SAGA: Journal of Technology and Information Systems
Vol 3, Issue 2, May 2025, Page 415-423
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v3i2.511

Classification of Ceramic Roof Tiles Using the CNN Method

Achmad Danu Zakaria, Ade Eviyanti*, Metatia Intan Mauliana, Azmuri Wahyu Azinar

Department of Informatics, Universitas Muhammadiyah Sidoarjo, Sidoarjo, Indonesia

Corresponding Author: adeeviyanti@umsida.ac.id

Abstract. The research on tile classification using Convolutional Neural Network (CNN) aims to improve and address issues in the sorting process within the tile manufacturing industry. The accuracy level in manual sorting processes is very low due to errors caused by visual limitations and physical fatigue. By leveraging the capabilities of Convolutional Neural Network (CNN), a model was developed to classify tiles. This research involved several processes, including literature review, dataset collection, dataset splitting, preprocessing, Convolutional Neural Network (CNN) design, training, testing, and result evaluation. The study used 69 tile images divided into three classes: KW 1, KW 2, and KW 3. The results of testing the Convolutional Neural Network (CNN) on tile classification using 100 epochs with a data split of 90% training and 10% validation yielded an accuracy rate of 100%.

Keywords: Classification, Deep Learning, Ceramic Roof tile, Image Processing

1 Introduction

The construction and building industry plays an important role in infrastructure development, one of which is roof tiles. Tile is one type of roof covering that is commonly used, serves as a protection from sunlight and rainwater, and has the ability to improve the aesthetic aspects of the house by adding beauty to its appearance[1] . There are various types of tile, such as ceramic tile, concrete tile, metal tile, asphalt tile, wood tile, and slate tile. Each type of tile has its own uniqueness and characteristics, including weather resistance, financial aspects, durability, and aesthetics .[2]

Based on data obtained from the Central Bureau of Statistics of Malang Regency, ceramic tile, metal tile, and clay tile are the most dominant roof types used by buildings, with a percentage reaching 92.48% in 2020. The preference for these types is due to their structural strength, resistance to various weather conditions, and higher aesthetic value compared to other types of roof tiles. In addition, the use of this roof tile is also driven by the abundant availability of raw materials and good production technology, resulting in a more efficient and economical product.

In the roof tile production process, the sorting stage is one of the crucial final processes to determine the level of product quality. The quality of roof tiles is assessed based on several parameters, including color, shape integrity, and texture. Currently, the sorting process of kermaik roof tiles is still done manually, with the resulting accuracy level being very low and inconsistent, due to visual limitations, physical fatigue, and subjective perceptions of each individual. Therefore, a more technological approach is needed to improve accuracy and consistency. The application of automation in the industrial world aims to overcome the potential for human error. This is motivated by the natural limitations of humans, who tend to experience fatigue and lack of precision when engaged in work that requires consistency over a long period of time .[3]

Technology is developing very quickly and can be accessed easily and at any time[4] . One of the current technological developments is image processing through the use of *deep learning*[5] . In *deep learning, Convolution neural Network (CNN)* is the method of choice that is often used in research[8] . The accuracy of the accuracy has made it the most frequently applied technique among various other methods in the world of image processing[9] . *Deep learning* is used in image processing to improve the efficiency of object recognition with high speed and accuracy, and is able to process large-scale data effectively[6] . Image processing is becoming a form of technology that not only solves the challenges in image manipulation, but also opens the door for the use of images in further research[7] . With the advantages of this technology, research on tile classification emerged as the latest development utilizing CNN.

Several previous studies have used *Convolution neural Network (CNN)* for object classification. One of the only studies for the classification of puppet images whose research results in accuracy, *loss*, precision, and *recall* values reaching 97%, 2%, 93%, and 87% respectively[10] . Another study of image classification of rice types resulted in an average value of *recall, precision*, and *F1-score* for each type, namely C4 rice 96.3%, glutinous rice

100%, brown rice 100%, and dragon rice 96%[11] . The next research is the classification of egg images to determine the quality of the accuracy results obtained by 80%, from 6 test data only 1 data that does not match the results[12] . The next research is the classification of fire and non-fire images, using 950 images to produce a model with an accuracy rate of 98.8% and a loss of 2.6% . [13]

Some studies that use *Convolution neural Network (CNN)* for classification above produce a fairly good percentage of the final results, and it can be concluded that this method shows good performance in image classification. Based on these findings, it is expected that tile classification research using CNN produces a model with a good final value.

2 Methods

This research is organized and designed with structured and systematic steps, ensuring that this research produces valid data in accordance with the desired research objectives. The software program used is google colab while for dataset storage using google drive[14] . In addition, this research is supported by hardware and software with specifications as in table 1 and table 2.

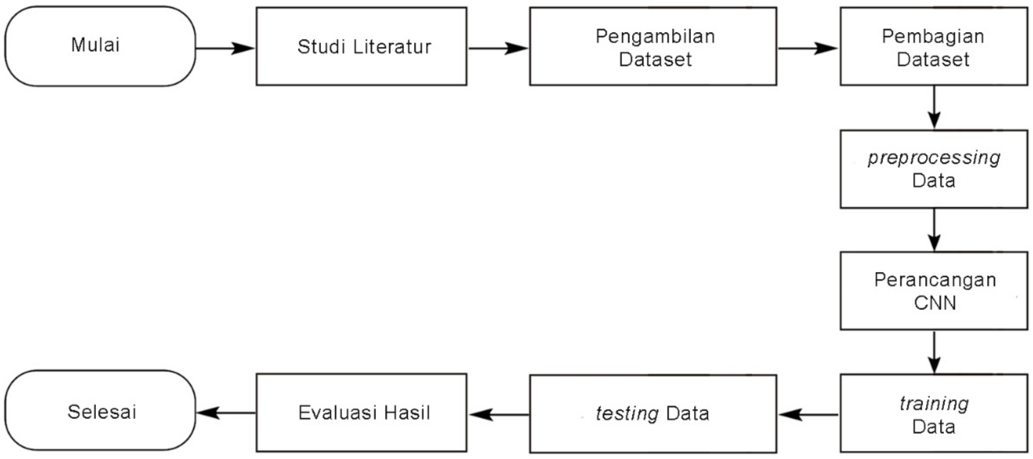


Figure 1. Research Flow

Figure 1 shows the steps including literature study, dataset retrieval, dataset sharing*, preprocessing, Convolution neural Network (CNN)* design, *training*, *testing*, and result evaluation.

Table 1. Hardware

No	Aspect	Specifications
1	*Processor*	Intel(R) Core(TM) i3-5010U CPU @ 2.10GHz 2.10 GHz
2	*Memory*	4 GB RAM
3	*Storage*	512 GB HDD
4	*Operating System*	Windows 10 Pro

Table 2. Software

No	Aspect	Software
1	Programming Language	Python
2	*Software Program*	*Google Collaboratory*

2.1 Literature Study

This initial stage was carried out by looking for information from several sources such as books and journals in accordance with the issues raised[10] . This stage is carried out to facilitate researchers in developing and finding new references. This literature study includes basic theory research, research methods that have been used before, and the latest findings in the field of image processing and the application of *Convolution neural Network (CNN)* in object classification.

2.2 Dataset Collection

In the dataset collection process, the image of the roof tile is taken directly using a camera in jpg format. By taking data from the right source, ensuring sufficient representation of each type of tile in the dataset used for the purposes of *training* and *testing* classification models.

Table 3. Ceramic Tile Image Dataset

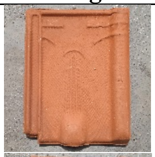
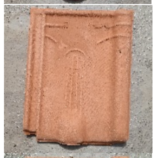
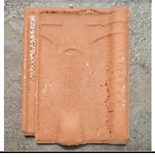
No	Type	Data	Classification	Image
1	KW 1	23 Images	Bright color and no crack	
2	KW 2	23 Images	Faded color and without crack	
3	KW 3	23 Images	Faded color and cracks	

Table 3 above is a grouping of dataset classes divided into three classes, each class has a label and representation. KW 1 representation is bright color and without cracks, KW 2 is faded color and without cracks, while KW 3 is faded color and with cracks. For each class, 23 images were sampled.

2.3 Dataset Distribution

After collecting the tile image dataset, the dataset is divided into 2 datasets, namely *training* and *testing* datasets. *The training* dataset is used to train the *Convolution neural Network (CNN)* model in order to recognize patterns in tile images. The *testing* dataset is used to objectively test the final performance of the model, providing a measure of accuracy and consistency in classifying ceramic roof tiles in real-world situations. This sharing of datasets helps prevent *overfitting* and ensures that the model can make good predictions on data that was not used during *training*.

Table 4. Dataset Split

Dataset	
Training	*Testing*
60	9

Table 4 explains the division of 69 datasets, 60 *training* data and 9 *testing* data*, the* datasets that have been divided will be grouped into three folders, namely the KW 1 folder, the KW 2 folder, and the KW 3 folder. This stage aims to facilitate labeling when *training* and *testing* are carried out.

2.4 Data Preprocessing

Before the ceramic tile image data is processed, pre-processing is first carried out, this stage is very important because the image images that have been collected have a variety of non-uniform pixel sizes. The pre-processing stage aims to solve various problems that can interfere with the results of the data process[4] . Some steps to perform pre-processing, namely

a. Data Augmented
   Data augmentation is used to artificially expand the size of the training dataset by creating modified versions of the images in the dataset. This technique prevents overfifting and improves the generalization ability of the model.

b. Data Generation
   The 'flow-from_directiry' method is used to generate batch image data with real-time augmentation, ensuring efficient use of memory.
c. Image Preprocessing for prediction.
   To perform prediction, the input image is preprocessed by converting it to an array, and normalizing the pixel values. The data preprocessing step implemented is normalizing the pixel size of the original image so that each image has similar pixel dimensions .[13]

2.5 Convolution Neural Network (CNN) Design

The choice of *Convolution neural Network (CNN)* method was driven by the nature of the dataset consisting of image data and also due to the significant amount of dataset. This decision was taken considering the need for an approach that not only offers fast classification performance, but also guarantees an optimal level of accuracy. The structure of *Convolution neural Network (CNN)* has several stages namely *convolution layer, pooling layer,* and *fully connected layer* .[15]

a. Feature Learning
   1. Convolution Layer
      The convolution process is done for the purpose of extracting the input image[16] . The convolution process involves shifting the kernel horizontally and vertically with a shift size referred to as the stride parameter. ReLU activation function is applied to convert negative values to 0.
   2. Pooling Layer
      Pooling layer is a process used to reduce the size of image data with the aim of improving color invariance[16] . One of the commonly used operations is max pooling, which divides the output of the convolution layer into several small grids and takes the maximum value from each grid to compose the reduced image metric. In short, the max pooling process is focused on the dimensions of the image to improve the efficiency of the calculation and avoid the possibility of overfitting.
b. Classification Stage
   1. Flatten Layer
      The flatten operation is performed to reshape the feature map into a vector, so that it can be used as input for the Fully Connected Layer. Fully Connected Layer is a multilayer perceptron with a predetermined number of neurons in the hidden layer. The dropout method is applied to partially disable the connection to each neuron, avoiding overfitting.

2.6 Training Data

Training process is the stage where *Convolution neural Network (CNN)* is trained to achieve higher accuracy in classification[4] . The data *training* stage of the *Convolution neural* network *(CNN)* is performed using an image dataset that is grouped into three classes. To increase variety and prevent *overfitting*, the dataset was augmented in *real-time* using *ImageDataGenerator*. The applied augmentations include *rescaling, shear, zoom,* and *horizontal flipping*.

The developed *Convolution neural Network (CNN)* model has multiple convolution layers with *ReLU* activation function, followed by a *pooling* layer to reduce the dimensionality of the features. After the convolution layer, the data is processed through a *fully connected* layer with *softmax* activation to output classification probabilities. *The training* process takes place using a split of 90% training data, 10% test data and 80% training data, 20% test data with *epochs* of 10, 50, and 100 and *a batch* size of 32, using the *categorical cross-entrophy loss* function and *adam optimizer*.

2.7 Testing Data

After the *training* stage, the *Convolution neural Network (CNN)* model goes through a *testing* process using a separate dataset, which was never used in the *training* process. This *testing* dataset has also passed the normalization process to ensure data consistency in data processing. This *testing* stage aims to provide an objective picture of the model's ability to classify tile types on data that is completely new and has never been seen in the model before.

The *testing* process is carried out by calculating the model's prediction of the images in the *testing* dataset. Each image is processed through the model to get its class prediction. This prediction result is then compared with the actual label to calculate its accuracy.

2.8 Evaluation of Results

The performance of the model is evaluated based on several metrics including accuracy, precision, *recall*, and *F1-score*, which are calculated to provide a more detailed picture of the model's classification performance for each class.

3 Results and Discussion

3.1 Training Results

The dataset *training* stage uses *Google Collaboratory* with 6 scenario experiments and in each experiment a *restart session* is carried out for accurate results. This process produces a new *training* file model stored on *Google Drive* with the name tile_clasification_scenario.h5. Table 5 is the result of *testing* with different iterations.

Table 5. Training Results

Training Data	Validation Data	*Epoch*	Loss	Accuracy
90%	10%	10	0.7231	66.66%
		50	0.3970	83.33%
		100	0.3591	83.33%
80%	20%	10	0.6074	66.66%
		50	0.4978	83.33%
		100	0.4927	83.33%

From the table 5 presented, it can be concluded that using the right combination and split results in high accuracy and low loss in the scenario with 100 *epochs* and 90% test data split and 10% validation, achieving the highest accuracy of 83.33% at the 85th epoch and the lowest loss of 0.3591 at the 29th epoch. This phenomenon shows that the relationship between loss and accuracy during model training is not always linear. The 29th epoch, despite a significant decrease in loss, did not show optimal accuracy. This indicates that the model experienced overfitting, where the model overfitted itself to the training data and was less able to generalize well to the validation data. On the other hand, the 85th epoch may represent the point where the model successfully balances its ability to recognize patterns in the validation data, even though the loss at that epoch is not at its lowest.

This discrepancy highlights that evaluation metrics such as loss and accuracy do not always align during training. Although the 29th epoch showed a low loss value, the accuracy of the model increased at the 85th epoch, which may indicate that the model successfully provided more precise predictions on the validation data. Therefore, although the loss value may be reduced at a certain epoch, the accuracy of the model may reach its highest point at a different epoch, reflecting the model's ability to generalize better.

This finding reminds us of the importance of monitoring various evaluation metrics to get a more comprehensive picture of model performance, as well as the need for a more in-depth evaluation approach to assess overall model effectiveness.

3.2 Testing Results

Based on the results of *training*, the *testing* process decided to use iterations of training data and validation data with *epoch* 100 with a ratio of 90% and 10%.

Table 6. *Testing* Results

No	Image Label	Class	Output	Status
1	Test1	KW3	KW3	Success
2	Test2	KW2	KW2	Success
3	Test3	KW1	KW1	Success
4	Test4	KW3	KW3	Success
5	Test5	KW2	KW2	Success
6	Test6	KW1	KW1	Success
7	Test7	KW1	KW1	Success
8	Test8	KW3	KW3	Success
9	Test9	KW2	KW2	Success

Based on Table 6, the results of *testing the* new tile image get the results of 9 successful images 0 failed images, for the calculation of accuracy calculated at the evaluation stage. The results of the tile classification screenshot on the *testing* data can be seen in Figure 2.

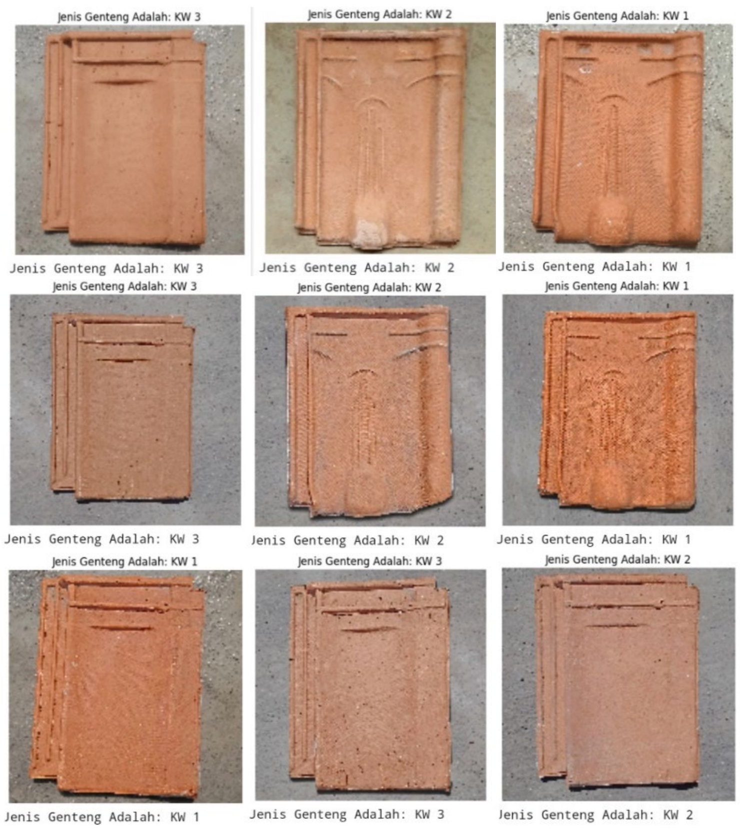

Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2
Jenis Genteng Adalah: KW 1
Jenis Genteng Adalah: KW 3
Jenis Genteng Adalah: KW 2

Figure 2. Test Data Classification Results

Figure 2 shows that the class representation on the tile is in accordance with the results of each classification.

3.3 Evaluation Results

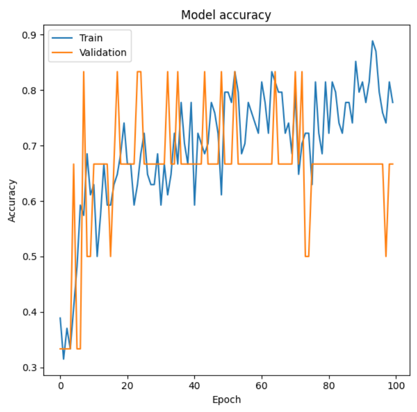


Figure 3. Accuracy Graph on *Training*

Figure 3 displays the accuracy graph from *training*, showing the increase in accuracy until the seventh *epoch*, showing a sharp increase in both training and validation data until it reaches a peak around the seventh epoch. This sharp increase indicates that the model managed to learn the basic patterns in the data well in the early stages of training, and was able to apply the patterns to the validation data effectively. There is a drop in accuracy at the ninth epoch in both sets, followed by significant fluctuations, this drop could be caused by several factors, including too high a learning rate, noise in the data, or temporary overfitting. The model may jump local minima or fit too specifically to the training data at this stage.

SAGA: Journal of Technology and Information Systems
Vol 3, Issue 2, May 2025, Page 415-423
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v3i2.511

The accuracy increases again after the initial fluctuations, reaching a peak again around the sixteenth epoch, this increase in accuracy indicates that the model may have adjusted the hyperparameters or discovered new patterns in the data, although the model is not yet fully stable in its learning. Accuracy on training data began to stabilize until around the seventieth epoch, while accuracy on validation data showed a downward trend. The stability of the training accuracy indicates that the model has reached convergence on the training data, but the decrease in accuracy on the validation data indicates overfitting, where the model is too specific on the training data.

The accuracy of the validation data continues to decrease until the last epoch, this decrease in accuracy is a clearer indication of overfitting, where the model cannot generalize well on validation data that was not seen before.

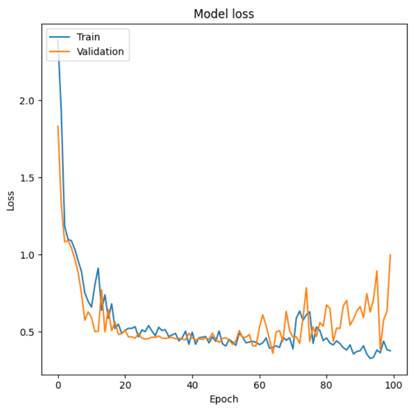


Figure 4. *Training* Loss Graph

After that, in Figure 4 the loss graph of the loss value decreases rapidly in both sets (training and validation) during the initial epoch. This sharp decrease shows that the model quickly learns from the data, indicating that the model is able to identify basic patterns in the data at an early stage of training. The loss values on both sets began to stabilize around the 20th epoch, although there were small fluctuations. This stabilization of the loss indicates that the model is approaching convergence. The small fluctuations may reflect normal variations in the learning process or the impact of the hyperparameters used.

An increase in the loss value on the validation set occurs after the 70th epoch, this increase in loss on the validation data indicates possible overfitting, where the model has overfitted itself to the training data and cannot generalize well to the validation data. The increasing loss fluctuation indicates instability in the model's performance on the data that was not seen before.

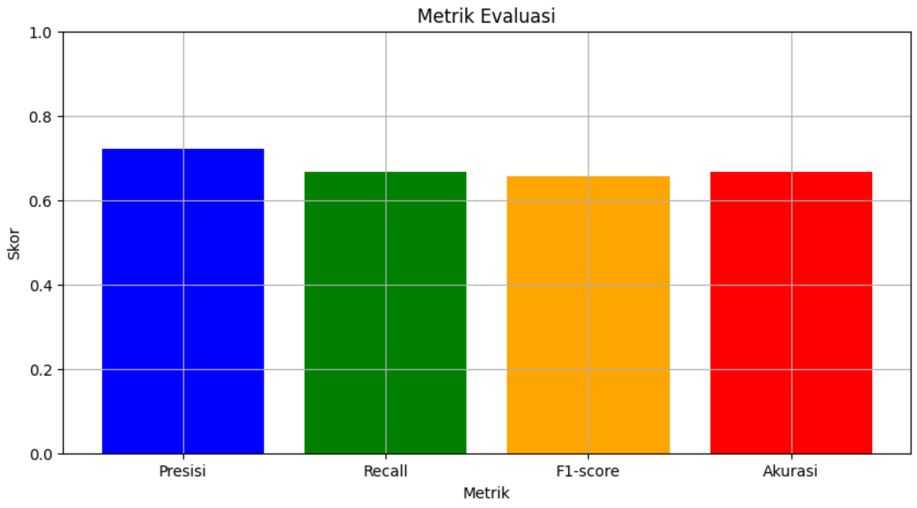


Figure 5. Metric evaluation diagram

In Figure 5, the model evaluation results diagram shows an accuracy of 66.67%, which indicates the proportion of correct predictions out of the whole data. The model precision is 72.22%, indicating that most of the positive predictions are correct. The model recall reached 66.67%, indicating that the model was able to identify two-thirds of the positive cases. The F1 Score of 65.56% reflects a fairly good balance between precision and recall, although there is room for improvement.

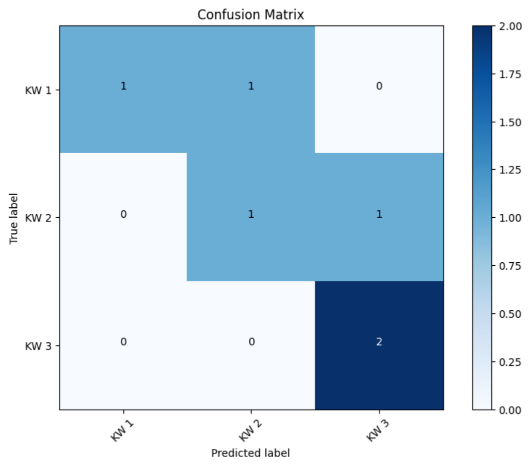


Figure 6. Confusion Matrix

In Figure 6 *Confusion matrix* for the Kw 1 class, the model has a precision of 100%. This means that every time the model predicts Kw 1, the prediction is correct. However, the recall for Kw 1 only reaches 50%, which indicates that the model is only able to detect 50% of all data that should be classified as Kw 1. With this combination, the F1 score for Kw 1 is 0.67, which indicates an imbalance between precision and recall.

For Kw 2, the model precision is 50%, indicating that only half of the Kw 2 predictions are correct. Recall is also at 50%, which means that the model can only detect 50% of the data that are actually Kw 2. The F1 score for Kw 2 is 0.5, reflecting an imbalance in precision and recall that is less than optimal.

In contrast, for Kw 3, the model shows excellent performance. The precision of Kw 3 is 100%, meaning that every prediction of Kw 3 is correct. In addition, the recall of Kw 3 also reached 100%, indicating that the model successfully detected all data belonging to the Kw 3 category. The F1 score for Kw 3 was 1, indicating that the model was highly effective in identifying Kw 3 with optimal precision and recall.

Overall, the model succeeds very well in identifying Kw 3, but its performance is less adequate for Kw 1 and Kw 2, especially in terms of recall.In the *testing* results, 9 successes and 0 failures were obtained, for evaluation of the *testing* results using equation calculations.

$$Akurasi = \frac{9}{9} \times 100\% = 100\%$$

From the results of the above calculations, it can be concluded that in *testing* data for the level of accuracy using test data as many as nine of the 3 classes reached 100%.

4 Conclusion

The model performed very well in testing with a perfect accuracy of 100% on the test data, but there were indications of overfitting during training, which affected the balance between accuracy and loss. The model performance showed significant variation between classes, with excellent results for Kw 3, but inadequate for Kw 1 and Kw 2, especially in terms of recall.

To improve the overall performance of the model, it is recommended to conduct additional experiments with parameter and epoch variations and apply regularization techniques such as dropout to reduce overfitting. In addition, improving the quality and quantity of training data as well as exploring data augmentation techniques can help the model to better recognize patterns and improve detection capabilities across all classes.

Reference

[1] A. Khairunisa and Aviasti, "Proposed Improvement of Tile Making Process by Using Taguchi Method at Mahkota Home Industry," *Bandung Conf. Ser. Ind. Eng. Sci.*, vol. 3, no. 1, Jan. 2023, doi: 10.29313/bcsies.v3i1.6614.

[2] D. Prasetyo and A. Nugroho, "Decision Support System for Tile Selection Using Analytical Hierarchy Process Method ISSN 2338-5677 Print ISSN 2548-6646 Online," vol. 11, no. 1, pp. 24-30, 2023.

[3] M. Irfa, F. I. Adhim, and F. Istiqomah, "Implementation of Pid Method to Control the Position of Servo Motor in Dough Weight Sorting System," vol. 10, no. 2, 2021.

[4] F. N. Cahya, N. Hardi, D. Riana, and S. Hadianti, "Classification of Eye Diseases Using Convolutional Neural Network (CNN)," vol. 10, pp. 618-626, 2021.

[5] Y. Pratama, E. Rasywir, D. Kisbianty, and B. Irawan, "Layer Pooling Experiment using Standard Deviation for Face Image Dataset Classification with CNN Method," vol. 5, no. 1, pp. 200-210, 2023, doi: 10.47065/bits.v5i1.3604.

[6] F. F. Maulana and N. Rochmawati, "Fruit Image Classification Using Convolutional Neural Network," vol. 01, pp. 104-108, 2019.

[7] A. Salsabila, R. Yunita, and C. Rozikin, "Identification of Flower Type Image using KNN Algorithm with HSV Color Extraction and GLCM Texture," *Technomedia J.*, vol. 6, no. 1, pp. 124-137, 2021, doi: 10.33050/tmj.v6i1.1667.

[8] A. B. Prakosa, F. T. Information, U. Kristen, and S. Wacana, "IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK (CNN) DEEP LEARNING MODELS ON CORN LEAF DISEASE IMAGES," no. April, pp. 107-116, 2023.

[9] A. Peryanto, A. Yudhana, and R. Umar, "Image Classification Using Convolutional Neural Network and K Fold Cross Validation," vol. 4, no. 1, pp. 45-51, 2020.

[10] M. Resa, A. Yudianto, and H. Al Fatta, "WAYANG WITH CONVOLUTIONAL NEURAL NETWORK ALGORITMA," vol. 2, no. 2, pp. 182-190, 2020.

[11] I. S. Ardiansyah, "Implementation of Convolutional Neural Network for Rice Type Classification Based on Digital Image," vol. 12, no. 1, pp. 4172-4182, 2023.

[12] C. Rahmad, E. Rohadi, and E. A. Widyatama, "APPLICATION FOR DETERMINING THE QUALITY LEVEL OF CHICKEN EGGS BASED ON THE COLOR AND TEXTURE OF SHADED IMAGES BY HUE, SATURATION, VALUE METHOD," pp. 9-14, 2020.

[13] Z. F. Abror, "CLASSIFICATION OF FIRE AND NON FIRE IMAGES USING CONVOLUTIONAL NEURAL NETWORK," vol. 24, no. 100, pp. 102-113, 2019.

[14] D. H. Firdaus, ImranBahtiar, L. D. Bakti, and E. Suryadi, "CLASSIFICATION OF CATARACT IN THE EYES USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD," vol. 1, no. 3, 2022.

[15] M. F. Naufal *et al.*, "Classification of Rock Paper Scissors Game Images Using Convolutional Neural Network," vol. 20, no. 1, pp. 166-174, 2021.

[16] A. S. Riyadi, I. Puspa, and S. Widayati, "CLASSIFICATION OF DOG AND CAT IMAGES USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD," vol. 5, pp. 2-6, 2021.