CHAPTER 4

ANALYSIS AND DESIGN

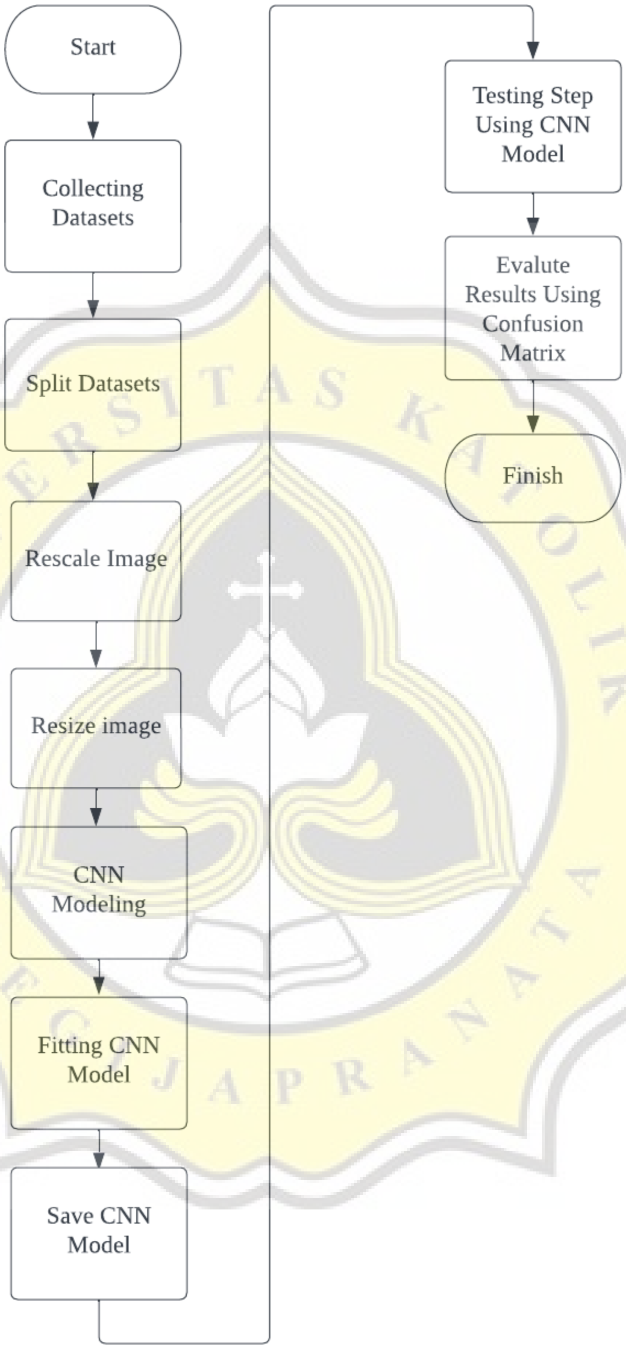

Figure 4.1 Flowchart of CNN Method

The classification process begins at the collecting dataset section. The dataset sourced from Kaggle named Meat Freshness Image Dataset (Shanawad, 2022). The dataset contains with meat image data with a 2,266 images data divided into three classes as fresh meat, half-fresh meat, and spoiled meat. There are 853 images data of fresh meat, 789 images data of half-fresh meat, and 624 images data of spoiled meat.

After the dataset collection process was done, the dataset was split in five different data variants. In the first variant, the data used in the training and validation data is 50% and 10%, while for the testing data it is 40%. In the second variant, the data used in training and validation data is 50% and 20%, while for testing data it is 30%. In the third variant, the data used in training and validation data is 60% and 20%, while for testing data it is 20%. In the fourth variant, the data used in training and validation data is 70% and 10%, while for testing data it is 20%. Finally, In the fifth variant, the data used in training and validation data is 80% and 10%, while for testing data it is 10%.

After the data split process was done, then the image processed by rescaling process. Pradnya and Kusumaningtyas[16] stated that image rescaling is done by reducing the values of the image matrix by dividing the matrix values in the range 0-255 by 255 so that obtained matrix values in the range 0-1 and resizing the image data into 128×128 pixels as shown below:

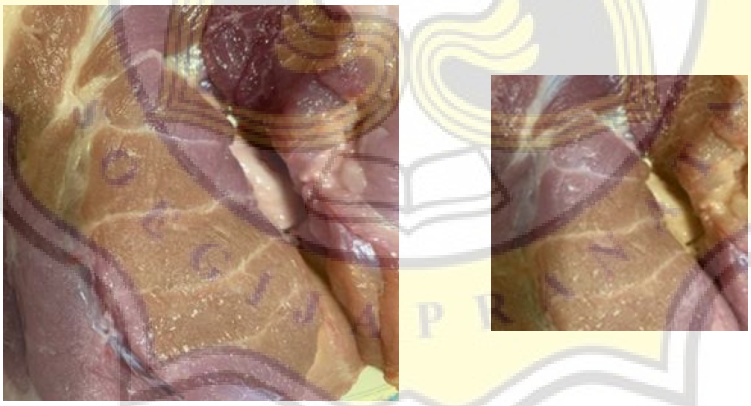

Figure 4.2 Original Image (Left), Resized Image (Right)

After the image rescaling and resizing process was done, then the image processed using the Convolutional Neural Network model below:

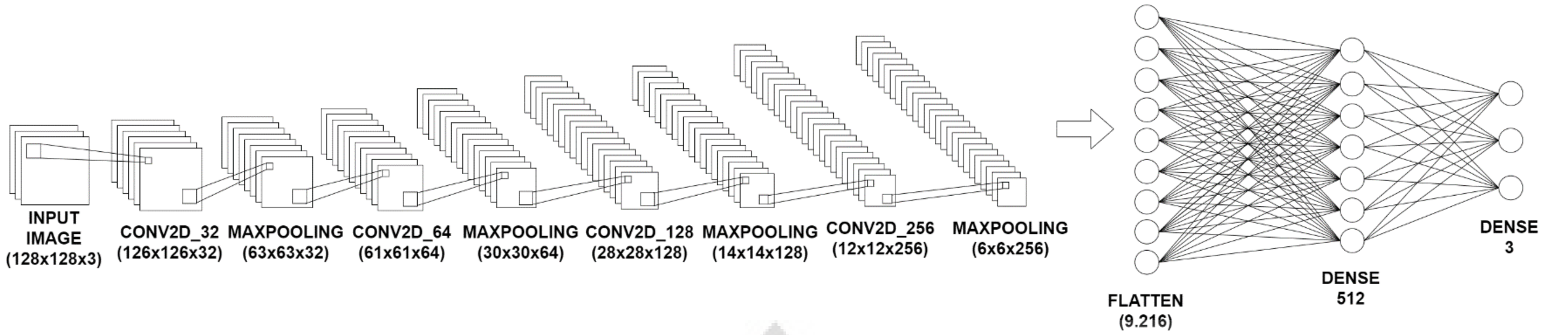

Figure 4.3 CNN Architecture Scheme

If presented in the form of a table, the architecture of the Convolutional Neural Network can be explained as follows:

Layer	Filter	Kernel Size	Stride	Dense Unit	Activation
Conv2D	32	3x3	1	-	ReLu
Maxpooling2D	-	2x2	2	-	-
Conv2D	64	3x3	1	-	ReLu
Maxpooling2D	-	2x2	2	-	-
Conv2D	128	3x3	1	-	ReLu
Maxpooling2D	-	2x2	2	-	-
Conv2D	256	3x3	1	-	ReLu
Maxpooling2D	-	2x2	2	-	-
Flatten	-	-	-	-	-
Dense	-	-	-	512	ReLu
Dropout (p=0.5)	-	-	-	-	-
Dense	-	-	-	3	Softmax

Table 4.1. Convolutional Neural Network Model

The Convolutional Neural Network model used in this project consists of four convolution layers with a 32, 64, 128, and 256 filters, 3×3 kernel size with one stride filter for each convolution process that carried out on the input image.

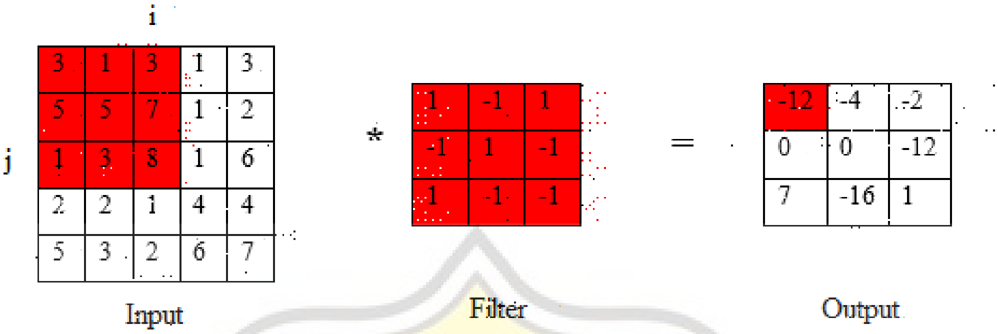

Figure 4.4 Convolution Process[17]

Sari[18] has shown that proses convolution (Figure 4.3) it's done by multiplying the matrix values contained in the input image by the matrix values of the kernel to generate new matrix values that form the features of the input image. The input of meat image with 128x128x3 dimension size, convoluted at the first convolution layer so that resulting in dimensions 126x126x32.

The values of the features of the image, then activated using the ReLu activation function. Ilahiyah and Nilogiri[19] stated that ReLu activation function is used to convert matrix values less than 0 into 0. After the activation process done, the features are processed on the max pooling layer. Sari[18] has shown that the max pooling process is performed by taking the largest value from the image feature matrix using a 2×2 kernel to reduce the output size of the image feature so that can reduce the calculations that processed in the network.

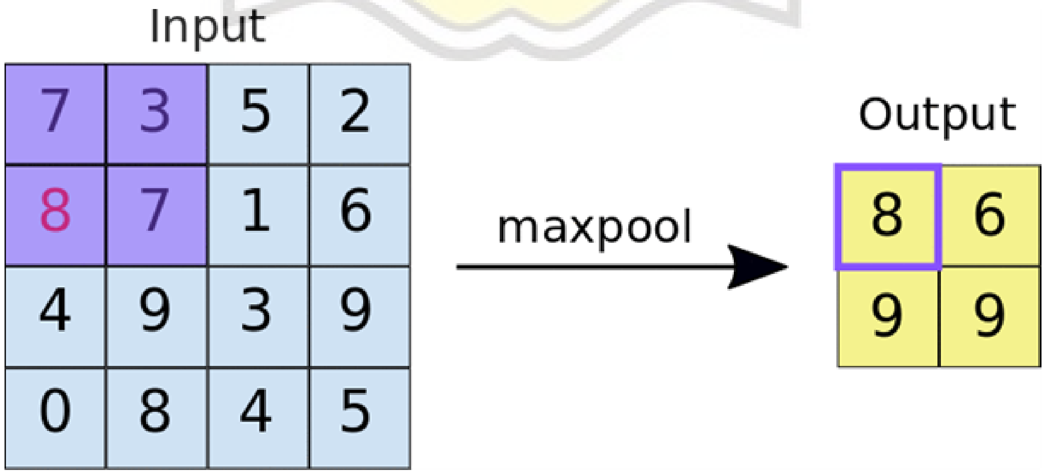

Figure 4.5 Maxpooling Process[4]

The maxpooling process on the image feature resulting from the previous convolution process resulting in dimensions 63x63x32.

The image feature that has been generated in the maxpooling process then becomes the input for the second convolution process on the second convolution layer. The image feature with dimensions of 63x63x32, is convoluted on the second convolution layer resulting in 61x61x64 dimensions. The resulting image feature is then activated using the ReLu activation function. The image feature that has been activated, then processed on the maxpooling layer so that it produces dimensions of 30x30x64.

The image feature that has been generated in the maxpooling process, then becomes the input for the third convolution process on the third convolution layer. The image feature with dimensions of 30x30x64, is convoluted on the third convolution layer resulting in dimensions of 28x28x128. The resulting image feature is then activated using the ReLu activation function. The image feature that has been activated, then processed on the maxpooling layer so that it produces dimensions of 14x14x128.

The image feature that has been generated in the maxpooling process, then becomes the input for the fourth convolution process on the fourth convolution layer. The image feature with dimensions of 14x14x128, is convoluted on the fourth convolution layer resulting in dimensions of 12x12x256. The resulting image feature is then activated using the ReLu activation function. The image feature that has been activated, then processed on the maxpooling layer so that it produces dimensions of 6x6x256.

After the convolution and pooling process is carried out, the obtained image features from the entire process are processed on the flatten layer.

Pradnya and Kusumaningtyas[16] stated that flatten is the process of reshaping the matrix of image features that take the form of a multidimensional array into one dimension as an input neuron in a fully connected layer. The image feature which generated from the previous maxpooling process is processed into one dimension resulting in 9,216 as a dimension value. The produced input neurons through the flatten process, are then interconnected with dense of 512 units and three units.

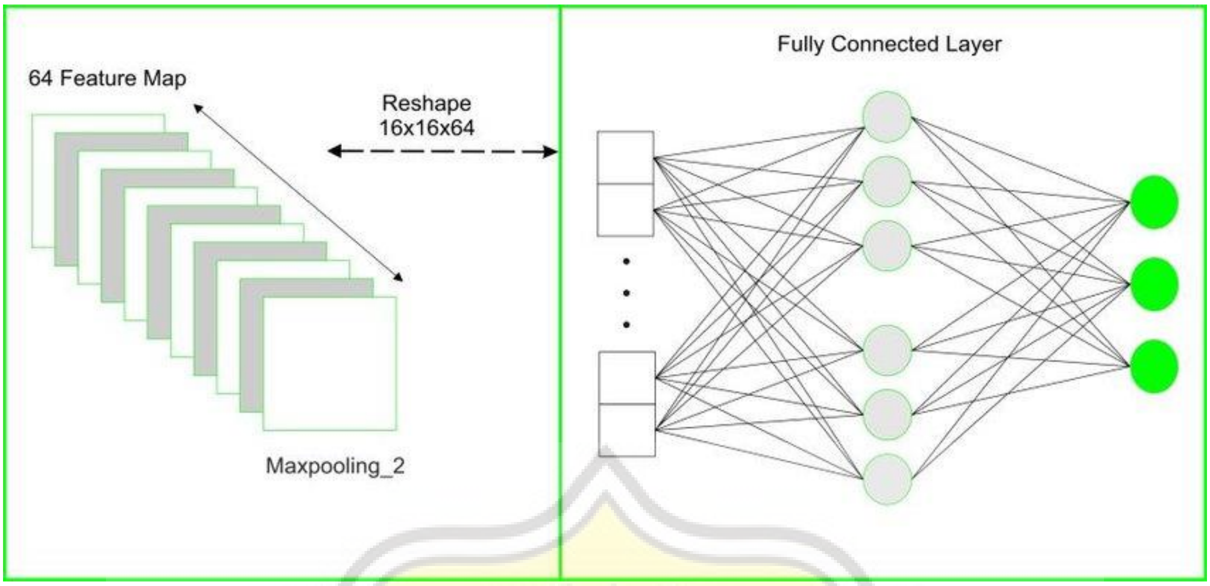

Figure 4.6 Flatten Layer and Fully Connected Layer[20]

Hidden layer (Figure 4.5) with 512 dense units was activated by the ReLu activation function to activate the results of calculations which calculate on the network that interconnected with each neuron on the fully connected layer. The result that has been activated, then regularized using a dropout (p = 0.5) to prevent overfitting. Wulandari et al.[21] stated that overfitting is caused by the presence of unneeded neuron in a fully connected layer. An output layer that implements three neuron units, activated using the Softmax activation function. Ilahiyah and Nilogiri[19] stated that the use of Softmax activation can obtain a good probabilistic interpretation of the multiclass classification process.

The designed Convolutional Neural Network model on Convolutional Neural Network modeling section, then compiled using the Adam optimizer (learning rate 1e-4). Wikarta et al.[22] stated that Adam's optimizer can produce Convolutional Neural Network models with high accuracy so it can to classify images properly. After the Convolutional Neural Network model is compiled, then the model is processed at the fitting model section using 50 epochs and 64 batch sizes. The fitting model process resulted a Convolutional Neural Network model that stored for use in the testing section.

The testing section is done using five variants testing data in accordance with the training data and validation data which specified in the data split process. The testing data is determined using a saved CNN model. The results of the tests, then evaluated using the following confusion matrix.

Confusion	Actual Values	Actual Values

Matrix	(Positive)	(Negative)
Predicted Values (Positive)	True Positive	False Positive
Predicted Values (Negative)	False Negative	True Negative

Table 4.2. Confusion Matrix

Confusion matrix (Table 4.1.) consists of True Positive, True Negative, False Positive, and False Negative. Lasniari et al.[23] stated that true positive is positive data that is identified as true, true negative is negative data that is identified as true, false positive is positive data that is identified as false, and false negative is a negative data that is identified as false. After determining the values of true positive, true negative, false positive, and false negative, then those values are inputted in the precision (1), recall (2), F1-score (3), and accuracy (4) function used to calculate the precision, recall, F1-score, and accuracy of the Convolutional Neural Network model based in determining the freshness of meat as follows:

$$Precision: \frac{TP}{(FP + TP)} \tag{1}$$

$$Recall: \frac{TP}{(FN + TP)} \tag{2}$$

$$F1 - score: \frac{(2\ x\ Recall\ x\ Precision)}{(Recall + Precision)} \tag{3}$$

$$Accuracy: \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \times 100\% \tag{4}$$