Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2022, 2022:110
https://doi.org/10.28919/cmbn/7726
ISSN: 2052-2541

MODIFIED-RESIDUAL NETWORK FOR MAIZE STALK ROTS DISEASES CLASSIFICATION

WAHYUDI SETIAWAN[1], YOGA DWITYA PRAMUDITA[2], RIRIES RULANINGTYAS[3,*]

[1]Department of Information System, University of Trunojoyo Madura, Bangkalan, Indonesia

[2]Department of Informatics, University of Trunojoyo Madura, Bangkalan, Indonesia

[3]Department of Physics, Universitas Airlangga, Surabaya, Indonesia

Abstract: In this article, image classification of maize stalk rots diseases was carried out. The experiment used primary data taken from maize plantations in Bangkalan, Madura. The data consists of three classes: healthy, anthracnose, and gibberella. For deep learning experiments, we augmented the primary data. The total data was 2,211 images. An investigation is composed of three sections. First, we used five different Convolutional Neural Network (CNN) architectures, and second, the best CNN will be modified. Finally, it performed varied layer types from the second section. The parameters used were epoch 10, learning rate 3.10e-4, and minibatch-size 64. The distribution of training, validation, and testing data were 40:40:20. The result shows the best performance for the first section is ResNet18. Next step, we modify ResNet18 into six different architectures. From the second section, the best results were ResNet18 and modified-ResNet, but modified-ResNet has less number of parameters. The third section's results showed accuracy, precision, and recall were 99.55%, 99.53%, and 99.73%, respectively. The modified-ResNet architecture is suitable for classifying maize stalk rots diseases.

---

*Corresponding author
E-mail address: riries-r@fst-unair.ac.id
Received

Keywords: convolutional neural network; image classification; maize stalk rots diseases; modified-ResNet; residual network.

2010 AMS Subject Classification: 92B20.

1. INTRODUCTION

Computer vision research in agriculture has been used widely. Computer vision in agriculture is an alternative to taking expert knowledge for various purposes such as planting [1]–[3], harvesting [4]–[6], advanced analysis of weather conditions [7], [8], weeding, and plant health detection and monitoring [9]–[11]. Computer vision has three steps: preprocessing, feature extraction, and classification. The first two critical steps are crucial for processing data. Preprocessing aims to ensure good image quality before data analysis is carried out. Feature extraction is tasked with being able to find essential features in the experimental data. Feature extraction can be of two types: manual and automatic. Manual feature extraction usually uses conventional machine learning, while automatic feature extraction uses deep learning. In conventional machine learning, feature extraction is obtained from color, shape, and texture features with various derived sub-features [12]–[15]. It is indeed troublesome to determine the most appropriate features. Sometimes, some studies use feature selection to get the best features [16], [17]. Currently, the existence of deep learning has extensive applications, including in agriculture.

We used a Convolutional Neural Network (CNN), one of the popular deep learning. An application of CNN in agriculture is the classification of maize diseases. Identification can be made through leaves, cobs, stalk rots, and roots. Previous studies have carried out many maize diseases but were limited to leaf diseases. Research by Hu et al. uses transfer learning with fine-tuning GoogleNet. The dataset comes from Plantvillage with 4,354 images with four classes of maize leaves and is divided into 70% for training and 30% for testing. The results showed accuracy, precision, and recall were 98.15%, 98.05%, and 98.02%, respectively [18].

Another study used a local dataset from Tanzania, consisting of two classes, infected and noninfected. The number of training data is 7,896, validation data is 1,692, and testing data is 1,692. This research uses sir Thomas corner detection in preprocessing. The CNN architecture consists of VGG16, VGG19, InceptionV3, and MobileResnetV2. The results showed the accuracy of 99.92%, 99.67%, 100%, and 100% respectively [19].

Subsequent research uses data from plantvillages with a limited number of 100 per class. Distribution of training and testing data 70:30. Test using fine-tuning VGG16. The final result showed an accuracy of 89% [20]. The study used two CNN: VGG and ResNet. The experiment uses 10,000 images with three classes. The percentage of training and testing data is 85:15. The results showed an accuracy of 98.83% [21].

In addition to classifying diseases on maize cobs, few references have been produced, studying the classification of eight classes, including maize leaf and cobs. The dataset consists of 3,852 images. The results showed 95.86% accuracy, 97.63% precision, and 83.45% recall [22]. Another research is using CNN two path way. The dataset has 17,502 images. The results showed an accuracy of 97.23% [23].

This study discusses the classification of diseases that attack maize stalk. Research on maize stalk has never been done before. It is also vital for detecting diseases other than leaves. Conditions of maize stalks have a more severe risk than other diseases because the stalk rots are the central part that can support the leaves and leaves midrib. So this problem will cause the plant collapse and reduce the number of maize kernels [24].

The experiment classified maize stalk into three classes: healthy, gibberella, and anthracnose. Disease classification in maize stalk using a variety of CNN architecture. It is to find out the best CNN architecture. This article has the following composition: section 1 is the introduction. Section 2 is a method, novelty, and dataset (primary and augmentation data). Section 3 discusses the results and discussing them. In the end, section 4 is the conclusions and suggestions for further research.

2. PRELIMINARIES

2.1 Overview

This research applies novelties including:

(i) The maize stalk data obtained primarily from maize fields in Bangkalan, Madura.

(ii) An augmented data with ten different techniques of augmentation. It produces 2,010 new images. The primary data only 201, so there are total 2,211 images.

(iii) Feature extraction and classification performed with CNN. We select the CNN architecture with the best accuracy. Furthermore, modifications to the best accuracy architecture. Simple changes are made, especially in the network's number of layers and kernels.

(iv) For comparison, we tested scenarios for varied types of CNN layers such as the effect of global average pooling, batch-normalization, Rectified Linear Unit (ReLU), leaky ReLU, and dropout.

2.2 Data collection

The image of the maize stalk was obtained through a photo with a mobile phone. We took photographs in sunny. The background is coated using white cardboard. The maize plants selected for the dataset were fifthy days old.

The dataset consists of three classes: healthy, anthracnose, and gibberela. Anthracnose is caused by the fungus Colletotrichum sp. [25]. While gibberella is caused by the fungus Gibberella zeae [26]. Each class has 67 images, so a total of 201 images of maize stalk rots. The sample of the primary data is shown in Figure 1.

Figure 1. Maize Stalk Rots (top to bottom: anthracnose, gibberella, dan healthy)

2.3 Data Augmentation

The amount of data per class was 67. It is not feasible for deep learning classification processes. For this reason, data augmentation is needed, which aims to modify or manipulate training data so that the number of data increases. Before data augmentation, we perform manual labeling by Bangkalan Agriculture Services.

Data size as an input of each CNN architecture is different. For Resnet, VGG, and Googlenet uses 224×224 pixels, while Alexnet uses 227×227 pixels. In this study, we carry out ten augmentation techniques: horizontal flip, vertical flip, rotate right, rotate left, auto-adjust colors, and gamma correction with values of 2,3,4,5 and 6. So the total data used is 2,211 images, consisting of 737 appeareances for each class.

2.4 The testing scenario

The experiment scenario consists of three steps:

(i) Maize stalk image classification using CNN alexnet [27], VGG [28], resNet [29], googlenet [30]. Classification using the existing CNN architecture. Input size 224×224 from CNN, including Googlenet, VGG16, and ResNet. While alexnet uses Alexnet 227×227 [31]. The parameter set is epoch 10, learning rate 3.10e-4, and minibatch-size 64.

(ii) Classification of maize stalk rots images using the best CNN modification from scenario 1. Perform conversion by changing the number of filter sizes and the number of layers.

(iii) Experiment using various epochs, learning rates, and minibatch sizes and studying the effect of layers on CNN architecture result from test scenario 2.

2.5 Maize Talk Rots Classification

This section describes the stages of the maize stalk roots classification as follows:

(i) Input data; the image adjusts to the size of each CNN architecture. Perform ten augmentation models as described in section 3.3. The total number of original and augmented data is 2,211 images.

(ii) Split training, validation, and testing 40:40:20. So the complete data for training is 884, validation also 884, while the total data for testing is 443 images. It is randomly choosing data for training and testing. The result of training and validation becomes the model.

(iii) We do testing data with the model of CNN from training and validation data. In the testing, do cross-validation between training and validation. Cross-validation aims to get the best model of the network used. We used 5-fold cross-validation. Figure 2 shows the proposed maize stalk rots classification.

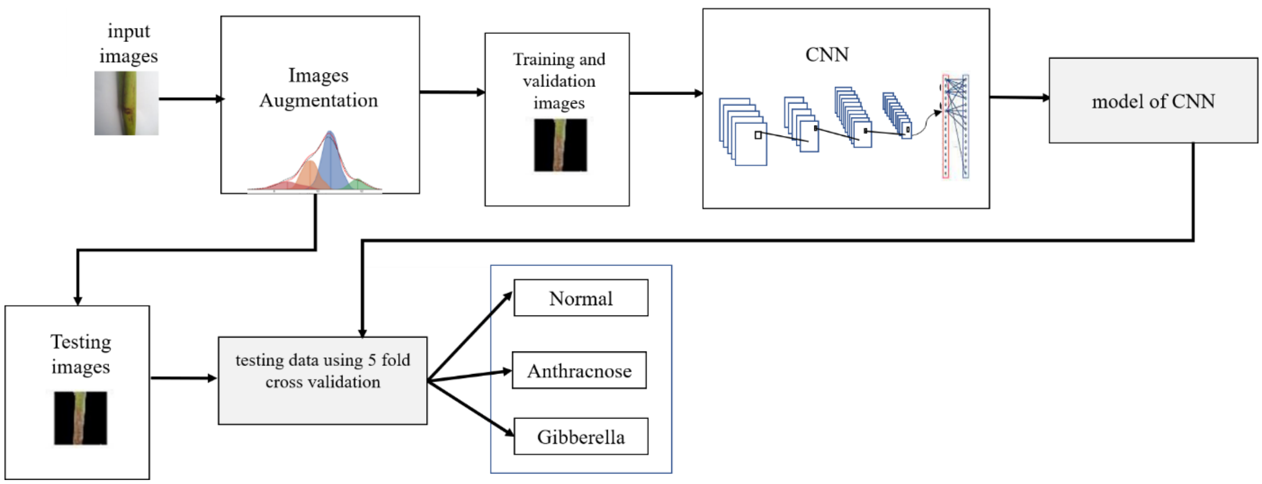


Figure 2. Block diagram of proposed maize stalk rots

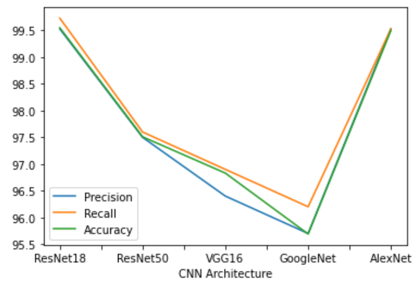
The CNN architecture used for this research consists of AlexNet, VGG16, Googlenet, Resnet50, and ResNet18. The test results are performance measures of the five CNN architectures. Performance measures involve accuracy, precision, and recall.

3. MAIN RESULTS

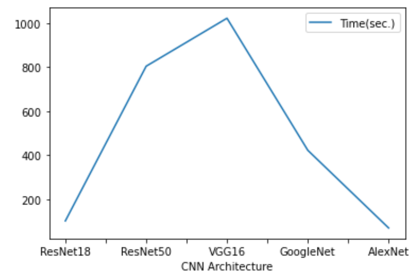
The maize stalk rots is run using Google Collab with specifications processor Core i7-7700, GTX 1060 6 Gb D5 amp, Solid State Drive 60 Gb. The experiment used tensorflow, keras, opencv, numpy and matplotlib in the phyton library.

3.1. Maize stalk rots classification using CNN

The first experiment carried out image classification of maize stalk rots disease using AlexNet, VGG16, GoogleNet, ResNet50, and ResNet18 architectures. Parameter used in the experiment are epoch 10, learning rate 3×10e-4, Stochastic Gradient Descent with Momentum (SGDM). The results of the first experiment are shown in Figure 3.



(a)



(b)

Figure 3 (a). Performences measure and (b) Computation time of first experiment

Figure 3 (a) shows the first experiment produces the best accuracy of 99.55%, precision of 99.53%, and recall of 99.73% using the ResNet18 architecture. While alexnet produced accuracy of 99.50%, precision of 99.53%, and recall of 99.53%. The computation time in figure 3b shows the fastest computation time by Alexnet at 70 sec. Meanwhile, ResNet18 needs 102 sec.

The results show that the dataset used does not require an intense deep network. ResNet18 has 18 layers consisting of 17 convolutional layers and one fully connected layer. The characteristic of the ResNet architecture is to have a skip connection. It aims to avoid the inability of deeper layers to learn and even identity mappings, the type of skip connection used in addition [29]. The number of parameters contained in the Resnet architecture is much smaller than the parameters in the Alexnet, GoogleNet and VGG. The number of parameters on Resnet18, Resnet50, Googlenet, Alexnet, and VGG16 are 11, 23, 23, 62, and 138 million, respectively. The number of parameters affects the computation time apart layer arrangement [32]. Furthermore, the test results in the first scenario are used as a reference for making modifications to the ResNet architecture.

3.2 Classification of Maize Stalk Rots Disease using modified-Resnet

3.2.1 Experiments using different filter dimensions and an amount of convolutional layers

The second experiment was conducted using a modified Residual Network (m-ResNet) architecture. The architecture of ResNet shows the difference in the size of the filter dimensions and the number of convolutional layers used. Table 1 shows the differences in the architectural structure of ResNet18 and m-ResNet.

Table 1. Comparison of ResNet18, 1st, 2nd, and 3rd m-ResNet Architecture

Layer name	ResNet18[33]	1st m-ResNet	2nd m-ResNet	3rd m-ResNet
conv1	7×7, 64 stride 2	7×7, 64 stride 2	7×7, 64 stride 2	7×7, 64 stride 2
pool1	3×3 maxpool stride 2	3×3 maxpool stride 2	3×3 maxpool stride 2	3×3 maxpool stride 2
conv2_x	$\begin{bmatrix}3 & 3 & 64\\3 & 3 & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 48\\3 & 3 & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 48\\3 & 3 & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 48\\3 & 3 & 64\end{bmatrix}\times 2$
conv3_x	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$
conv4_x	$\begin{bmatrix}3 & 3 & 256\\3 & 3 & 256\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 256\\3 & 3 & 256\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 256\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$
conv5_x	$\begin{bmatrix}3 & 3 & 512\\3 & 3 & 512\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 512\\3 & 3 & 512\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 256\\3 & 3 & 512\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 256\end{bmatrix}\times 2$
avg_pool	7 × 7, avg_pool, stride 1	7×7, avg_pool, stride 1	7×7, avg_pool, stride 1	7×7, avg_pool, stride 1
fcl	4×1, 512, stride 2	4×1, 512, stride 2	4×1, 512, stride 2	4×1, 256, stride 2

Table 2. Comparison of 4th, 5th, and 6th m-ResNet Architecture

Layer name	4th m-ResNet	5th m-ResNet	6th m-ResNet
conv1	7×7, 64 stride 2	7×7, 64 stride 2	7×7, 64 stride 2
pool1	3×3 maxpool stride 2	3×3 maxpool stride 2	3×3 maxpool stride 2
conv2_x	$\begin{bmatrix}3 & 3 & 48\\3 & 3 & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 64\\3 & 3 & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 64\\3 & 3 & 64\end{bmatrix}\times 2$
conv3_x	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 128\end{bmatrix}\times 2$
conv4_x	$\begin{bmatrix}3 & 3 & 96\\3 & 3 & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3 & 3 & 256\\3 & 3 & 256\end{bmatrix}\times 1$	
conv5_x	$\begin{bmatrix}3 & 3 & 128\\3 & 3 & 256\end{bmatrix}\times 1$		
avg_pool	7 × 7, avg_pool, stride 1	7×7, avg_pool, stride 1	7×7, avg_pool, stride 1
fcl	4×1, 256, stride 2	4×1, 256, stride 2	4×1, 128, stride 2

The dimensions of the filter in the convolutional layer are different in each architecture. The difference is in the 2nd, 3rd, 4th, and 5th convolutional layers. Only modifications are made in the 1st to 3rd m-Resnet architecture on the number of filter dimensions. The 4th m-Resnet architecture eliminates the 5th convolutional layer. The 5th m-Resnet architecture consists of convolutional layers 1, 2, 3, and 4a. The 6th m-Resnet has convolutional layers 1, 2, and 3a. The comparison of filter dimensions on m-ResNet is shown in Table 2.

Table 2. Total of filter dimension

Architecture	ResNet18	1st *m*-ResNet	2nd *m*-ResNet	3rd *m*-ResNet	4th *m*-ResNet	5th *m*-ResNet	6th *m*-ResNet
filter dimension	4,800	4,704	3,936	2,464	2,240	1,728	704

After modifying the number of filter dimensions and the number of convolutional layers used, the m-Resnet architectures were tested. The results of scenario two are shown in Table 3.

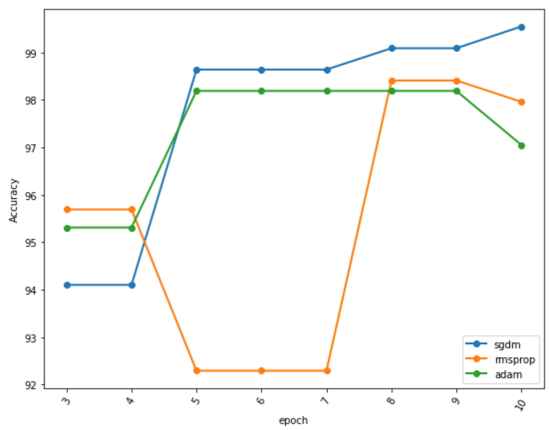
Table 3. Comparison of the results using m-ResNet

Architecture	Precision(%)	Recall(%)	Accuracy(%)	Time
ResNet18	99.53	99.73	99.55	2 min. 42 sec.
1st m-ResNet	99.32	99.32	99.32	2 min. 32 sec.
2nd m-ResNet	99.53	99.73	99.55	2 min. 25 sec.
3rd m-ResNet	99.09	99.09	99.09	2 min. 03 sec.
4th m-ResNet	98.87	98.87	98.87	2 min. 14 sec
5th m-ResNet	99.53	99.57	99.50	1 min. 47 sec
6th m-ResNet	94.24	94.24	94.24	1 min. 23 sec

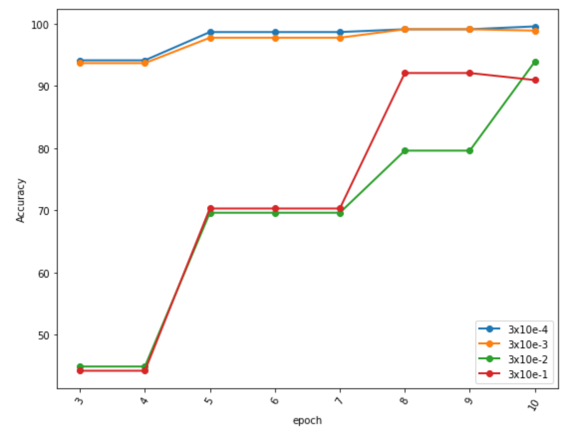
The second experiment results show the architecture with the best performance produced by ResNet18 and 2nd m-ResNet, accuracy up to 99.55%. Meanwhile, the computation time is lower for the 2nd m-ResNet architecture because of the fewer parameters that must update. A comparison of test results shows that the difference in accuracy is not up to 1%, except for the 6th m-Resnet architecture.

3.3 The experiment by changing the parameter value and type of layer

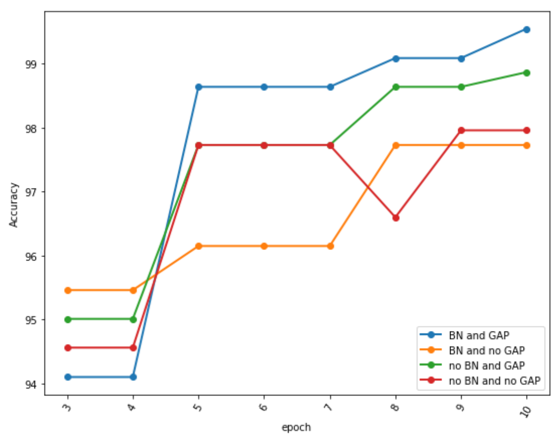
The third experiment is to set difference of learning rate and optimization of gradient descent. Next, the effect test is carried out Batch Normalization (BN), Global Average Pooling (GAP), ReLU, and Dropout (DO) layer. The third experiment were carried out with epochs one to ten on 2nd m-Resnet architecture. The result shows in figure 4.



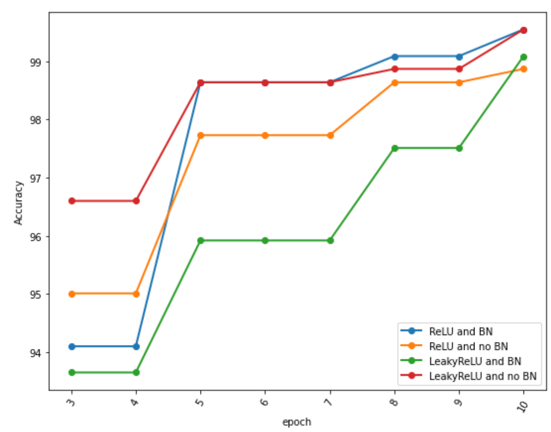
(a)



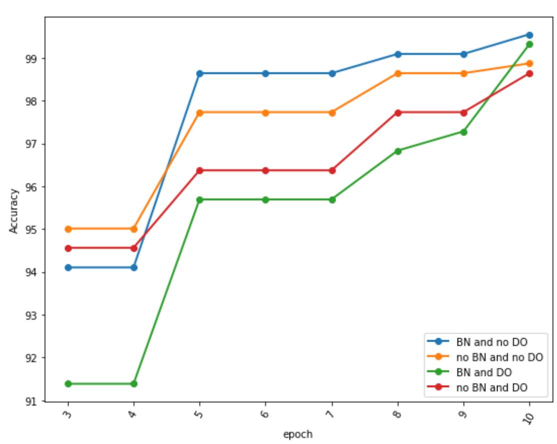
(b)



(c)



(d)



(d)

Figure 4. The result of third experiment. Accuracy based on difference value of (a) gradient descent, (b) learning rate, (c) Batch Normalization (BN) and Global Average Pooling (GAP), (d) Rectified Linear Unit (ReLU) and BN, (e) BN and Dropout (DO)

Figure 4 (a) shows the results of the comparison of accuracy using stochastic gradient descent with momentum (SGDM), Root Means Square Propagation (RMSProp), and Adaptive Moment Optimization (adam). The results show that using the SGDM method is better than the other two methods. Figure 4 (b) shows a comparison of accuracy using various learning rate values including 3×10e-4, 3×10e-3, 3×10e-2, and 3×10e-1. Best accuracy using learning rate 3×10e-4. Figure 4 (c) shows the results of testing the effect of the BN and GAP layer. The combination is carried out with four types: BN and GAP, BN and without GAP, GAP and without BN, and without BN and GAP. The trial results show the highest accuracy using a combination of BN and GAP.

Figure 4 (c) shows the experiment result for the ReLU and BN layers. The combination is carried out with four types: ReLU and BN, ReLU and without BN, Leaky ReLU and BN, and leaky ReLU without BN. The results show the highest accuracy using a combination of ReLU and BN and leaky ReLU without BN. The accuracy of each is the same, 99.55%. Figure 4 (d) shows the results of testing the effect of BN and DO layer. The combination is carried out with four types: BN and DO, BN and without DO, DO and without BN, and without BN and DO. The test results show the highest accuracy using a combination of BN and without DO. The experiment shows that layer type arrangement, gradient descent, and learning rate affects accuracy result.

4. CONCLUSION

A disease classification on maize stalk rots has been carried out. The primary data with three classes of maize stalk rots (healthy, gibberella, and anthracnose). We add the data with image augmentation. The total data experiment was 2,211.

We carried out various scenarios, including the CNN architecture (ResNet18, ResNet50, VGG16, GoogleNet, and Alexnet). Next, we modify the ResNet18 architecture, which we identify as 1^{st} to 6*th* m-ResNet. The test results show the best performance using 2^{nd} m-ResNet with an accuracy, precision, and recall are 99.55%, 99,53%, and 99.73% respectively.

In addition, the various layer types were tested, including GAP, BN, dropout, and activation of LeakyReLU. The test results show that the effect of the dropout layer remains consistent even by removing BN on specific layers. The results achieved the same accuracy of 99.55%

For further research, we will be classifying all parts of the maize plant (leaf, stalk rots, cobs, and kernel) diseases. The method can be combined with the characteristics of a particular CNN architecture. In addition, hyperparameter tuning is also required to determine the best parameters that can produce optimal performance.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Z. Li, R. Guo, M. Li, Y. Chen, G. Li, A review of computer vision technologies for plant phenotyping, Computers Electron. Agric. 176 (2020), 105672. https://doi.org/10.1016/j.compag.2020.105672.

[2] K. Mitsuno, Y. Nomura, T. Kurita, Channel planting for deep neural networks using knowledge distillation, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, Milan, Italy, 2021: pp. 7573–7579. https://doi.org/10.1109/ICPR48806.2021.9412760.

[3] M. Koklu, I.A. Ozkan, Multiclass classification of dry beans using computer vision and machine learning techniques, Computers Electron. Agric. 174 (2020), 105507. https://doi.org/10.1016/j.compag.2020.105507.

[4] M. Benavides, M. Cantón-Garbín, J.A. Sánchez-Molina, et al. Automatic tomato and peduncle location system based on computer vision for use in robotized harvesting, Appl. Sci. 10 (2020), 5887. https://doi.org/10.3390/app10175887.

[5] M. Faisal, M. Alsulaiman, M. Arafah, M.A. Mekhtiche, IHDS: Intelligent Harvesting Decision System for Date Fruit Based on Maturity Stage Using Deep Learning and Computer Vision, IEEE Access. 8 (2020), 167985–167997. https://doi.org/10.1109/access.2020.3023894.

[6] X. Zhang, M. Karkee, Q. Zhang, et al. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples, J. Field Robotics. 38 (2020), 476-493. https://doi.org/10.1002/rob.21998.

[7] Y. Liu, N. Noguchi, L. Liang, Development of a positioning system using UAV-based computer vision for an airboat navigation in paddy field, Computers Electron. Agric. 162 (2019), 126–133. https://doi.org/10.1016/j.compag.2019.04.009.

[8] P. Ramos-Giraldo, C. Reberg-Horton, A.M. Locke, et al. Drought stress detection using low-cost computer vision systems and machine learning techniques, IT Prof. 22 (2020), 27–29. https://doi.org/10.1109/mitp.2020.2986103.

[9] W.H. Su, D.C. Slaughter, S.A. Fennimore, Non-destructive evaluation of photostability of crop signaling compounds and dose effects on celery vigor for precision plant identification using computer vision, Computers Electron. Agric. 168 (2020), 105155. https://doi.org/10.1016/j.compag.2019.105155.

[10] O.M. Olaniyi, E. Daniya, J.G. Kolo, et al. A computer vision-based weed control system for low-land rice precision farming, Int. J. Adv. Appl. Sci. 9 (2020), 51-61. https://doi.org/10.11591/ijaas.v9.i1.pp51-61.

[11] C.L. Chang, K.M. Lin, Smart agricultural machine with a computer vision-based weeding and variable-rate irrigation scheme, Robotics. 7 (2018), 38. https://doi.org/10.3390/robotics7030038.

[12] E. Alehegn, Ethiopian maize diseases recognition and classification using support vector machine, Int. J. Comput. Vision Robotics. 9 (2019), 90. https://doi.org/10.1504/ijcvr.2019.098012.

[13] M. D. Chauhan, R. Walia, C. Singh, Detection of maize disease using random forest classification algorithm, Turk. J. Computer Math. Educ. 12 (2021), 715-720.

[14] F. Lin, D. Zhang, Y. Huang, et al. Detection of corn and weed species by the combination of spectral, shape and textural features, Sustainability. 9 (2017), 1335. https://doi.org/10.3390/su9081335.

[15] Y. Wei, L. Wei, T. Ji, et al. A novel image classification approach for maize diseases recognition, Recent Adv. Electric. Electron. Eng. 13 (2020), 331–339. https://doi.org/10.2174/2352096511666181003134208.

[16] N. Leena, K.K. Saju, Classification of macronutrient deficiencies in maize plant using machine learning, Int. J. Electric. Computer Eng. 8 (2018), 4197-4203. https://doi.org/10.11591/ijece.v8i6.pp4197-4203.

[17] K.H. Choi, S. Kim, Illumination-invariant vegetation detection for a vision sensor-based agricultural applications, Int. J. Electric. Computer Eng. 11 (2021), 1284-1292. https://doi.org/10.11591/ijece.v11i2.pp1284-1292.

[18] R. Hu, S. Zhang, P. Wang, et al. The identification of corn leaf diseases based on transfer learning and data augmentation, in: Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering, ACM, Beijing China, 2020: pp. 58–65. https://doi.org/10.1145/3403746.3403905.

[19] F.S. Ishengoma, I.A. Rai, R.N. Said, Identification of maize leaves infected by fall armyworms using UAV-based imagery and convolutional neural networks, Computers Electron. Agric. 184 (2021), 106124. https://doi.org/10.1016/j.compag.2021.106124.

[20] M. Sibiya, M. Sumbwanyambe, Automatic fuzzy logic-based maize common rust disease severity predictions with thresholding and deep learning, Pathogens. 10 (2021), 131. https://doi.org/10.3390/pathogens10020131.

[21] Y. Wu, Identification of maize leaf diseases based on convolutional neural network, J. Phys.: Conf. Ser. 1748 (2021), 032004. https://doi.org/10.1088/1742-6596/1748/3/032004.

[22] J. Chen, W. Wang, D. Zhang, et al. Attention embedded lightweight network for maize disease recognition, Plant Pathol. 70 (2020), 630–642. https://doi.org/10.1111/ppa.13322.

[23] J. Zhang, Q. Ma, X. Cui, et al. High-throughput corn ear screening method based on two-pathway convolutional neural network, Computers Electron. Agric. 175 (2020), 105525. https://doi.org/10.1016/j.compag.2020.105525.

[24] T.A. Jackson-Ziems, J.M. Rees, R.M. Harveson, Common stalk rot diseases of corn, Papers in Plant Pathology. (2014), 532. http://digitalcommons.unl.edu/plantpathpapers/532.

[25] Y. Suryadi, T.P. Priyatno, I.M. Samudra, et al. Control of anthracnose disease (colletotrichum gloeosporioides) using nano chitosan hydrolyzed by chitinase derived from burkholderia cepacia isolate E76, J. AgroBiogen. 13 (2018), 111-122. https://doi.org/10.21082/jbio.v13n2.2017.p111-122.

[26] M. Dragich, S. Nelson, Gibberella and Fusarium ear rots of maize in Hawai'i, Plant Disease, (2014), PD-102, 1-8. https://www.ctahr.hawaii.edu/oc/freepubs/pdf/PD-102.pdf.

[27] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 1–9, 2012.

[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015, pp. 1–14.

[29] K. He, X. Zhang, S. Ren, Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015 (2015).

[30] C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015: pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594..

[31] A.R. Luaibi, T.M. Salman, A.H. Miry, Detection of citrus leaf diseases using a deep learning technique, Int. J. Electric. Computer Eng. 11 (2021), 1719-1727. https://doi.org/10.11591/ijece.v11i2.pp1719-1727.

[32] I. Kandel, M. Castelli, Transfer learning with convolutional neural networks for diabetic retinopathy image

classification. A review, Appl. Sci. 10 (2020), 2021. https://doi.org/10.3390/app10062021.

[33] P. Napoletano, F. Piccoli, R. Schettini, Anomaly detection in nanofibrous materials by CNN-based self-similarity, Sensors. 18 (2018), 209. https://doi.org/10.3390/s18010209.