the next layer. The results can conduct training well and produce higher testing performance compared to CNN Sequentials. The existence of a "skip connection" also saves the use of parameters. ResNet 152 layers have a much lower number of parameters than VGG. So that computing can be done faster. ResNet won ILSVRC in 2015 with an error rate of 3.57% [18–20].

## 2.5 Evaluation

Performance measurements are performed using confusion matrices. Through the confusion matrix, we can find out the accuracy of the model we make with performance measures accuracy, recall, and precision. Figure 3 is a confusion matrix for multiclass.

Predicted

Actual

| Classes | A | B | C |
|---|---|---|---|
| A | TN (True Negative) | FP (False Positive) | TN (True Negative) |
| B | FN (False Negative) | TP (True Positive) | FN (False Negative) |
| C | TN (True Negative) | FP (False Positive) | TN (True Negative) |

**Fig. 3.** Confusion matrix for calculating class B performance

## 3 Result and Discussion

The trial was conducted by comparing 3architectures namely AlexNet, SqueezeNet, and ResNet. For ResNet do a comparison between ResNet18, ResNet50, and ResNet101. The test run was conducted on a single CPU. The results of the trial are shown in Table 1.

The results of the trial showed the best accuracy obtained ResNet50. The highest accuracy was 95.59%. Computing time 79 min. 57 s. In Table 2, a comparison of the CNN architecture tested in terms of salient features, number of convolutional layers and number of parameters is presented.

Comparison between CNN architectures:

1. Salient feature. AlexNet relies on deeper layers. While SqueezeNet has a fire module that squeezes and expands each module. While resnet uses skip connection
2. The number of convolutional layers does not affect the accuracy result. Each CNN architecture has its own characteristics. AlexNet when compared to SqueezeNet has a smaller number of convolutional layers. While resnet50 has a greater number of convolutional layers than AlexNet and SqueezeNet.
3. The number of parameters plays a role against the compute time. To get the best accuracy. Resnet requires one epoch with 66 iterations. The time required for computing is 23 min. 10 s.

**Table 1.** Comparison Result Corn Leaf Diseases Classification Using CNN

| Architecture CNN | Accuracy (%) | Iteration | Time |
|---|---|---|---|
| **AlexNet** | 85.07 | 36 | 8 min. |
| **SqueezeNet** | 88.67 | 150 | 33 min. 37 s. |
| **ResNet18** | 86.60 | 60 | 23 min. 10 s |
| **ResNet50** | 95.59 | 66 | 79 min. 57 s. |
| **ResNet101** | 85.43 | 54 | 137 min. |

**Table 2.** Architecture CNN Layer and Parameter

| CNN | Salient feature | Convolution layer | Parameter (million) |
|---|---|---|---|
| **AlexNet** | Deeper | 8 | 62 |
| **SqueezeNet** | Fire-module | 10 | 4. 8 |
| **ResNet18** | Skip connection | 5 | 11.5 |
| **ResNet50** | Skip connection | 22 | 23.9 |
| **ResNet101** | Skip connection | 48 | 43.8 |

## 4 Conclusion

In this study, it has been discussed about the classification of disease images on corn leaves. The classification consists of four classes, namely common rust, gray leaf spot, healthy, and northern leaf blight. The number of images in each class is 1,000 images, except gray leaf spot only 500 images. The trial used learning rate parameters of 0.0001, epoch 1, batch-size 8, and adaptive moment estimation (Adam). The trial results show the best accuracy achieved by ResNet50 up to 95.59%.

**Acknowledgment.** This research was made possible thanks to the University of Trunojoyo Madura's Faculty of Engineering and Institute for Research and Community Service. Additionally, we are indebted to the Multimedia and Network Laboratory research team for their assistance in making this study a success. The UTM DIPA Fund No. Supports this research article. For Independent Research, see 225/UN46.4.1/PT.01.03/2022.

## References

1. X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, no. c, pp. 30370–30377, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2844405.
2. V. Maeda-Gutiérrez *et al.*, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: https://doi.org/10.3390/app10041245.
3. S. B. Jadhav, V. R. Udupi, and S. B. Patil, "Convolutional neural networks for leaf image-based plant disease classification," *IAES Int. J. Artif. Intell.*, vol. 8, no. 4, pp. 328–341, 2019, doi: https://doi.org/10.11591/ijai.v8.i4.pp328-341.
4. M. Jiang, P. Wu, and F. Li, "Detecting dark spot eggs based on CNN GoogLeNet model," *Wirel. Networks*, vol. 4, 2021, doi: https://doi.org/10.1007/s11276-021-02673-4.
5. A. Darkwah Akwaboah, "Convolutional Neural Network for Image Classification," 2019. [Online]. Available: http://www.cs.jhu.edu/~cwang107/files/cnn.pdf.
6. F. O. Giuste and J. C. Vizcarra, "CIFAR-10 Image Classification Using Feature Ensembles," 2020. [Online]. Available: http://arxiv.org/abs/2002.03846.
7. S. Singla, S. Singla, and S. Feizi, "Improved_Deterministic_L2_Robu," pp. 1–30, 2022.
8. B. Chandler and E. Mingolla, "Mitigation of Effects of Occlusion on Object Recognition with Deep Neural Networks through Low-Level Image Completion," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: https://doi.org/10.1155/2016/6425257.
9. D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. February, pp. 3642–3649, 2012, doi: https://doi.org/10.1109/CVPR.2012.6248110.
10. K. Schlachter, C. DeFanti, S. Herscher, K. Perlin, and J. Tompson, "Beyond Photo Realism for Domain Adaptation from Synthetic Data," 2019, [Online]. Available: http://arxiv.org/abs/1909.01960.
11. Y. You, Z. Zhang, C. Hsieh, J. Demmel, & K. Keutzer. "Imagenet training in minutes". In Proceedings of the 47th International Conference on Parallel Processing. pp. 1–10, 2018.
12. B. Recht, R. Roelofs, L. Schmidt, & V. Shankar. "Do imagenet classifiers generalize to imagenet?". In International Conference on Machine Learning, pp. 5389–5400, 2019.

13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” 2012.
14. S. Lu, Z. Lu, and Y. D. Zhang, “Pathological brain detection based on AlexNet and transfer learning,” *J. Comput. Sci.*, vol. 30, pp. 41–47, 2019, doi: https://doi.org/10.1016/j.jocs.2018.11.008.
15. M. Z. Alom *et al.*, “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches,” 2018, [Online]. Available: http://arxiv.org/abs/1803.01164.
16. H. J. Lee, I. Ullah, W. Wan, Y. Gao, and Z. Fang, “Real-Time vehicle make and model recognition with the residual squeezenet architecture,” *Sensors (Switzerland)*, vol. 19, no. 5, 2019, doi: https://doi.org/10.3390/s19050982.
17. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” in *ICLR*, 2017, pp. 1–13, doi: https://doi.org/10.1007/978-3-319-24553-9.
18. S. Xie, R. Girshick, and P. Doll, “Aggregated Residual Transformations for Deep Neural Networks http://arxiv.org/abs/1611.05431v2,” *Conf. Comput. Vis. Pattern Recognit.*, pp. 1492–1500, 2017.
19. P. Goyal *et al.*, “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour,” 2017, [Online]. Available: http://arxiv.org/abs/1706.02677.
20. K. He, X. ZHang, S. Ren, and J. Sun, “Deep residual learning for image steganalysis,” *Multimed. Tools Appl.*, pp. 1–17, 2015, doi: https://doi.org/10.1007/s11042-017-4440-4.