

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1. Implementation

To get started, the user needs to take pictures with the camera input in the website. After that, user can determine the matching method between cosine similarity or euclidean distance. When a user take a picture of his/her face , the image will be processed by calling a function preprocess_image() and Convolution models called builtModel(). The output of this process is a 128D vector that represented user face. This vector will compared to training dataset vector by calling a function named countcosine(). The output of this process is score of the cosine similarity between user face and training dataset face. If the result is smaller than 0.014 (Cosine Similarity) or 0.17 (Euclidean Distance) then the system will called an attendance() function to insert the checkin time into a database. If the result is bigger than 0.014 (Cosine Similarity) or 0.17 (Euclidean Distance) then the system will alert the user that Face is not verified.

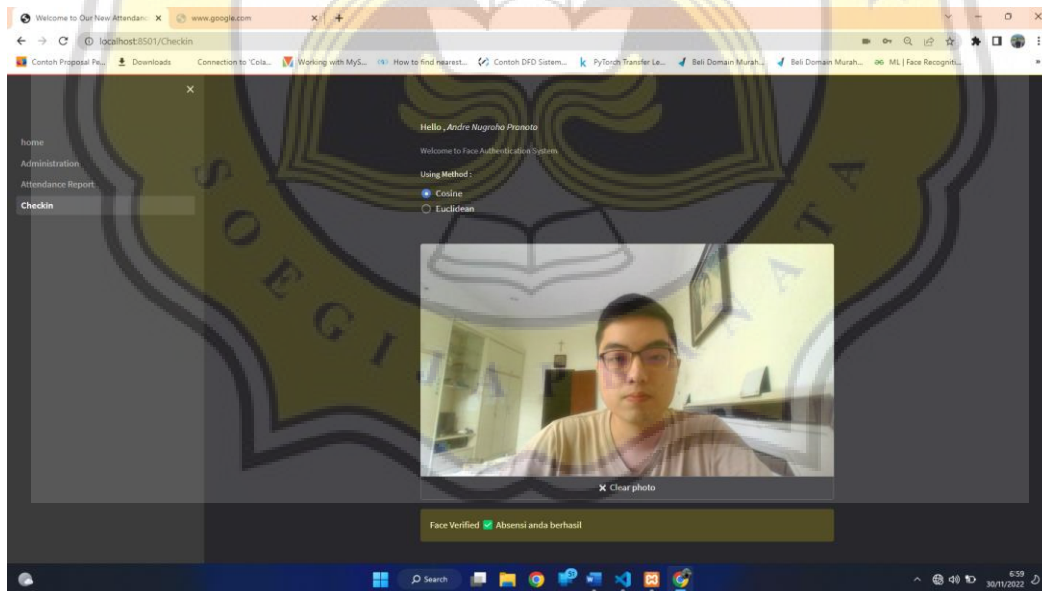


Figure 5.1 1 User Successfully taking Attendance

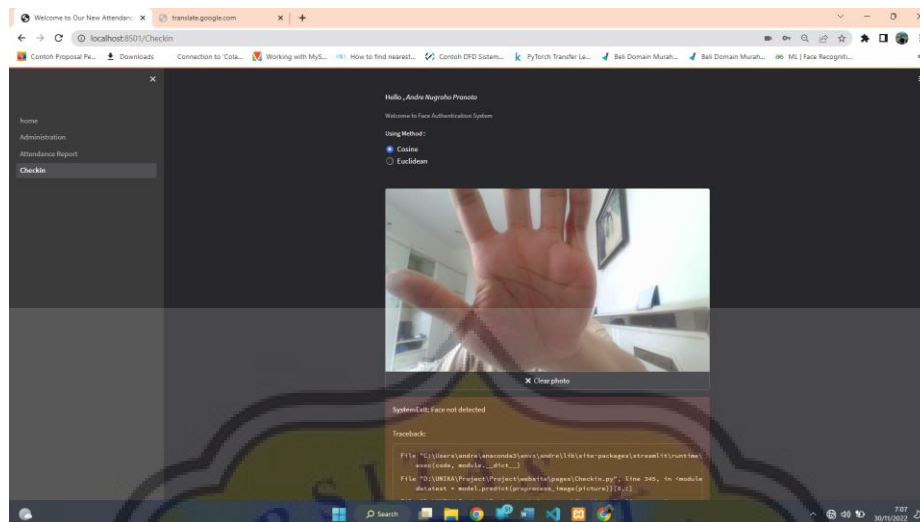


Figure 5.1 2 User Face Not Detected

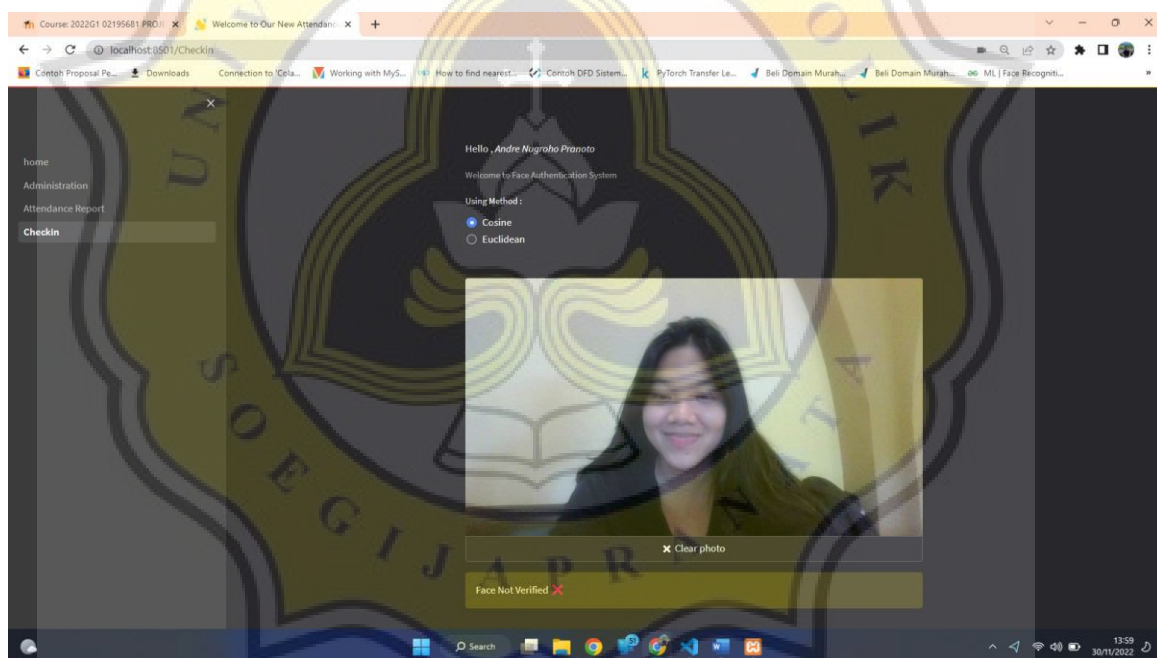


Figure 5.1 3 User Face Not Verified

Compared to manual attendance, figure 5.1.1 to 5.1.3 above can show how the system can prevent fraud such as Indonesian said as “Titip Absen”. The user can only make attendance using his own face. When the user makes attendance by “Titip Absen” using other person, the system will not recognize that person and not make attendance for that user. The system can do this job under 5 seconds. Thanks to Openface Algorithm that is lightweight, so it can work pretty fast in terms of recognizing face.

Besides being efficient for attendance, this website is also efficient for displaying attendance reports. This website can present attendance data in the form of a pie chart.

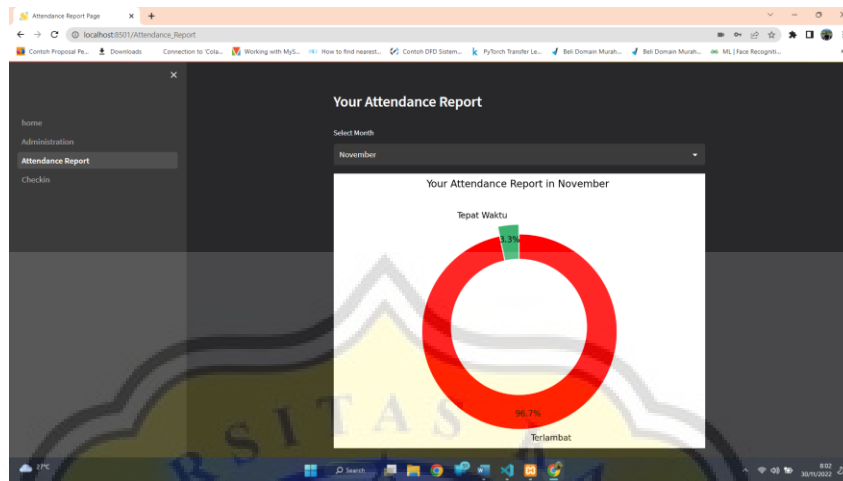


Figure 5.1 4 User Attendance Report

5.2. Results

Testing of this project is done by taking attendance using different faces and the same faces in the dataset. The results of the test will be divided into 4 category below.

Table 5.2 1 Positive & Negative Prediction

	Positive Prediction	Negative Prediction
True	True Positive (TP)	True Negative (TN)
False	False Positive (FP)	False Negative (FN)

Category 1 is True Positive result. This result will be obtained when the program successfully recognizing the same face of the dataset. Category 2 is True Negative result. This result will be obtained when the program unrecognized face that different from the dataset. Category 3 is False Positive. This result will be obtained when the program can recognized the face that different from the dataset. Category 4 is False Negative. This result will be obtained when the program unrecognized the face that same person in the dataset. The results in this project can

be seen using the confusion matrix and table below. There are 5 faces of “Will Smith” that used as the training dataset. The testing data are contains 20, 50, and 100 faces. All are split into portions of 70% incorrect faces and 30% correct faces and different from the training dataset.

Table 5.2 2 TP,TN,FP, FN

	n=20				n=50				n=100			
	Cosine		Euclidean		Cosine		Euclidean		Cosine		Euclidean	
	Positive Predict	Negative Predict	Positive Predict	Negative Predict	Positive Predict	Negative Predict	Positive Predict	Negative Predict	Positive Predict	Negative Predict	Positive Predict	Negative Predict
True	6	12	5	13	14	26	13	27	24	51	24	53
False	2	0	1	1	9	1	8	2	19	6	17	6

Table 5.2 3 Results of Classification Report

	n=20			n=50			n=100		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
Cosine Similarity	88%	93%	90%	79%	84%	80%	73%	76%	75%
Euclidean Distance	82%	85%	85%	78%	82%	80%	74%	78%	77%

The data from Table 5.2.2 and 5.2.3 shows how the algorithm performs. It seems that cosine similarity and euclidean distance has almost the same level of precision, accuracy, and recall. The highest score 90% of accuracy is obtained when using testing data of 20. However when the data testing was increased to 50 and 100 , the score dropped and stayed at >75%. This could happen because when the testing data is small, there is less chance of this algorithm making mistakes

To get the above results, it is necessary to determine a suitable threshold. This project uses 30 times of looping to find the best threshold. Below is the results from the loop :

Table 5.2 4 Finding Best Threshold with 60 Data Testing

	Threshold/ Highest Precision	Threshold/ Highest Recall	Threshold/ Highest Accuracy
Cosine Similarity	0.014 / 84%	0.017 / 80%	0.017 / 80%
Euclidean Distance	0.17 / 85%	0.18 / 81%	0.18 / 81 %

From the Table 5.2.4 shows the result of the loop. The best threshold for Cosine Similarity to get highest precision (84%) is 0.014 , highest recall (80%) is 0.017 , and highest accuracy (80%) is 0.017. The best threshold for Euclidean distance to get highest precision (85%) is 0.17 , highest recall (81%) is 0.18 , and highest accuracy (81%) is 0.18. Based on data above , this project determine to uses 0.014 threshold for cosine similarity and 0.17 threshold for euclidean distance , in order to get the highest precision score.

Table 5.2 5 Performance Time

	Performance (Avg. Time)
Algorithm	<1 seconds
User taking Attendance	<10 seconds

The data from table 5.2.5 are the performance measure of the website and algorithm in terms of taking attendance. This algorithm can complete authentication task less than 1 seconds while the average time spent by users to make attendance is less than 10 seconds.

This performance results can be obtained using the same hardware as well. This project uses Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz (4 CPUs), 8192MB RAM , and NVIDIA GeForce MX130 GPU with CUDA Enabled.

The use of this system still has some disadvantages in addition to the above-mentioned advantages. With the correct face images, this system can still be manipulated. so that users who gonna take attendance can tricked this system using only their photo picture. Below is an example of taking attendance using a picture :

Figure 5.2.1. Experiment Fraud of Face Authentication

