Rain Prediction Using Convolutional Neural Network (CNN) Method Based on Digital Image

Alya Syifa Ihsani, Anggunmeka Luhur Prasasti, Wendi Harjupa, Umar Ali Ahmad and Reza Rendian Septiawan

September 23, 2021

RAIN PREDICTION USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD BASED ON DIGITAL IMAGE

Alya Syifa Ihsani[1], Anggunmeka Luhur Prasasti[2], Wendi Harjupa[3]
Umar Ali Ahmad[4], Reza Rendian Septiawan[5]

[1,2,3,4,5] Computer Engineering, Electrical Engineering Faculty, Telkom University
[3] Center for Atmospheric Science and Technology, Indonesian National Institute of Aeronautics and Space

Abstract. Rain is a natural condition that occurs in Indonesia. Irregular rain patterns result in hydrometeorological disasters such as floods and landslides. These natural disasters often occur in Indonesia, not only causing material losses, but natural disasters also often take lives. To reduce the impact of natural disasters, it is necessary to predict rain which is one of the factors in natural disasters. Because of this reason, the rain prediction system was developed using the Convolution Neural Network (CNN) method in this research. One thousand cloud image data from the technology test center camera and space and atmospheric observations directed at the sky in the city of Garut were used for the training process. It consists of two categories, cloudy images and rain images, to build prediction models. The simulation process is carried out by inputting a cloud image through several processes such as preprocessing, feature extraction, and learning process, so this system can predict the rain the occurrence of rain. The tests are carried out to find the most optimal parameters in order to get the best accuracy. Obtained parameters such as data partition 80:20, learning rate 0.001, and epoch 50 resulting in accuracy reaching 98%. This system can predict the rain in the next hour well based on the accuracy results.

Keywords: Cloud image, convolutional neural network (CNN), image processing, rain prediction.

1 Introduction

Rain is a condensation process of water vapor that rises to the atmosphere and turns into water droplets that fall to the earth (Triatmodjo, 2008). Irregular rain patterns are caused by damage to the system in the hydrological cycle. This condition causes hydrometeorological natural disasters such as floods and landslides. These natural disasters often occur in Indonesia and cause environmental damage, material losses but can take lives (Hermon, 2012).

In this research, a system that can predict rain using a Convolutional Neural Network (CNN) algorithm based on digital images reduces the impact of these natural disasters. CNN is an algorithm that can process two-dimensional data (Suartika E. P, Wijaya, &

Soelaiman, 2016) by applying the Neural Network function, which can imitate the function of the human brain (Farhah, Prasasti, & Paryastro, 2021). so that the algorithm can learn its features in a complex image (Rohim, Sari, & Tibyani, 2019).

Currently, many studies have been built for various things using the CNN algorithm to study an object. For example, the design of the Arabic recognition application. This study aims to help pilgrims perform Hajj and Umrah to translate Arabic using smartphones (Rosyda, Irawan, & Prasasti, 2019). The expression classification for user experience testing for video games. This research is to determine user satisfaction in playing gameplay. It will be very helpful for their product developers. The system has succeeded in classifying various facial expressions such as angry, afraid, sad, happy, neutral, disgusted, and surprised in real-time (Isman, Prasasti, & Nugrahaeni, 2021). Handwritten Javanese Character Recognition, preserving Javanese script so that it can be used for daily communication, by utilizing the CNN algorithm to learn 20character classes to build a software that automatically displays handwritten Javanese language. (Dewa, Fadhilah, & Afiahayati, 2018). Transliteration of Hiragana and Katakana Handwritten, this research makes it easier for someone to learn Japanese, such as handwritten hiragana and katakana using a combination of CNN and SVM algorithms (Nugroho & Harjoko, 2021). Batik Motif Image Classification, designed to recognize various types and information of batik by classifying batik from motifs (Wicaksono, Suciati, Fatichah, Uchimura, & Koutaki, 2017).

In this research, a rain prediction system was built using a thousand cloud images consisting of two categories, cloudy images, and rain images, obtained from the Space and Atmospheric Observation and Technology Test Center (BUTPAA) directed at the sky in the city of Garut. The data used for the learning process using the CNN algorithm, generating a model used for prediction. From this process, the system can predict the rain in the next hour.

2 Data and Method

2.1 Data

The main data used in this study is cloud images which taken from the lapsed-time camera. Following is the procedure of data processing.

2.1.1 Cloud Image Dataset

We are using image data of 1000 images from two categories of cloudy and rainy with a size of 1080 x 1920 pixels. The difference between the two categories can be seen if the cloudy image is blue, bright and the clouds form separately. the image of the rain is gray, tends to be dark, and the clouds are lumpy. **Figure 1** is a sample of cloudy and rain image.

Figure 1. Cloud and Rain Image Data Sample

2.1.2 Pre-processing

The data size of 1920 x 1080 pixels is cropped and equated to 1180 x 690 pixels. Cropping is used to make an image into a uniform and this process reduces the dimensions of the image and removes objects that are not needed (Peryanto, Yudhana, & Umar, 2020) so that only cloud images can be seen. This process is used to avoid noise during the learning process. **Figure 2** is an example of an image that has the original image, and **Figure 3** has a cropped size.

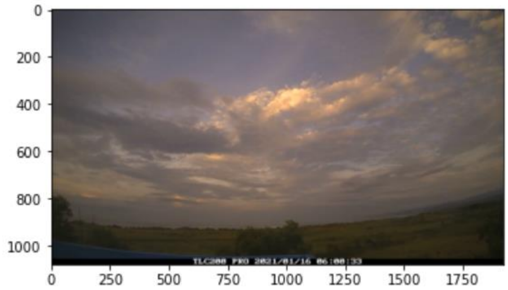


Figure 2. image size 1920 x 1080

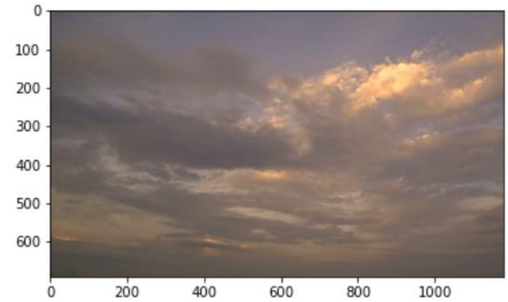


Figure 3. image size 1180 x 690

After changing the image size, the following process is contrast manipulation images using a computer (Panchani, Doshi, & Linbasiya, 2014) such as sharpen the image quality so that the colors in the image data will look clearer and brighter (Putra, Purboyo, & Prasasti, 2017). **Figure 4** and **Figure 5** are a comparison between the original image and image quality enhancement:

Figure 4. original quality image

Figure 5. image sharpening results

2.1.3 Training and Validation data

From the amount of image data obtained, it is divided into two parts randomly consisting of:

- Training data: data used to train image data using Convolutional Neural Network and produce a CNN model.
- Validation data: data that is used to avoid overfitting and is used to test the CNN model's performance.

Data are separated and grouped into different directories and names. as in **Figure 6** is an illustration of the grouping of image data

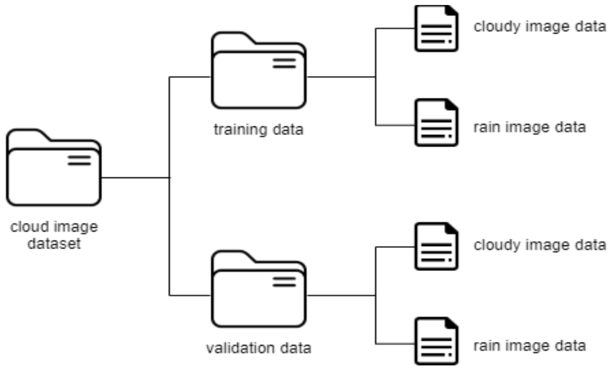


Figure 6. Illustration of image data grouping

2.2 Method

2.2.1 Research Design

This rain prediction system starts by inputting cloud image data into cloudy and rain images in png format. The first process is preprocessing, and the image size is changed to the same size. Second, they improve the image by sharpening the image quality so that the colors in the image data will look clearer and brighter. After that, the image data is divided into two parts, training data, and validation data. An augmentation process processes the training data to add images of different variations based on existing images. This training data is used for the training process using the Convolutional Neural Network (CNN) algorithm to obtain a specific input image feature and produce a CNN model. Next, the model will be tested and product recall, precision, and accuracy. The model that has been tested is used for the prediction process using new data and produces an output in the form of information on the occurrence of rain in the future. **Figure 7** shows an overview of the rain prediction system design:

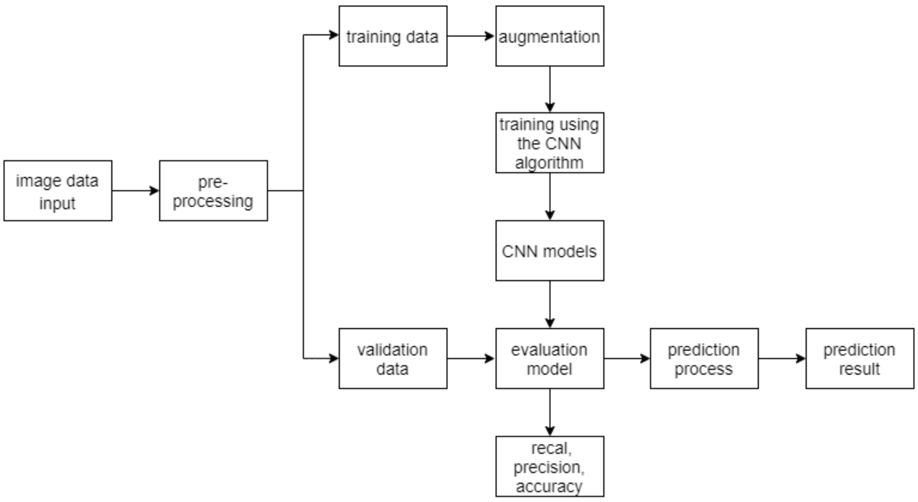


Figure 7. Rain prediction system design

2.2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is one type of Neural Network architecture that can provide information processed from a connected neuron. CNN imitates how the human brain can recognize objects seen (Farhah, Prasasti, & Paryastro, 2021). Convolutional Neural Network (CNN), the development of the Multi-Layer Perceptron (MLP), has a unique convolution and merging layer to study an object (Primartha, 2018). CNN has a high network depth and is widely applied to image data so that it belongs to the type of Deep Neural Network (Suartika E. P, Wijaya, & Soelaiman, 2016), is the development of the concept essential machine learning that uses layers that more (Thohari & Hertantyo, 2018). **Figure 8** shows two main parts to the Convolutional Neural Network (CNN) architecture, feature learning, and classification. The image also shows several processes for processing image data. For example, convolutional layer, pooling layer, and fully-connected layer. convolution is a process to extract essential features in the input image (Hidayat, Darusalam, & Irmawati, 2019)**.** The pooling layer serves to reduce the image that aims to increase the position invariance of the features. Can present data to be smaller, easier to process, and easy to control overfitting. The pooling process commonly used is max pooling (Suyanto, 2018).This research uses three convolution layers and two Neural Network layers as shown in **Table 1.**

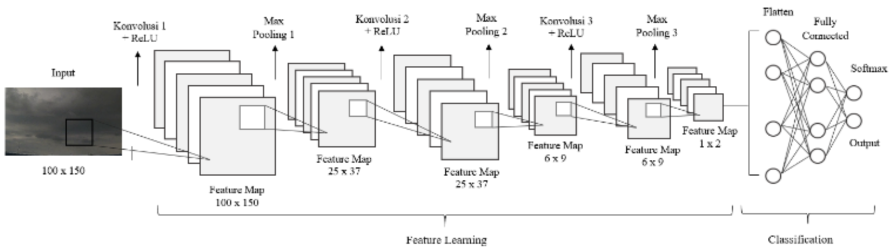


Figure 8. Convolutional Neural Network Architecture used in the system

Table 1. Layer of CNN used in the system

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 100, 150, 64)	4864
max_pooling2d_1 (MaxPooling2	(None, 25, 37, 64)	0
conv2d_2 (Conv2D)	(None, 25, 37, 128)	204928
max_pooling2d_2 (MaxPooling2	(None, 6, 9, 128)	0
conv2d_3 (Conv2D)	(None, 6, 9, 256)	819456
max_pooling2d_3 (MaxPooling2	(None, 1, 2, 256)	0
flatten_1 (Flatten)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 1024)	525312
dense_3 (Dense)	(None, 2)	2050

Total params: 1,819,266
Trainable params: 1,819,266
Non-trainable params: 0

2.2.3 Feature Learning

In the feature learning section, there are three convolution layers, and zero paddings are applied to add a value of 0 to all parts of the image input. Zero paddings are used to manipulate the output size in the convolution layer so that it does not decrease drastically and prevent a lot of information from being lost during the feature learning process.

The convolution layer has many layers, starting from 64, 128, and 256 layers. Kernel size 5x5 and stride or one-time kernel migration. This kernel is used for the feature ecstasy process of the image, which will move from the first value to the end and obtain a new pixel size. Next, enter the normalization process using the ReLU activation function. If there is a negative pixel value, it will be normalized to 0. Using a max-pooling size of 4x4, the pooling layer reduces the pixel value and produces a new pixel size. The output of the process is called a feature map. This stage continues according to the number of convolution layers used. To calculate the dimensions of the feature map, we use equation 1.

$$output = \frac{n + 2p - k}{s} + 1 \tag{1}$$

$\boldsymbol{n}$ is the length or height of the input image, $\boldsymbol{2p}$ zero paddings, $\boldsymbol{k}$ is the length or height of the kernel, and $\boldsymbol{s}$ is the displacement of the kernel or stride.

2.2.4 Classification

The classification section begins by converting a 2D feature map into a 1D vector, and this process is flattened. There is a neural network layer with layers 512 and 1024 connected by lines called weights using fully connected. A fully connected process use to transform data dimensions so that they can be classified linearly. (Pangestu, Rahmat, & Anggraeny, 2020)

This process produces output in the form of predictions of rain using the softmax activation function. Softmax can generate a label from the probability calculation process. From the resulting label, it is converted into a vector with a value between 0 and 1, and when added up, it will have a value of one (llahiyah & Nilogiri, 2018). Equation 2 is the Softmax Classifier.

$$f_j(z) = \frac{e^z j}{\sum_k e^z k} \tag{2}$$

The function f_j is the result of every jth element in the class output vector. The z function is a hypothesis given by the training model to be classified by the Softmax function.

3 Results

In this study, the test scenario was carried out to determine the model's performance generated from the learning process using the CNN algorithm. The scenarios used are data partition, learning rate and epoch. This test uses a confusion matrix to find out the three values of accuracy, precision and recall.

3.1 Confusion Matrix

Confusion Matrix is a method used to evaluate the algorithm's performance in classifying images from different classes (Rahman, Darmawidjadja, & Alamsyah, 2017). **Table 2** is a way to find out the values in the confusion matrix.

Table 2. Confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

The following variables serve to evaluate the classification algorithm.

- TP (True Positive) : positive data that is predicted correctly
- TN (True Negative) : correctly predicted negative data
- FP (False Positive) : negative data predicted as positive data
- FN (False Negative) : positive data predicted as negative data

3.2 Accuracy

Accuracy Parameters that can measure how accurate the model is in classifying new data correctly. To find out the accuracy is calculated using equation 3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \, x \, 100 \tag{3}$$

3.3 Precision

Precision is a parameter that can measure the accuracy of the requested data with the prediction results provided by the model. To find out the precision is calculated using the equation 4.

$$Precision = \frac{TP}{TP + FP} \, x \, 100 \tag{4}$$

3.4 Recall

Recall parameter that calculates the success of the model in predicting the entire object correctly. To find out the recall is calculated using the equation 5.

$$Recall = \frac{TP}{TP + FN} \, x \, 100 \tag{5}$$

3.5 Data Partition Testing Scenario

This test has different data comparisons of 90:10, 80:20, 70:30, 60:40 and 50:50. From this test, the highest accuracy is obtained at 98% on the 80:20 data partition. Estimated time during training 15 minutes 46 seconds. Table 3 show the scenario results of data partition testing.

Table 3. Scenario Results of Data Partition Testing

Epochs 50, Adam's optimization (lr = 0.001)				
Data Partition	Precision	Recall	Accuracy	Time
50:50	83 %	99 %	89 %	11 min 2 s
60:40	93 %	96 %	95 %	12 min 39 s
70:30	97 %	96 %	97 %	13 min 36 s
80:20	97 %	99 %	98 %	15 min 46 s
90:10	93 %	100 %	96 %	16 min 48 s

From the results obtained, the data partition can affect the level of accuracy, the highest accuracy is 98% on the 80:20 data partition. the training data used is more than the validation data, so the system can learn more data variations. After that, there is sufficient validation data to avoid overfitting, and during training the accuracy increases steadily.

3.6 Learning Rate Testing Scenario

This test compared the accuracy results obtained from the learning rate test used 0.01, 0.001, and 0.0001. The highest accuracy is 98%. The estimated training time is 14 min 35 seconds. **Table 4** shows the learning rate scenario result.

Table 4. Learning Rate Scenario Results

80:20 data partition, epoch 50, Adam optimization				
Learning Rate	Precision	Recall	Accuracy	Time
0.01	24 %	23 %	24.5 %	15 min 46 s
0.001	98 %	99 %	98.5 %	14 min 35 s
0.0001	97 %	99 %	98 %	16 min 42 s

From the results obtained, the learning rate can affect accuracy, where the learning rate is the number of steps taken in the training process. The size of the learning rate used can affect the speed during the training process. It can be seen that the 0.01 learning rate produces the lowest accuracy because the step when training is too extensive, so the system cannot learn the object correctly. While the highest accuracy is obtained at a learning rate of 0.001 with an accuracy of 98.5%, this is due to the small training steps to learn more specific objects.

3.7 Epoch Testing Scenario

This test compares the accuracy results obtained from the 10, 20, 30, 40, and 50. The highest accuracy was obtained at 98%, epoch of 50. The estimated time during training was 15 min 46 seconds. The epoch result can be seen in **Table 5**.

Table 5. Epoch Scenario Results

80:20 data partition, Adam optimization, learning rate 0.001				
Epoch	Precision	Recall	Accuracy	Time
10	92 %	97 %	93 %	3 min 10 s
20	99 %	92 %	95.5 %	6 min 10 s
30	98 %	93 %	95.5 %	9 min 31s
40	88 %	99 %	90.5 %	12 min 31 s
50	97 %	96 %	98 %	15 min 46 s

From the results obtained, epoch can affect the level of accuracy, epoch is the process of learning training data in one round. From this test, the highest accuracy is 98% at epoch 50. From this test, many epochs can increase accuracy and when training is not overfitting.

3.8 Model Convolutional Neural Network (CNN)

After conducting an evaluation using the architecture that has been designed, the data partition is 80:20, the learning rate is 0.001, and the epoch 50 results in an accuracy of 98%. The model is saved using the "*model.save(model_name*" function. The saved model can be used for the prediction process. **Figure 10** shows the convolution neutral network model.

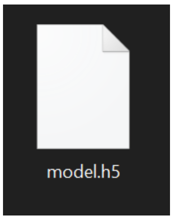


Figure 10. Convolutional Neural Network Model

3.9 Graphical User Interface (GUI)

In this study, a Graphical user interface (GUI) was designed to display a rain prediction system visually. There is a feature to input a new image, pressing the choose file button. After that, the image size is changed, and the image quality is sharpened. the system will predict using the CNN model that has been tested and obtain the best accuracy, and we can find out rain information in the next hour. **Figure 11** is the Graphical user interface (GUI) of this rain prediction system.

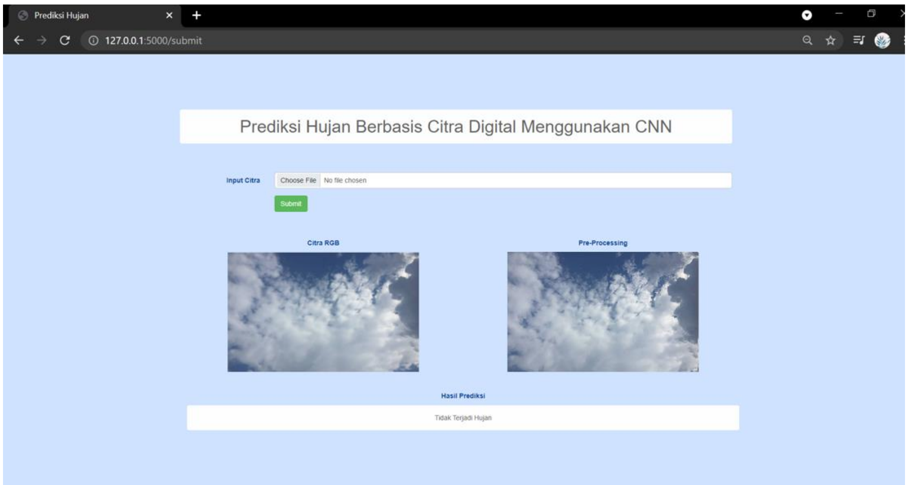
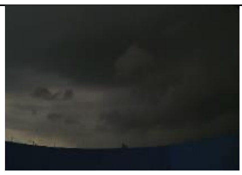


Figure 11. GUI of the rain prediction system

3.10 Cloud prediction simulation

Table 6 shoes the prediction simulation using cloud images in different conditions which are inputted using the GUI.

Table 6. prediction simulation predictions

No	Sample Images	System Prediction In The Next Hour	Validation
1		Predict No Rain	Correct
2		Predict Rain	Correct
3		Predict No Rain	Correct

4		Predict Rain	Correct
5		Predict No Rain	Correct

Sample image of cloud number 1, the clouds are grey, and the sky is clear. This cloud condition does not cause rain. Sample image of cloud number 2, the cloud is grey, dark, and the cloud's shape is lumpy. This cloud condition causes rain. Sample image of cloud number 3, the cloud is white, the sky is clear, and the cloud is separate. The cloud condition does not cause rain. Sample image number 4, the clouds are grey, and the shape of the shadows is lumpy, but the clouds do not look dark because they are illuminated by the sun, even though these cloud conditions cause rain. Sample image of cloud number 5, the cloud is white, the sky is clear, and the cloud forms are separated. This cloud condition does not cause rain. Of all the cloud image samples used, the system can correctly predict the rain conditions in the next hour.

4 Conclusions

From the design of this rain prediction system using the Convolutional Neural Network algorithm, it can be concluded:

- The Convolutional Neural Network algorithm can predict rain in the next hour, with an accuracy of up to 98%
- Combination of parameters that produces the best accuracy. Parameters obtained, data partition 80:20, learning rate 0.001 and epoch 50.

References

Dewa, C. K., Fadhilah, A. L., & Afiahayati. (2018). Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 83-94.

Farhah, A., Prasasti, A. L., & Paryastro, M. W. (2021). Implementasi Recurrent Neural Network dalam Memprediksi Kepadatan Restoran Berbasis LSTM. *Jurnal Media Informatika Budidarma, v*(2), 524-531.

Hermon, D. (2012). Mitigasi Bencana Hidrometeorologi. Padang: UNP Press .

Hidayat, A., Darusalam, U., & Irmawati. (2019). DETECTION OF DISEASE ON CORN PLANTS USING CONVOLUTIONAL NEURAL NETWORK METHODS. *Jurnal Ilmu Komputer dan Informasi*, 51-56.

Isman, F. A., Prasasti, A. L., & Nugrahaeni, R. A. (2021). Expression Classification For Uset Experience Testing Using Convolutional Neural Network. *International Conference on Artificial Intelligence and Mechatronics System (AIMS).* Universita delgi .

llahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Jenis Tumbuhan Berdasarkan CNN. *JUSTINDO (Jurnal Sistem & Teknologi Informasi Indonesia), III*(2), 49-56.

Nugroho, N. E., & Harjoko, A. (2021). Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 221-232.

Panchani, A., Doshi, H., & Linbasiya, N. (2014). PREDICTION OF RAINFALL USING IMAGE PROCESSING. *International Journal of Advance Engineering and Research* , 61-67.

Pangestu, R. A., Rahmat, B., & Anggraeny, F. T. (2020). IMPLEMENTASI ALGORITMA CNN UNTUK KLASIFIKASI CITRA LAHAN DAN PERHITUNGAN LUAS. *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, 166-174.

Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation . *Journal of Applied Informatics and Computing (JAIC)*, 45-51.

Primartha, R. (2018). *Belajar Machine Learning.* Bandung: Informatika Bandung.

Putra, R. D., Purboyo, T. W., & Prasasti, A. L. (2017). A Review of Image Enhancement Methods. *International Journal of Applied Engineering Research , 12*(23), 13596-13603.

Rahman, M. F., Darmawidjadja, M. I., & Alamsyah, D. (2017). KLASIFIKASI UNTUK DIAGNOSA DIABETES MENGGUNAKAN KLASIFIKASI UNTUK DIAGNOSA DIABETES MENGGUNAKAN. *JURNAL INFORMATIKA*, 36-45.

Rohim, A., Sari, Y. A., & Tibyani. (2019). Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, III*, 7037.

Rosyda, S. R., Irawan, B., & Prasasti, A. L. (2019). Design of arabic recognition application using Convolutional Neural Network . *Journal of Engineering And Applied Sciences , 14*(19), 6982-6990.

Suartika E. P, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. *Jurnal Teknik ITS, V*, A65.

Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjutan.* Bandung: Informatika Bandung.

Thohari, A. N., & Hertantyo, G. B. (2018). Implementasi Convolutional Neural Network untuk Implementasi Convolutional Neural Network untuk. *Electrical*

Engineering, Telematics, Industrial Technology, and Creative Media, (pp. 50-55). purwokerto: INSTITUT TEKNOLOGI TELKOM PURWOKERTO.

Triatmodjo, B. (2008). *Hidrologi Terapan.* Yogyakarta: Beta Offset.

Wicaksono, A. Y., Suciati, N., Fatichah, C., Uchimura, K., & Koutaki, G. (2017). Modified Convolutional Neural Network Architecture for Batik Motif Image Classification. *IPTEK, Journal of Science,,* 26-30.