CHAPTER 4
ANALYSIS AND DESIGN

4.1. Analysis

The face image will be cropped and isolated from the another object or background with Haarcascade face detection library. So that all input images will produce faces that are already isolated and have the output position of the face that is always the same. The OpenFace models expect images that have (96x96) RGB as input.The OpenFace process will produce an output 128 Dimensional vectors. This vector is used to compare whether the two faces are equal or not.

There are some conditions that must be met. In the face image that is used for taking attendance authentication must be have a good light condition. Low light condition will blurry the shape of the face so it can effect the accuracy of the algorithm. This authentication also affected by image resolution as well, so it really need a good condition of camera.

The positive results of this research are indicated by when an employee succeed in taking attendance using their facial identity, while another person can't using their facial identity to taking attendance for another employee. The negative results of this research are indicated by when the system accidently identified a wrong face , such as can't taking attendance with their own face , either other person can taking attendance for another employee.

This facial authentication system can replace a manual attendance system that still uses a signature. This attendance system has a higher efficiency.

4.2. Design

The code of this project is run in the Jupyter Notebook and Visual Studio Code. Jupyter notebook is used to test the accuracy and calculate the confusion matrix of this research. Meanwhile, Visual Studio Code is used to run Streamlit which is the Face Authentication website's User Interface. The reason of using Jupyter Notebook instead of Google Colab is because the computing speed on the local runtime of my computer's GPU is faster than Google Colab.

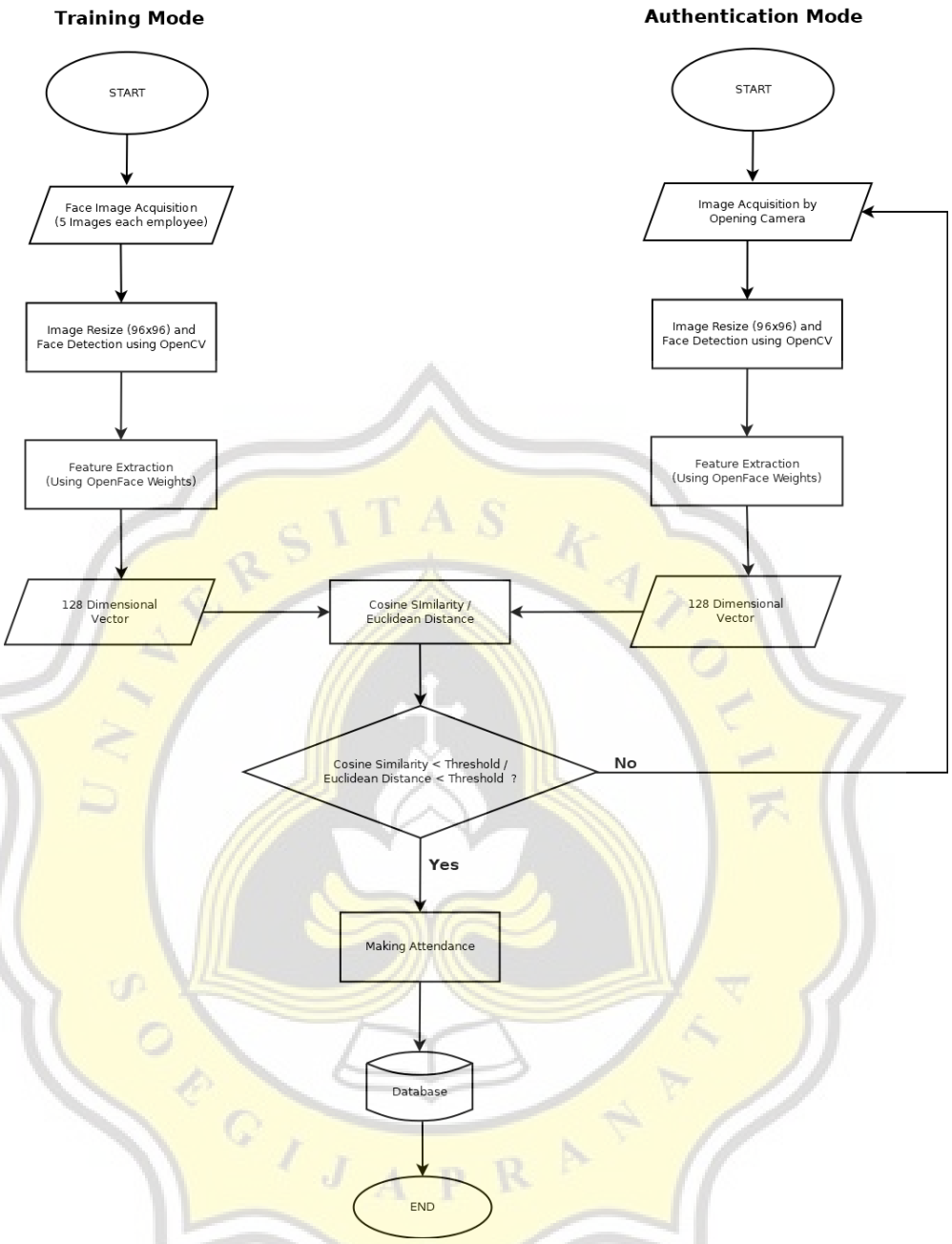

Figure 4. 1 Face Authentication Flowchart

The training dataset is stored in a folder of each employee. Each employee required to upload their face identity images to be used as training data. This image will be collected into one folder with a name based on the username. This all images will be through the face detection process and resized to a size of 96x96 which is OpenFace model expects. The detected faces will

be isolated by cropping the face from the background. So however the position of the face , it will be cropped from the background. Here's how faces isolated :

Figure 4. 2 Face Image Input

From above input image, image will be proceed to crop and isolated from the image background and other object. This process called an face detection , which is in this project using a haarcascade library to doing an detection face job. I use Haarcascade Face Detection because of how fast it can identify faces. Speed is important because I developed an attendance website for this research.

After the face detection process , the color form of face image which is in BGR will be converted to RGB and resize into OpenFace model expect (96x96).

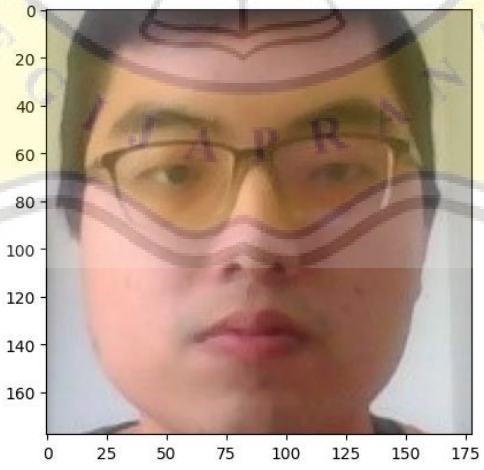

Figure 4. 3 Face Detection and Cropped Representation

After identifying faces in an image, the system will crop the faces and turn them into a feature extraction algorithm, which create feature map to produces a face embedding that represents the features of the face. In this research, I used an Openface Algorithm to create face embeddings which is produce a 128D vectors. Here's how Openface create a face embeddings :

Table 4. 1 Openface Models [11]

Type	Output Size	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj
conv1 (7 × 7 × 3,2)	48 × 48 × 64						
max pool + norm	24 × 24 × 64						m 3 × 3, 2
inception (2)	24 × 24 × 192		64	192			
norm + max pool	12 × 12 × 192						m 3 × 3, 2
inception (3a)	12 × 12 × 256	64	96	128	16	32	m, 32p
inception (3b)	12 × 12 × 320	64	96	128	32	64	l_2, 64p
inception (3c)	6 × 6 × 640		128	256,2	32	64,2	m 3 × 3, 2
inception (4a)	6 × 6 × 640	256	96	192	32	64	l_2, 128p
inception (4e)	3 × 3 × 1024		160	256,2	64	128,2	m 3 × 3, 2
inception (5a)	3 × 3 × 736	256	96	384			l_2, 96p
inception (5b)	3 × 3 × 736	256	96	384			m, 96p
avg pool	736						
linear	128						

Table 4.1 above shows the convolution in the case of producing a face embedding from a face image. when the facial image passes through the convolution layer, the face images will be processed using the inception (3a,3b,3c,4a,4e,5a,5b) to produce a 128-dimensional face embedding.