# CHAPTER 4
# ANALYSIS AND DESIGN

## 4.1. Analysis

The first processed image is to convert to grayscale then convert to black and white to determine the face, eye, and nose area. After the face and eyes are detected, the program will check whether the nose is detected or not. If the nose is not detected, then the face is wearing a mask, if the nose is detected, then the face is not wearing a mask.

In the face image that is detected, it must be in sufficient light conditions. Image resolution also influences this detection, it must be in good condition. For the detection of more than one face in the frame can be detected and will be notified whether the face is wearing a mask or not.

This condition will calculate the percentage of positive and negative tests. Positive tests such as successfully detecting a mask with a face using a mask and not detecting a mask with a face without a mask. Then there are negative tests such as successfully detecting masks without a face using a mask, not detecting a mask with a face using a mask.

## 4.2. Design

The following is a flow chart for mask detection:

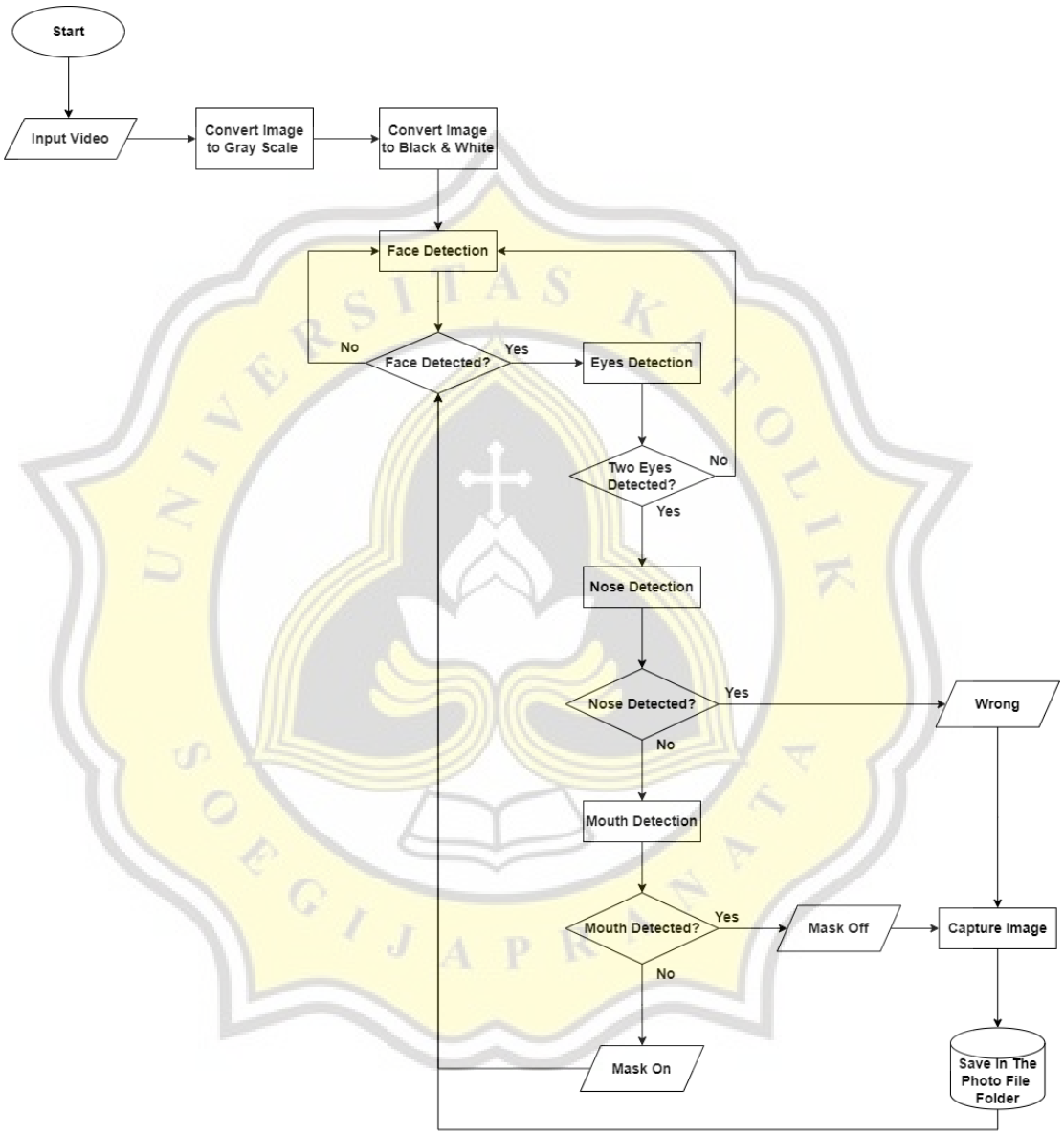


*Figure 1 Mask Detection Flow Chart*

Firstly, a video containing an image of the use of a mask will be played. The video will be converted to grayscale and then converted to black and white. After conversion, the program will detect whether there is a face object in the video, if a face is detected, the next step is to detect an eye object, if an eye is detected it will detect a nose object. If no nose object is detected, then the face is wearing a mask.

If the face is detected with a nose but no mouth is detected, then the person is not wearing a mask properly and will be captured and saved in the photo file folder without a mask. Likewise, faces that are detected by nose and mouth will be captured and stored in a file folder without a mask. After that the mask detector will detect the face again from the beginning. Here's how the program detects masks:


*Figure 2 First View Video*

From the image there is a grayscale process that converts the image to gray. The goal is to remove the red, green, blue component colors and make the color intensity the same. Here's the result convert image to grayscale:

*Figure 3 Convert Image to Grayscale*

The next stage is to change the image to black and white. Black and white to process the pixels in the image because there are only two colors for pixels, namely black and white. In addition, it aims to distinguish between objects and backgrounds.

*Figure 4 Convert Image to Black and White*

$$I_{BW}(x,y) = \frac{I_R(x,y) + I_G(x,y) + I_B(x,y)}{3}$$

$I_R(x,y)$ = Red pixel value point (x,y)

$I_G(x,y)$ = Green pixel value point (x,y)

$I_B(x,y)$ = Blue pixel value point (x,y)

$I_{BW}(x,y)$ = Black and White pixel value point (x,y)

The way of the Haar Cascade Algorithm in detecting an object's face, eyes, nose, and mouth is to train as many positive and negative images as possible and put them into an XML file. This XML file will speed up detecting face, eyes, nose, and mouth objects. The size of the evaluated base box is 24x24 with a scale of 1.25. Here's how Haar Cascade detects objects:

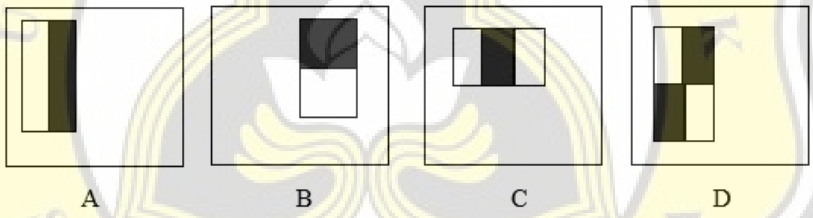


*Figure 5 Haar Feature Face*

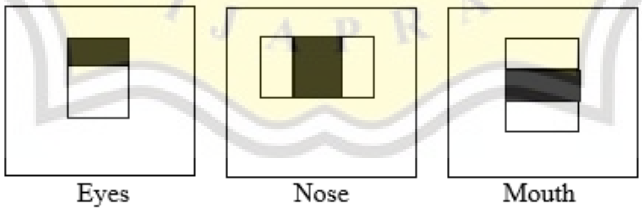


*Figure 6 Haar Feature Eyes, Nose, Mouth*

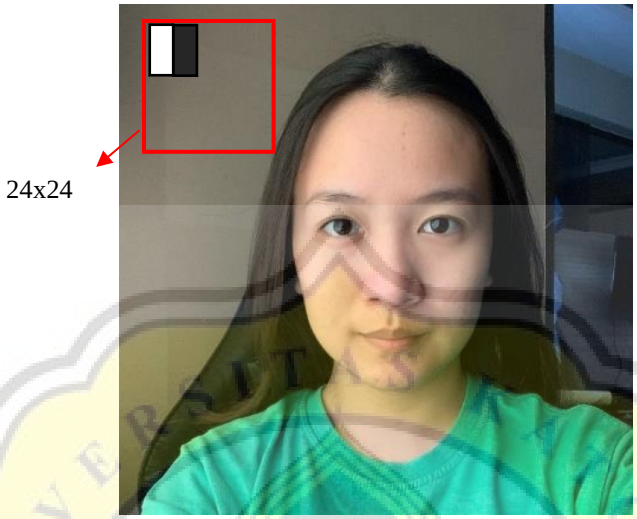


*Figure 7 Haar Feature Work*

The Haar Feature will move the slide all the way down. The Haar will detect whether each box passed is the object being searched for or not. The Haar will move the slide all the way down. The Haar will detect whether each box passed is the object being searched for or not. To get the difference between the pixels in the dark areas and the pixels in the bright areas, you can use the formula:

$$F(Haar) = \sum F\ white - \sum F\ black$$

The next step is the image integral is calculated from the original image so that each pixel is the sum of all the pixels that lie on the left and above in the original image.

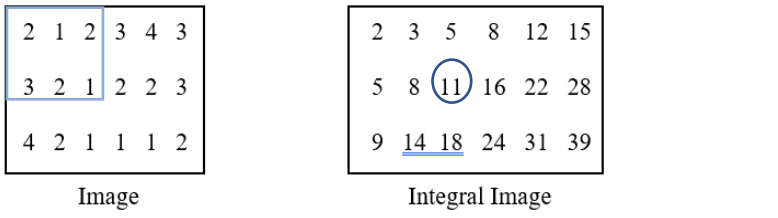


*Figure 8 Integral Image*

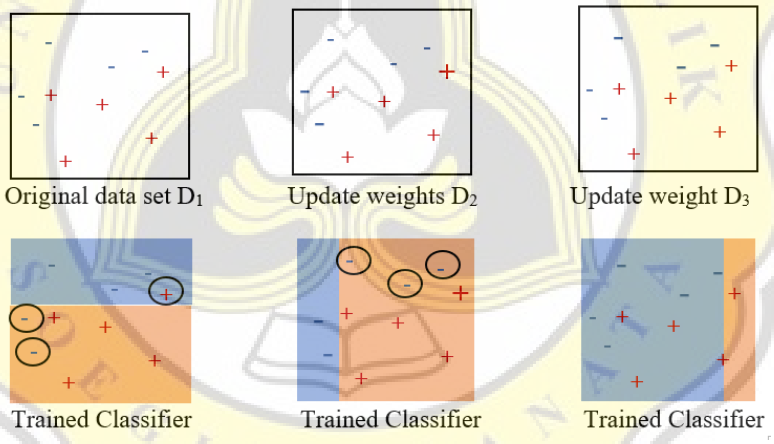


*Figure 9 AdaBoost Classifier*

Classification uses AdaBoost to select the smallest error. Class 1 if it is positive (>=0), class 2 if positive (<0). After Training, weight each classifier and combine them. From this all, it will be determined whether the object is a face or not, whether it is an eye or not, whether it is a nose or not, and whether it is a mouth or not.

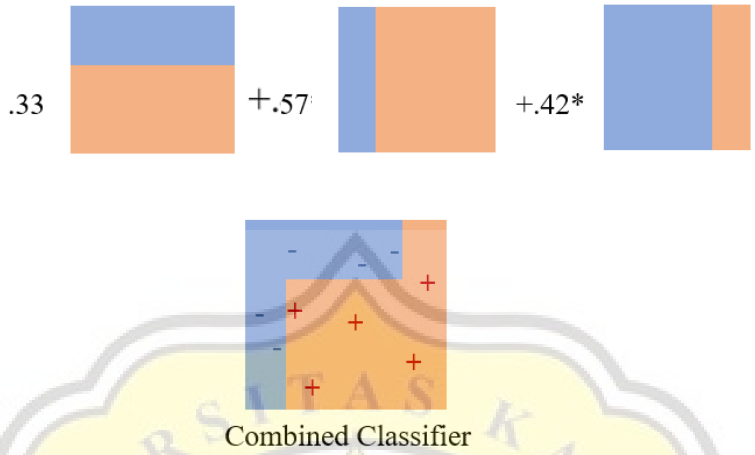


*Figure 10 Combined Classifier AdaBoost*

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon t}{\epsilon t}\right)$$

$\alpha_t$= weight

$\epsilon t$= error rate

After the AdaBoost classification is trained, the region with the smallest error will be obtained. There is a weight to calculate the class. The blue region is assigned a value of -1 and the red region is assigned a value of +1. the results obtained are calculated by weight according to the results in the combined classifier.

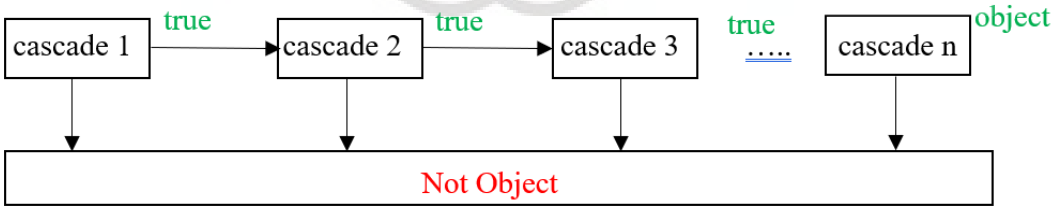


*Figure 11 Cascade Classifier*

After that, the last part is the cascade classifier stage. This stage determines which objects and which are not objects. If it's an object, then cascade will be true. if it is not an object then the cascade will be immediately denied entry to a non-object.

To detect the mask, it will be checked whether the nose and mouth of the face is closed or not. The detected nose and mouth indicate that the face is not wearing a mask. While the nose and mouth that are not detected indicate that the face is wearing a mask. When the nose is detected, and the mouth is not detected it means that the person is not wearing the mask properly.

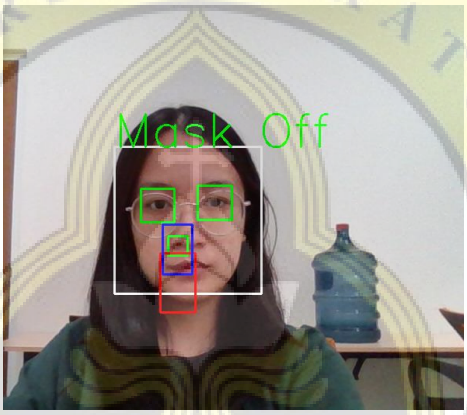


*Figure 12 Face is Not Wearing a Mask*

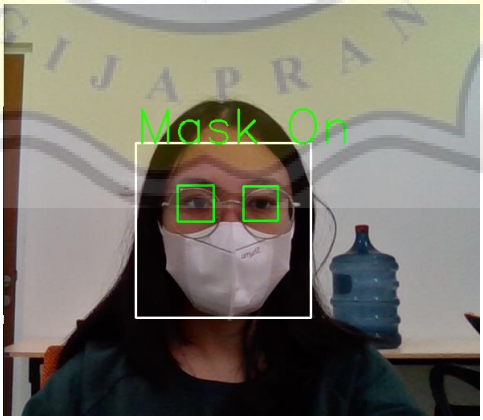


*Figure 13 Face is Wearing a Mask*

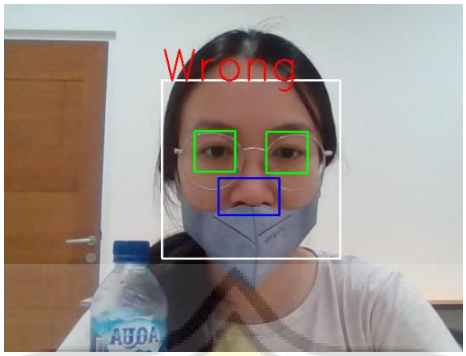


*Figure 14 Not Wearing a Mask Correctly*