Identifying Types of Corn Leaf Diseases with Deep Learning

Rahul Firmansyah[1] and Nur Nafiiyah[1]
[1]Informatics Engineering Study Program, Faculty of Engineering, Lamongan Islamic University, Indonesia
Corresponding author: Nur Nafiiyah (e-mail: mynaff@unisla.ac.id).

ABSTRACT The government is trying to increase corn yields to meet the Indonesian population's food needs and for export abroad. Some farmers have yet to gain experience with the types of diseases in corn, so they need tools or systems to guide and provide information to new farmers. Many previous studies have developed automatic systems to identify corn leaf diseases, with the goal of increasing corn crop production by early recognition and control. We propose a system for identifying types of corn leaf diseases using the CNN (Convolutional Neural Network) method to be more precise in recognizing corn diseases early on. The methods used in previous research mostly used deep learning with high accuracy results above 90%. CNN is one of the deep learning methods, so we use it to identify types of leaf diseases. Our data comes from Kaggle; we process it first. The Kaggle dataset has corn plants similar to those in Indonesia, so we use this data with identification classes (Blight, Common rust, Gray leaf spot, and Healthy). The training data is 2000 images with 500 images for each class, and the testing data is 120 images with 30 images for each class. The evaluation results show that the classification process using the CNN method has an accuracy of 84.5%. The results we produced for identifying types of corn leaf disease still lack accuracy in their prediction, indicating the need to improve the CNN architecture model.

KEYWORDS CNN, Corn Leaves, Identification, Type of Disease

I. INTRODUCTION

The need for corn for food in Indonesia is increasing, and the government is trying to strengthen national food. Areas where corn is grown include North Sumatra, South Sumatra, Lampung, Central Java, East Java, Nusa Tenggara, North Sulawesi, South Sulawesi, and Maluku. And the government has developed a strategy to increase corn yields to meet Indonesia's demand and for export. Although some farmers are keen to increase rice production, several obstacles have arisen, such as disease and pest attacks on corn [1]. Farmers with experience in corn production are better equipped to handle the various diseases and pests that affect the crop. However, for novice and inexperienced farmers, recognizing the different types of corn diseases and pests requires information and guidance. Several previous studies have created a simulation system for identifying types of disease in corn [2]. An automatic system-based identification system simply inputs an image of a corn leaf and it will display information on the type of corn leaf disease.

The automatic system for identifying types of leaf disease uses machine learning methods with extraction feature methods from texture and color from RGB (Red Green Blue), HSV (Hue, Saturation, Value), L*a*b images [3]-[5]. On average, the automatic system for identifying leaf disease types using machine learning (Naive Bayes, K-Nearest Neighbor (k-NN), SVM (Support Vector Machine) has an accuracy of 70-90%. So there is a lot of research trying to increase accuracy for identification. The aim of developing a system for identifying leaf disease types is to help increase corn crop production. Because if diseases in corn can be controlled and recognized early, there is a chance of increasing crop production.

Research related to identifying types of leaf diseases using deep learning methods continues to develop, both using pretrained transfer learning architectures and creating your own architecture [1][6]-[11]. From previous research, the process of identifying types of corn leaf disease using the CNN (Convolutional Neural Network) method has an accuracy of above 90%. Therefore, to improve accuracy, we used CNN to classify the types of corn leaf diseases. We hypothesized that modifying the CNN architecture could improve the accuracy of detecting corn leaf disease types The purpose of this research is to create a system to detect types of corn leaf diseases. Differences between our research and previous ones [10], We create a CNN architecture with four times the number of convolution layers and our image size is 256x256.

II. LITERATURE REVIEW

Research related to identifying types of corn leaf diseases is included in table 1.

TABLE I
LITERATURE REVIEW

No	Method	Results
1	Feature extraction using texture (contrast value, correlation, energy, homogeneity, average, standard deviation) from L*a*b images, and classification process using k-NN [3]	Accuracy 73.3%
2	Using GLCM feature extraction from grayscale images, and HSV image feature values, then classified using k-NN [4]	70% Accuracy
3	Identifying types of leaf diseases using pretrained deep learning methods [8]	Validation data accuracy 88%
4	Identification of types of corn leaf disease from the mean features, standard deviation of RGB, HSV, and YCbCr images totaling 18 features, and 4 GLCM features (contrast, correlation, homogeneity, and energy), and the classification process with SVM [12]	99.5% Accuracy
5	Classification of types of corn leaf diseases using deep learning, input image size 32x32 [10]	94% Accuracy
6	Classification of types of corn leaf disease using ResNet50 and 224x224 image input [9]	98.3% Accuracy
7	Classifying types of corn leaf disease using HSV and GLCM (Angular Second Moment, Inverse Difference Moment, entropy and correlation) feature extraction, k-NN classification method [5]	84% Accuracy
8	Create a simulation system for identifying corn diseases, but based on 46 symptoms and 15 types of pest diseases [2]	Accuracy 73.3%
9	Identify types of corn leaf diseases using pretrained deep learningEfficientNetB0 architecture [11]	96% Accuracy
10	Identify types of corn leaf disease with CNN and 150x150 color image input [7]	94% Accuracy
11	Identify types of corn leaf disease with CNN and 50x50 image input [6]	99.9% Accuracy
12	The process of extracting the image features of corn leaves using CNN VGG-16 and 150x150 images, and then the process of classifying the types of corn leaf diseases using SVM, k-NN, and MLP [13]	SVM accuracy 93.8%, k-NN 92.1%, and MLP 94.4%
13	Classification of types of corn leaf disease with AlexNet and an input image size of 256x256 [1]	90% Accuracy

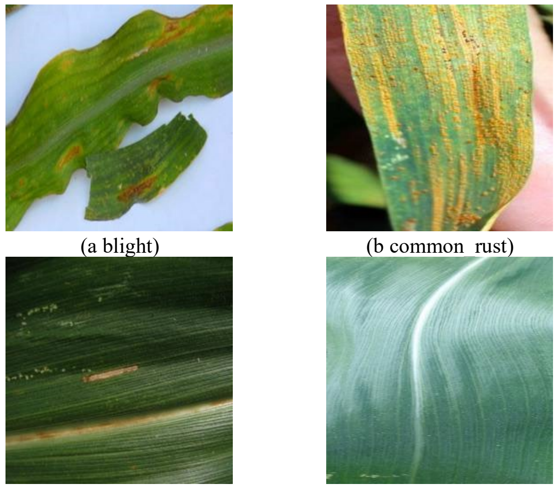
III. *METHOD*

A. DATASETS

Research data is taken from the Kaggle dataset [14]. We resize the image to 256x256. The distribution of training and testing data is presented in table 2. The number of classes in this study is four: blight, common_rust, gray_leaf_spot, and healthy, and each has the image data of corn leaves as shown in Figure 1. The data received from Kaggle was grouped by class in the form of folders. Images are stored in folders for each class.

TABLE II
DATASETS

No	Type	Training	Testing	Total
1	Blight	500	30	530
2	Common_rust	500	30	530
3	Gray_leaf_spot	500	30	530
4	Healthy	500	30	530
Total		2000	120	2120



(a blight) (b common_rust)
(c gray_leaf_spot) (d healthy)

FIGURE 1. Example of a corn leaf dataset

B. DEEP LEARNING

Convolutional Neural Networks are very similar to standard artificial neural networks, or units arranged in the form of an acyclic graph (a graph without any cycles in it), which can be represented as a collection of neurons. The difference between CNNs is that there are hidden layers that are only connected by a subset of neurons in the previous layer. This kind of connection allows CNN to implicitly understand features. The CNN architecture produces hierarchical feature extraction through the use of filters trained for a specific purpose. In the first layer, the focus is often on recognizing edges or color changes. In the second layer, attention shifts to shape recognition. Filters in subsequent layers are generally focused on learning details from partial parts of objects, both those seen on a small scale and those seen on a larger scale. The last layer in the CNN is used to identify the object as a whole. In this feature extraction layer, an image entered into the model will be encoded into numbers. This layer consists of two elements, namely the Convolutional layer and the Polling Layer. The convolution process in image data aims to produce features from the input image using filters. These filters have weights designed to detect object characteristics, such as curved lines, edges, or color changes. The activation function is an operation for recognizing nonlinearity and improving the representation of the model. The ReLU activation function is the output value of the neuron can be expressed as 0 if the input is negative. If the input value is positive, then the output of the neuron is the activation input value itself. Pollor subsampling is the process of reducing the size of image data or matrices with the aim of overcoming unnecessary fluctuations (overfitting) by the model. At this stage, the commonly used method is Max Pooling, which is known for using the area of the pooling input feature map to get the maximum value. This method is popular because it takes a region of the input feature map and extracts its maximum value. Flatten can convert all 2-dimensional arrays smoothed by feature maps into a single linear vector to become a fully connected input layer. A fully connected layer comes from the previous process of determining the features most related to a particular class. The function of this layer is to unite all nodes into one size.

Image Intensity Value

115	120	100	85	75	100	100	110	115	120
105	75	25	25	25	30	35	40	105	75
110	120	125	25	25	30	40	45	110	120
175	120	140	20	30	30	15	15	175	120
100	115	160	115	10	10	15	15	100	115
175	170	160	15	15	15	10	10	175	170
150	130	120	15	20	20	15	10	150	130
140	140	110	5	5	10	10	10	140	140
100	115	160	115	10	10	15	15	100	115
175	170	160	15	15	15	10	10	175	170

FIGURE 2. Image intensity value

Image Intensity Value

0	0	0	0	0	0	0	0	0	0	0	0
0	115	120	100	85	75	100	100	110	115	120	0
0	105	75	25	25	25	30	35	40	105	75	0
0	110	120	125	25	25	30	40	45	110	120	0
0	175	120	140	20	30	30	15	15	175	120	0
0	100	115	160	115	10	10	15	15	100	115	0
0	175	170	160	15	15	15	10	10	175	170	0
0	150	130	120	15	20	20	15	10	150	130	0
0	140	140	110	5	5	10	10	10	140	140	0
0	100	115	160	115	10	10	15	15	100	115	0
0	175	170	160	15	15	15	10	10	175	170	0
0	0	0	0	0	0	0	0	0	0	0	0

filter

1	0	1
1	0	1
1	0	1

Convolution Result

195	345	305	225	240	235	280	355	345	220
315	580	450	375	295	300	355	505	510	330
315	680	385	370	160	170	190	480	415	390
355	810	515	490	230	135	145	455	430	385
405	910	555	515	205	95	95	490	445	450
415	865	560	485	190	85	80	465	450	425
440	855	475	430	80	75	75	500	470	465
385	780	520	425	175	75	75	430	420	390
425	845	560	460	170	65	70	450	460	415
285	595	415	345	155	50	50	300	310	275

FIGURE 3. Example of convolution

Convolution Result

195	345	305	225	240	235	280	355	345	220
315	580	450	375	295	300	355	505	510	330
315	680	385	370	160	170	190	480	415	390
355	810	515	490	230	135	145	455	430	385
405	910	555	515	205	95	95	490	445	450
415	865	560	485	190	85	80	465	450	425
440	855	475	430	80	75	75	500	470	465
385	780	520	425	175	75	75	430	420	390
425	845	560	460	170	65	70	450	460	415
285	595	415	345	155	50	50	300	310	275

Max Pooling Result

580	450	300	505	510
810	515	230	480	430
910	560	205	490	450
855	520	175	500	470
845	560	170	450	460

FIGURE 4. Example of max pooling

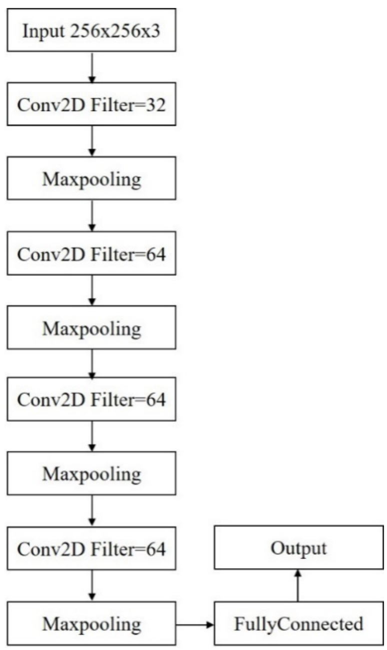


FIGURE 5. Proposed CNN Architecture

	blight	common rust	gray leaf spot	healthy
blight	24	0	5	1
common rust	2	27	0	1
gray leaf spot	8	0	21	1
healthy	0	0	0	30

FIGURE 6. Metric confusion results

Softmax activation transforms values from a numeric vector into a probability vector, where each possible value is proportional to the relative scale of each value in the vector. Each output value from softmax activation is interpreted as a probability in each class.

Image as in Figure 2. We took a sample of a particular part with a size of 10x10. We illustrate the convolution process of an image (Figure 3) with a size of 10x10 (Figure 2) and a filter size of 3x3. An image of size 10x10 has varying intensity values . It is then multiplied by a 3x3 filter, which results in the convolution of the same image of size 10x10, but the intensity value of each pixel is different. The convolution results take the maximum value for every 2x2 pixels to produce a max pooling process (Figure 4). The max pooling result is the best feature result from the maximum value, and the image size is reduced to 2 times smaller, for example, initially 10x10 to 5x5.

This research proposes a CNN architecture, as in Figure 5. We propose a convolution layer four times and a pooling layer four times, and the results of the feature extraction layer or convolution layer are trained. Input image of corn leaves measuring 256x256 in color. The feature map resulting from the convolution layer is 16x16 in size, meaning it has 256 feature maps.

A convolution layer is a layer that carries out the convolution process, namely multiplying each image pixel with a filter. The purpose of the convolution layer is to produce features from the image. The pooling layer is a layer that takes the best features from the convolution layer in order to represent the average image or the maximum. Our proposal uses a convolution architecture four times and pooling four times to make the extracted features more detailed. The more pooling layers, the more detailed the feature values obtained and caused the image size to decrease.

IV. RESULTS

We conducted training data experiments using optimizer={'rmsprop','sgdm'}, and learning rate={0.01;0.001}. We carried out training four times, each with 50 epoch iterations. Optimizer training results='rmsprop' with learning rate=0.001 in table 3 and optimizer training results='rmsprop' with learning rate=0.01 in table 4. Optimizer training results='sgdm' with learning rate=0.001 in table 5 and with a learning rate value = 0.01 in table 6. The results of the confusion metric evaluation are as in Figure 6. In the confusion metric evaluation results, identifying the type of leaf disease that has 100% accuracy is healthy. Moreover, the evaluation results of confusion metrics with low accuracy are gray leaf spots of only 63%.

TABLE III
TRAINING OPTIMIZER='RMSPROP' WITH LEARNING RATE 0.001

Epoch	*Iteration*	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:14	16.41%	2.0853	0.0010
4	50	00:10:52	67.19%	1.8346	0.0010
7	100	00:22:12	87.50%	0.3519	0.0010
10	150	00:33:31	77.34%	0.4690	0.0010
14	200	00:44:51	90.63%	0.2029	0.0010
17	250	00:56:11	95.31%	0.1122	0.0010
20	300	01:07:27	92.97%	0.2130	0.0010
24	350	01:18:43	96.09%	0.1001	0.0010
27	400	01:29:57	96.09%	0.0992	0.0010
30	450	01:41:13	99.22%	0.0410	0.0010
34	500	01:52:45	84.38%	0.4603	0.0010
37	550	02:04:23	100.00%	0.0211	0.0010
40	600	02:15:37	99.22%	0.0740	0.0010
44	650	02:26:51	100.00%	0.0091	0.0010
47	700	02:38:07	100.00%	0.0174	0.0010
50	750	02:49:30	99.22%	0.0430	0.0010

TABLE IV
TRAINING OPTIMIZER='RMSPROP' WITH LEARNING RATE 0.01

Epoch	*Iteration*	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:13	28.13%	2.3345	0.0100
4	50	00:11:34	54.69%	2.6889	0.0100
7	100	00:22:56	71.09%	0.6431	0.0100
10	150	00:34:28	64.84%	1.5427	0.0100
14	200	00:46:00	81.25%	0.4941	0.0100
17	250	00:57:31	85.16%	0.3802	0.0100
20	300	01:09:05	82.81%	0.3834	0.0100
24	350	01:20:37	93.75%	0.1867	0.0100
27	400	01:32:08	92.19%	0.1770	0.0100
30	450	01:43:32	85.16%	0.3998	0.0100
34	500	01:54:49	95.31%	0.1365	0.0100
37	550	02:06:08	96.09%	0.1779	0.0100
40	600	02:17:25	95.31%	0.1193	0.0100
44	650	02:28:42	95.31%	0.1163	0.0100
47	700	02:39:59	92.97%	0.1245	0.0100
50	750	02:51:17	97.66%	0.0610	0.0100

TABLE V
TRAINING OPTIMIZER='SGDM' WITH LEARNING RATE 0.001

Epoch	*Iteration*	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:13	16.41%	2.0853	0.0010
4	50	00:11:37	87.50%	0.2358	0.0010
7	100	00:23:25	90.63%	0.2112	0.0010
10	150	00:35:12	96.09%	0.1714	0.0010
14	200	00:46:39	100.00%	0.0492	0.0010
17	250	00:58:01	100.00%	0.0443	0.0010
20	300	01:09:20	100.00%	0.0279	0.0010
24	350	01:20:37	100.00%	0.0547	0.0010
27	400	01:31:54	100.00%	0.0232	0.0010
30	450	01:43:10	97.66%	0.1204	0.0010
34	500	01:54:23	100.00%	0.0401	0.0010
37	550	02:05:35	98.44%	0.0845	0.0010
40	600	02:16:48	93.75%	0.2289	0.0010
44	650	02:28:01	93.75%	0.1443	0.0010
47	700	02:39:14	96.09%	0.1229	0.0010
50	750	02:50:26	99.22%	0.0314	0.0010

TABLE VI
TRAINING OPTIMIZER='SGDM' WITH LEARNING RATE 0.01

Epoch	*Iteration*	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:12	21.09%	2.2360	0.0100
4	50	00:11:24	80.47%	0.6456	0.0100
7	100	00:22:56	86.72%	0.3969	0.0100
10	150	00:34:28	91.41%	0.1420	0.0100
14	200	00:46:02	99.22%	0.0703	0.0100
17	250	00:57:35	98.44%	0.0731	0.0100
20	300	01:09:04	99.22%	0.0532	0.0100
24	350	01:20:31	99.22%	0.0336	0.0100
27	400	01:31:57	99.22%	0.0493	0.0100
30	450	01:43:33	97.66%	0.0896	0.0100
34	500	01:55:13	95.31%	0.1032	0.0100
37	550	02:06:53	88.28%	0.2562	0.0100
40	600	02:18:33	97.66%	0.0858	0.0100
44	650	02:30:09	99.22%	0.0336	0.0100
47	700	02:41:35	100.00%	0.0172	0.0100
50	750	02:53:03	100.00%	0.0312	0.0100

Table 7 results from the average accuracy value when testing data by changing the optimizer= {'rmsprop', 'sgdm'}, and learning rate= {0.01; 0.001}. The result of changes in the optimizer that has the highest accuracy is 'sgdm', and the learning rate is 0.001.

TABLE VII
TESTING EVALUATION RESULTS

Optimizer	*Learning rate*	Testing Accuracy (%)
SGDM	0.001	87
SGDM	0.01	84
RMSPROP	0.01	82
RMSPROP	0.001	85

TABLE VIII
COMPARISON RESULTS OF RELATED RESEARCH

Method	Accuracy (%)
Our Proposal	84.5
AlexNet[1]	90
CNN[6]	99.9
CNN[7]	94
ResNet50[9]	98.3
Deep Learning[10]	94
EfficientNetB0[11]	96

Table 8 compares deep learning/CNN methods for recognizing corn leaf diseases. Our proposal has low accuracy compared with previous research.

V. CONCLUSION

We created a system for identifying leaf disease types using deep learning. Our dataset is sourced from Kaggle, and we only use 2120 images with four disease classes: blight, common rust, gray leaf spot, and healthy. The testing results for identifying types of corn leaf disease were 84.5%.

ACKNOWLEDGMENTS

Thank you to Lamongan Islamic University.

AUTHORS CONTRIBUTION

Rahul Firmasyah: reviewing, writing, and testing research; **Nur Nafiiyah:** revising and reviewing manuscripts and experiments;

COPYRIGHT

REFERENCES

[1] QN Azizah, "Classification of Corn Leaf Disease Using the AlexNet Convolutional Neural Network Method,"*sudo J. Tech. Inform.*, vol. 2, no. 1, 2023, doi: 10.56211/sudo.v2i1.227.
[2] MM Suhadi, MA Helmi, and W. Setiawan, "SIMULATION OF CLASSIFICATION OF PESTS AND DISEASES IN CORN

USING NAIVE BAYES,"*J. Simantec*, vol. 10, no. 1, 2022, doi: 10.21107/simantec.v10i1.11686.
[3] M. Khoirotul Ummah, Nur Nafi'iyah, "Identification of Corn Leaf Diseases Based on Texture Using K-Nn,"*Musamus J. Technol. Inf.*, vol. 02, no. 01, 2019, [Online]. Available: http://www.ejournal.unmus.ac.id/index.php/it/article/view/2419%0Ahttps://www.ejournal.unmus.ac.id/index.php/it/article/download/2419/1318.
[4] EH Rachmawanto and HP Hadi, "OPTIMIZATION OF FEATURE EXTRACTION ON KNN IN CLASSIFICATION OF CORN LEAF DISEASES,"*Dynamic*, vol. 26, no. 2, 2021, doi: 10.35315/dynamic.v26i2.8673.
[5] MA Setyawan, P. Kasih, M. Ayu, and D. Widyadara, "Classification of Corn Leaf Disease Based on HSV Color Space and Texture Features Using the K-NN Algorithm," in*National Seminar on Technology Innovation at UN PGRI Kediri*, 2022, pp. 67–72.
[6] AB Prakosa, Hendry, and R. Tanone, "Implementation of a Deep Learning Convolutional Neural Network (CNN) Model on Corn Leaf Disease Images for Plant Disease Classification,"*J. Educator. Technol. Inf.*, vol. 6, no. 1, 2023.
[7] D. Iswantoro and D. Handayani UN, "Classification of Corn Plant Diseases Using the Convolutional Neural Network (CNN) Method,"*J. Ilm. Univ. Batang Hari Jambi*, vol. 22, no. 2, 2022, doi: 10.33087/jiubj.v22i2.2065.
[8] MI Rosadi and M. Lutfi, "Identifying Types of Corn Leaf Disease Using Deep Learning Pre-Trained Models,"*J. Explor. IT!*, vol. 13, no. 2, 2021.
[9] IP Putra, R. Rusbandi, and D. Alamsyah, "Classification of Corn Leaf Disease Using the Convolutional Neural Network Method,"*J. Algorithm.*, vol. 2, no. 2, 2022, doi: 10.35957/algoritme.v2i2.2360.
[10] AD Nurcahyati, RM Akbar, and S. Zahara, "Classification of Disease Images on Corn Leaves Using Deep Learning with the Convolution Neural Network (CNN) Method,"*SUBMIT J. Ilm. Technol. Information and Science*, vol. 2, no. 2, 2022, doi: 10.36815/submit.v2i2.1877.
[11] F. Sarasati, F. Septia Nugraha, and U. Radiyah, "Utilization of Deep Learning Methods for Disease Classification in Corn Plants,"*J. Infortech*, vol. 1, no. 1, 2022.
[12] R. Suhendra, I. Juliwardi, and S. Sanusi, "Identification and Classification of Corn Leaf Disease Using Support Vector Machine,"*J. Technol. Inf.*, vol. 1, no. 1, 2022, doi: 10.35308/.v1i1.5520.
[13] J. Kusuma, Rubianto, R. Rosnelly, Hartono, and BH Hayadi, "Classification of Leaf Diseases in Corn Plants Using Support Vector Machine, K-Nearest Neighbors and Multilayer Perceptron Algorithms,"*J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, 2023, doi: 10.52158/jacost.v4i1.484.
[14] Kaggle, “Corn Leaf Disease Dataset.” https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset.