CHAPTER 4
ANALYSIS AND DESIGN

4.1. Analysis

The method that the researcher use is quantitative research to get the most effective object detection architecture for health violations detection (between “Faster R-CNN ResNet50 V1 640x640”, “SSD ResNet50 V1 FPN 640x640 (RetinaNet50)”, and “SSD MobileNet V2 320x320”). The researcher implemented three different data scales for each architecture. The data scale consists of 20%, 50%, 80%. The researcher will compare some indicators like :

1. Mean average precision (mAP)

Based on the cocodataset.org website [21], mean average precision is the indicator of the precision overall images, classes, and IoU thresholds. The formula for single-precision = True Positive/(True Positive+False Positive). The greater the value of mAP, the researcher considers the performance to be better

2. Mean average precision (mAP) for large object

Based on the cocodataset.org website [21], the mean average precision for the large object is the indicator of the precision that only calculates the result of large object detection (> 96x96 size). The formula for single-precision = True Positive/(True Positive+False Positive). The greater the value of mAP, the researcher considers the performance to be better

3. Mean average precision (mAP) for medium object

Based on the cocodataset.org website [21], the mean average precision for the large object is the indicator of the precision that only calculates the result of medium object detection (32x32 until 96x96 size). The formula for single-precision = True Positive/(True Positive+False Positive). The greater the value of mAP, the researcher considers the performance to be better.

4. Mean average precision (mAP) for small object

Based on the cocodataset.org website [21], the mean average precision for the small object is the indicator of the precision that only calculates the result of medium object detection (< 32x32 size). The formula for single-precision = True Positive/(True Positive+False Positive). The greater the value of mAP, the researcher considers the performance to be better.

5. Average recall (AR) @100

Based on the cocodataset.org website [21], the average recall @100 large is the average recall across all images with a maximum of 100 detections across all images. across all classes, and across all IoU thresholds. The formula for single-recall = True Positive/(True Positive+False Negative). The greater the value of AR, the researcher considers the performance to be the better.

6. Average recall (AR) @100 Large

Based on the cocodataset.org website [21], the average recall @100 large is the average recall across all images with the maximum 100 detections across all images with large size (>= 96x96), across all classes, and across all IoU thresholds. The formula for single-recall = True Positive/(True Positive+False Negative). The greater the value of AR, the researcher considers the performance to be the better.

7. Average recall (AR) @100 Medium

Based on the cocodataset.org website [21], average recall @100 medium is the average recall across all images with the maximum 100 detections across all images with a medium size (32x32 until 96x96), across all classes, and across all IoU thresholds. The formula for single-recall = True Positive/(True Positive+False Negative). The greater the value of AR, the researcher considers the performance to be the better.

8. Average recall (AR) @100 Small

Based on the cocodataset.org website [21], average recall @100 small is the average recall across all images with the maximum 100 detections across all images with a small size (< 32x32), across all classes, and across all IoU thresholds. The formula for single-recall =

True Positive/(True Positive+False Negative). The greater the value of AR, the researcher considers the performance to be the better.

4.2. Design

4.2.1. Data Preparation, Model Preparation, Training, Evaluation, and Analysis

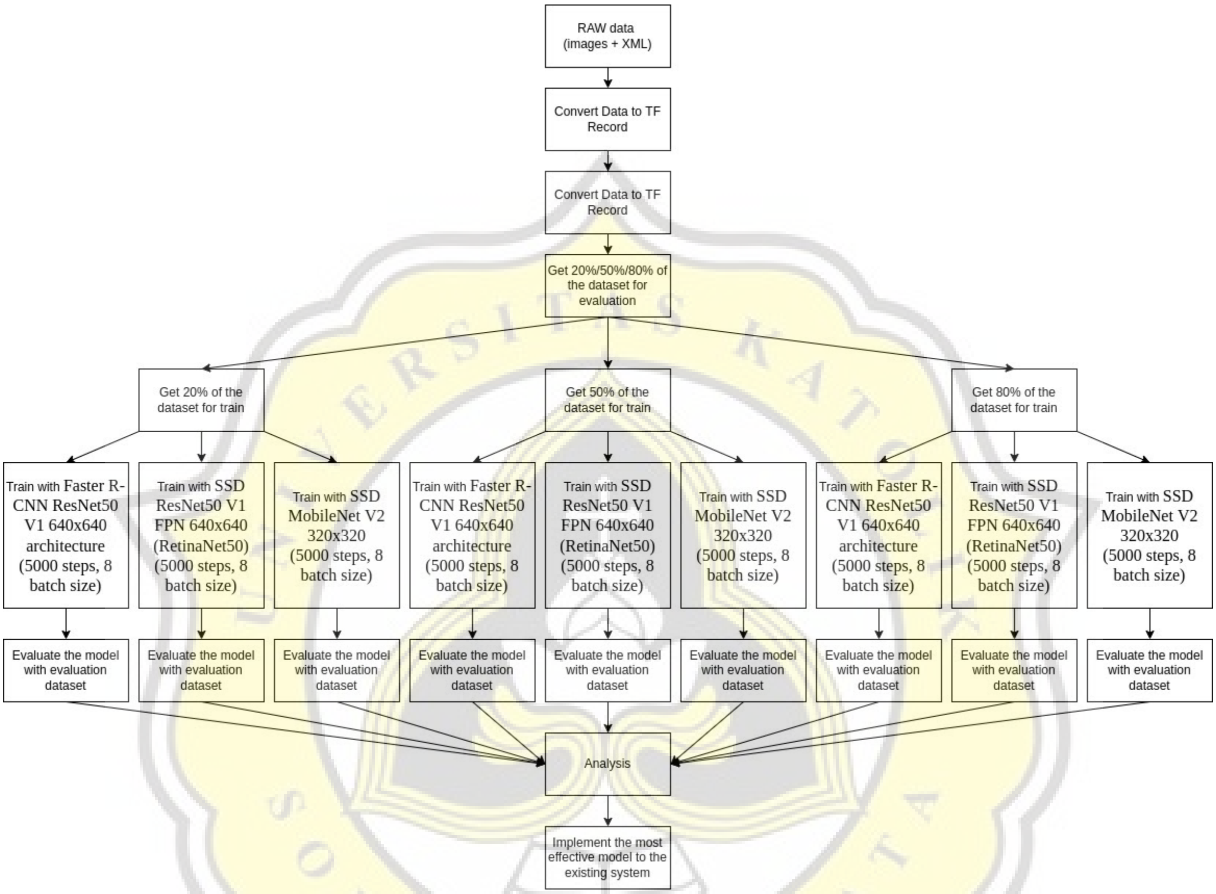

Figure 4.1: Data Preparation, Model Preparation, Training, Evaluation, and Analysis Flow

Figure 4.1 shows the flow of data preparation, model preparation, training, evaluation, and analysis flow. For this section, the researcher will prepare the data and convert the data to TF Record type. After converting to the TF Record type, the researchers took the 20%/50%/80% of the dataset that will be used for the evaluation in the further step.

After the data is ready, the researcher will implement three models for each split data. The model that is implemented is “Faster R-CNN ResNet50 V1 640x640”, “SSD ResNet50

V1 FPN 640x640 (RetinaNet50)", and "SSD MobileNet V2 320x320". The result of these steps is the trained model that is ready to evaluate/use.

After training all of the models from each split data, the model will be evaluated to see the performance using the evaluation dataset that was already prepared in the previous step. The results of these steps are several indicators that will be used for the next steps.

From all of the results of the evaluations, the researcher will compare and analyze the results. From this step, the researcher got knowledge about the most effective models from this research. The results of this step will be useful for the model implementation to the existing system.

4.2.2. Implementation of the Most Effective Model to the Existing System

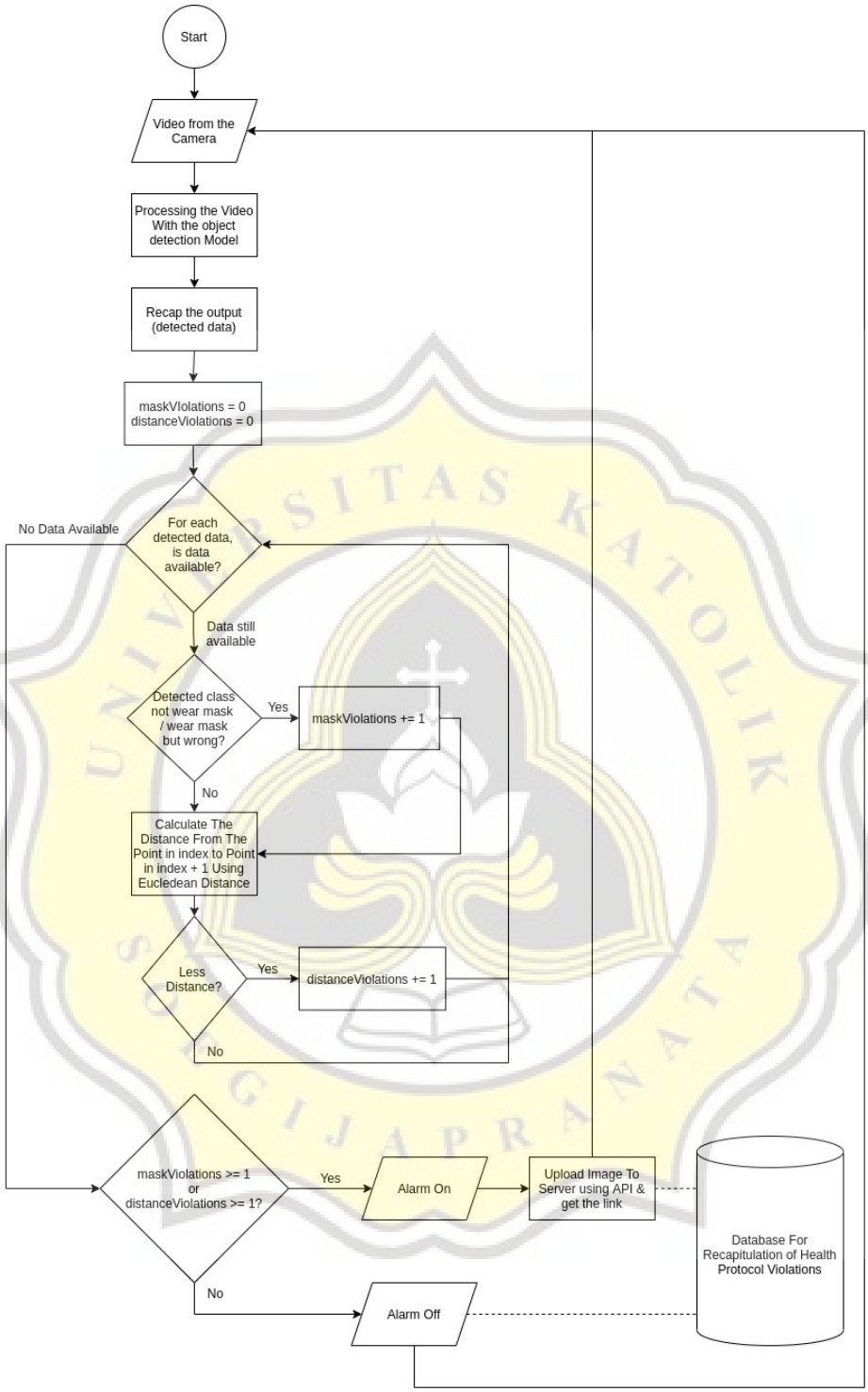

Figure 4.2: Existing System Flow Chart

The existing system is the system that already announced at the Student Creativity Program 2021 held by the Ministry of Education, Culture, Research, and Technology with the team Stephen Royanmart Patrick (Informatics'18) as a leader, Vivian Davina Hendrawan (Informatics'19), Ninda Setyowati (Informatics'19), Melvin Kurniawan Hartono (Electrical Engineering'18), and Samuel Aditya (Electrical Engineering'18). Figure 4.2 shows the flow of the system.

The system will receive the video from the camera. After that, the video will be processed with the object detection models that have already been selected by the previous step of the research. The output of this step is/are the bounding box/es of the detected object. The output also can be None (if the system found no results).

The researcher declares 2 default variables to save the total of mask violations (example: maskViolations) and distance violations (example: distanceViolations) with the value for each variable = 0.

After that, the system will check if any object is found by the system at the previous step. If any, the system will do a loop for each detected data (until no data is available). In the loop, the system will check if the detected class "not wear a mask" / "wear mask but wrong". **If detected**, the system will add 1 value to the maskViolations variable and go to the next step. **Else**, the system directly processes the next steps.

At the next step, the system will check the distance of the object in the current index to the object in the current+1 index. For the calculation, the researcher use euclidean distance formula ($d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$). **If the distance is less than the standard value**, the system will add 1 value to the distanceViolations variable and go to the next step. **Else**, the system directly processes the next steps.

If all objects are already processed/no data can be processed, the data will check the value of the maskViolations and distanceViolations. **If the value of maskViolations>=1 or value of the distanceViolations>=1**, the system will give the command to turn on the alarm. After that, the system will upload the image to the image hosting using API for the proof. The time, the total of maskViolations, the total of the distanceViolations, and the link of the image will be uploaded to the database. **Else**, the system will give the command to turn off the

alarm. After that, the total of maskViolations, the total of the distanceViolation, and “-” as the link of the image will be uploaded to the database. The system will go back to the previous step and do an infinite loop for 24 hours.