CHAPTER 4
ANALYSIS AND DESIGN

4.1. Data Acquisition and Preprocessing

The dataset used in this research is acquired from the website Kaggle.com by the name of 'Fruit fresh and rotten for classification'. The data consist of fresh and rotten images of fruits, in this case apple, banana, and orange. The dataset is already split by the creator of the dataset for training and testing. The training and testing dataset contains fresh and rotten fruits images categories for apples, bananas, and oranges, each type of fruits containing more than 1000 images. To do this research the author used 300 images of the fresh and rotten oranges data for training and 20 images of the fresh and rotten oranges data for testing. The training images data is resized to 64x64 resolution using Power Toys application to make the training process faster and less resource consuming. To use the images for testing the author need to acquire the coordinates of the object in the image. To do this the author uses a paint application to get the coordinates of objects in the image. This is an example of the ground truth data that is acquired by using paint application:

Table 1: Ground Truth Data Table

No	***File name***	***Ground Truth Detail***		
		Label	***Accuracy***	***Coordinate***
1	jeruksegar(1).png	Segar	100%	58,99,455,439
2	jeruksegar(2).png	Segar	100%	25,31,354,306

3	jeruksegar(22).png	Segar	100%	89,31,291,121;
				46,122,286,355;
				287,63,501,278

Table 5.1 is an example of the ground truth data that are acquired using the paint application and stored in a ground truth array. The ground truth data can only be used if the input image has the same name as the filename in ground truth data.

4.2. Design

This research is done by turning a CNN into object detection then evaluating it with MAP. CNN is used in this research because it's a commonly used algorithm to classify an image. To turn a CNN into object detection there are 3 steps that need to be done.

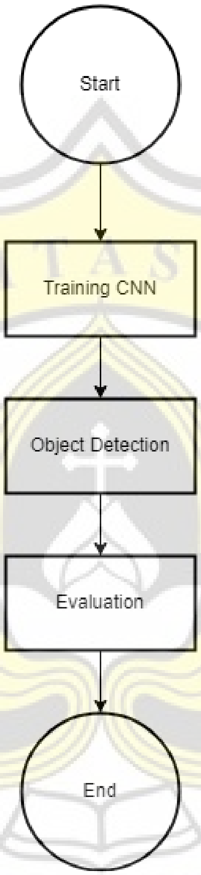

Figure 1:
Research Step

Figure 1 shows the steps how this research is done. The first step is to train the cnn, in this process the cnn is trained with the training dataset, after the training is done the next step is to turn the cnn into an object detection model, then the final step is to evaluate the object detection model. Here is the detailed explanation in each steps of this research :

4.2.1. Training CNN

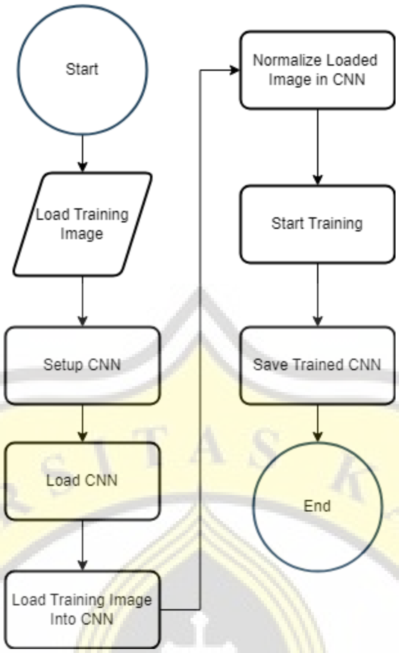

Figure 2: CNN Training Flowchart

Figure 2 shows how CNN is trained. The first step is to load the training image and save it into an array. This was done using a function from the p5 library. The next step is to set up the CNN. This was done to determine what kind of machine learning that needs to be acquired from the ml5 library and determine the input resolution. After setting up, the cnn is then acquired using the ml5 library and saved into a cnn variable, then the training image get loaded into the cnn. Before training the loaded image needs to be normalized so it can be read by the cnn. After training the CNN is saved so it can be used later.

4.2.2. Object Detection

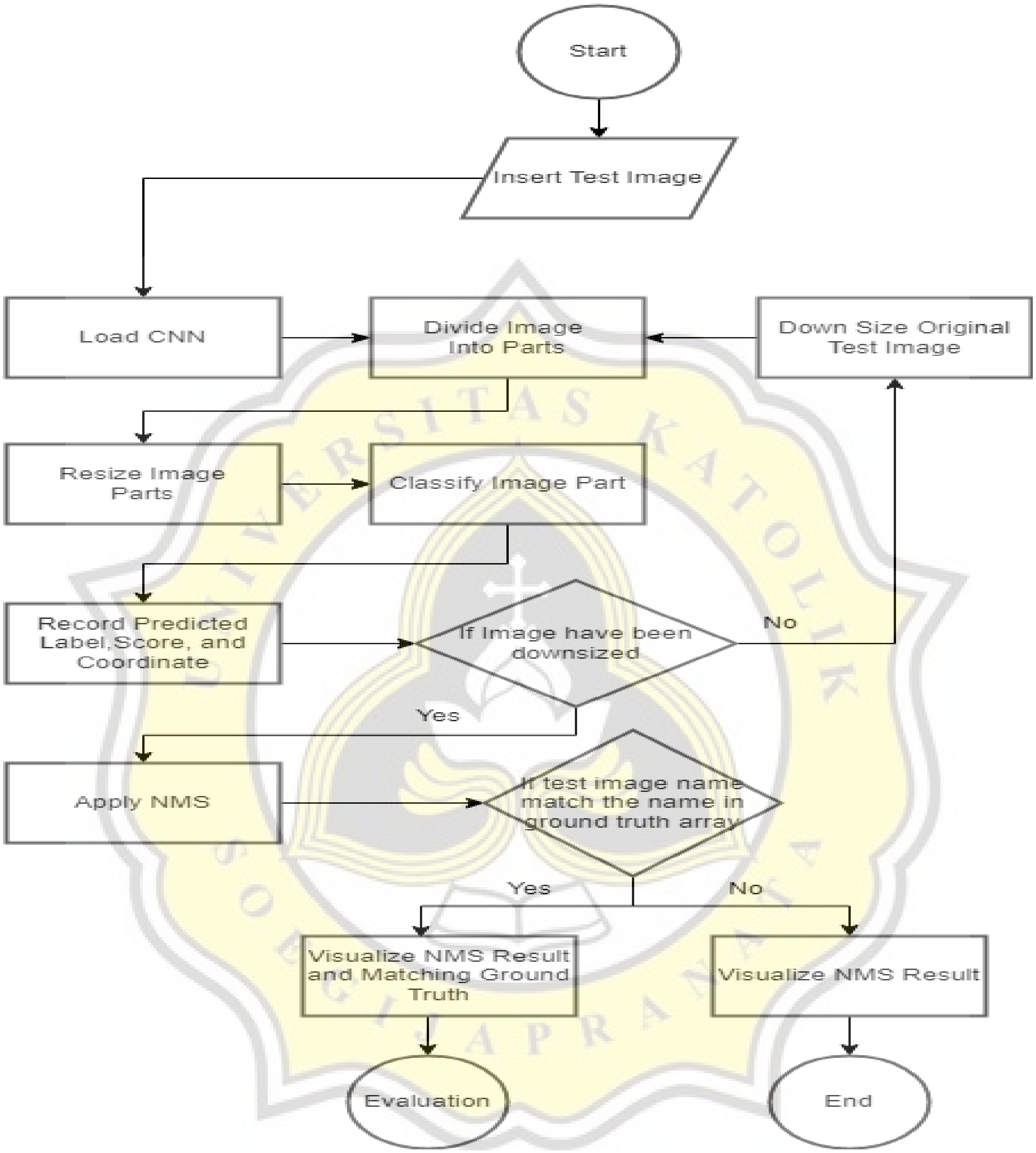
Start
Insert Test Image
Load CNN
Divide Image Into Parts
Down Size Original Test Image
Resize Image Parts
Classify Image Part
Record Predicted Label,Score, and Coordinate
If Image have been downsized
No
Yes
Apply NMS
If test image name match the name in ground truth array
Yes
No
Visualize NMS Result and Matching Ground Truth
Visualize NMS Result
Evaluation
End

Figure 3: Object Detection Flowchart

Figure 3 shows how the CNN gets turned into object detection. First the test image is inserted, after the test image is inserted then the trained CNNs that have been saved from the previous step in chapter 5.2.1 are loaded. Then the test image is divided into small even parts in this process the x, y, w, and h coordinates data of the divided parts in the test image is acquired . Because cnn is trained using 64x64 training images, the divided image parts need to be resized first so it can be classified by the cnn. After classifying the image parts, the data from the classifying process and the x, y, w, and h coordinates data are recorded. Then it checks if the test image has been downsized, if it is not then the test image is downsized first then the downsized test image went through the same process after loading the cnn until the data of the downsized test image have been recorded. Then it applies the NMS (Non Max Surpression) function to determine which is the object in the image based on the recorded data, NMS works by finding intersections between recorded data.

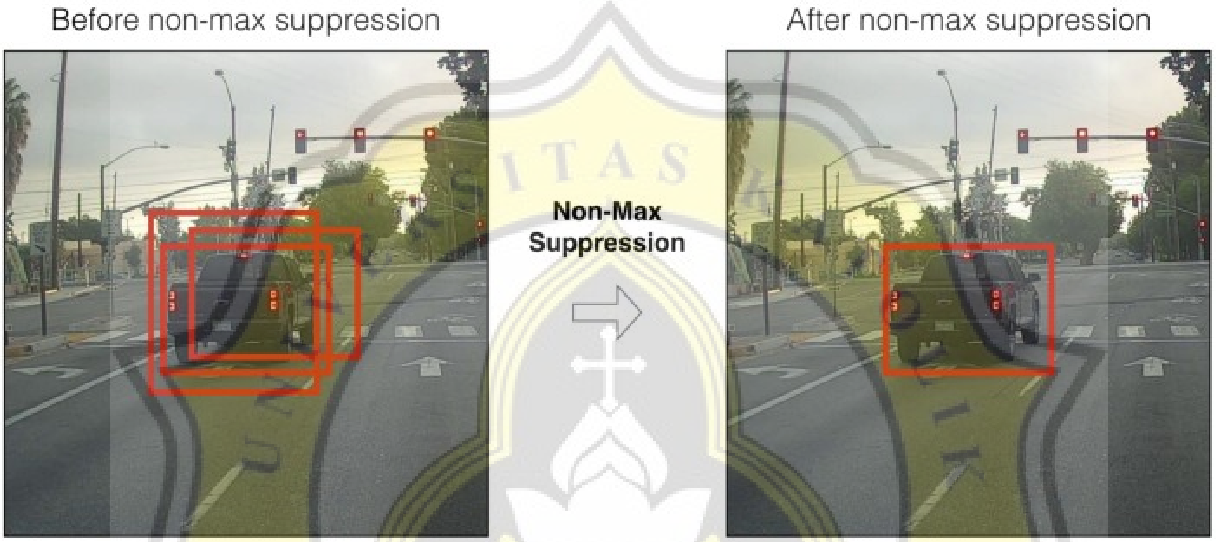

Figure 4: NMS Example

NMS determines which part of an image is an object by looking at how many times that part of images gets predicted by CNN. This is the purpose of downsizing the test image and processing it after processing the test image, so the NMS can determine which part of the image is the object by intersecting the data between the test image and the downsized test image.

After applying the NMS function, the next step is checking if the test image name matches the file name in the ground truth array. If it matches then the result from the NMS function and the matching ground truth data is visualized then the evaluation process will be started. If it didn't match only the NMS function results will be visualized and the evaluation process won't be started.

4.2.3. Evaluation

This step only starts if the inserted test image file name in chapter 5.2.2 matches the filename in the ground truth array.

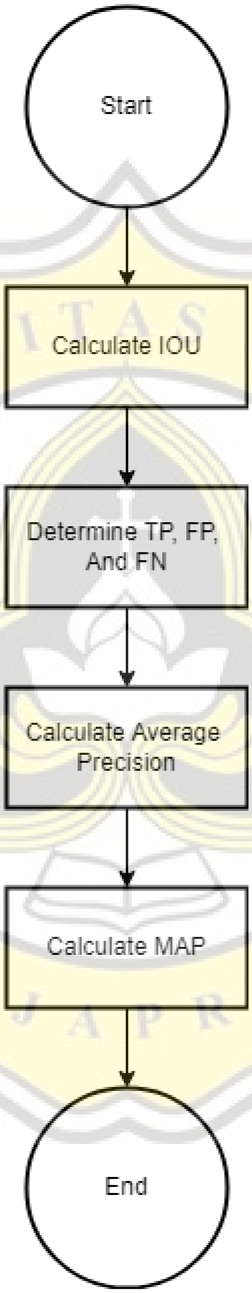

Figure 5:
Evaluation Process
Flowchart

The evaluation process started by calculating the IoU(Intersection over Union) between the ground truth data and the test image data.

$$IoU = \frac{Ground\,Truth \cap Prediction}{Ground\,Truth \cup Prediction}$$

Figure 6: IoU Formula

To calculate the IoU, the intersection and union between ground truth data and prediction data needs to be calculated then the results are divided. To determine TP(True Positive), FP(False Positive), and FN(False Negative) all the results of the IoU process need to be compared with the threshold variable. TP means the prediction is correct because it matches ground truth data. FP Means the prediction did happen but it does not fully match with the ground truth data. FN means the prediction did happen but the result didn’t match with the ground truth data. TN(True Negative) isn’t used in this process because TN means the prediction didn’t happen and there is no ground truth data. The IoU results will be considered TP if the results value is higher than threshold value, it will be considered FP if the IoU results is higher than 0 but lower than threshold, and it will be considered FN if the IoU results is 0.

After getting the TP,FP and FN data then the next step is to calculate Average Precision for each class of the prediction data. To do this the Precision and Recall needs to be calculated first.

$$Precision = \frac{CumTP}{Number\ of\ Detection}$$

Figure7: Precision Formula

$$Recall = \frac{CumTP}{Number\ of\ Ground\,Truth}$$

Figure8: Recall Formula

Figure 7 and Figure 8 show the formula to calculate precision and recall. CumTP (Cumulated True Positive) is the total of how many TP have been determined. Number of Detection is the total of how many objects got detected by the object detection, Number of Ground Truth is the total ground truth object in the test image. Precision and Recall are calculated at the same loop as TP, FP, and FN.

After calculating Precision and Recall both results are sorted from lowest to highest in order of the recall. After sorting the duplicate recall value needs to be removed and take the recall value that have the highest precision value between the duplicated recall value to get the Pinterp (Interpolated Precision).

$$Pinterp\left(r_i + 1\right) = max\ p\left(\acute{r}\right)$$

Figure 9: Pinterp formula

$$AP@Threshold=\sum_{i}\left(r_i+1-r_i\right)Pinterp\left(r_i+1\right)$$

Figure10: Average Precision Formula

After calculating the Pinterp the next step is to calculate the Average Precision using the formula in Figure 10. Average Precision counted separately for each class in the prediction.

$$MAP@Threshold=\frac{1}{n}\sum_{i=1}^{n}AP_i\,for\;n\,classes$$

Figure11: MAP Formula

After getting the average precision then the results are summed and divided by the number of classes to get the MAP results.