Face Recognition for Attendance System Detection

Rudy Hartanto
Department of Electrical Engineering
and Information Technology
Gadjah Mada University
Yogyakarta, Indonesia
rudy@ugm.ac.id

Marcus Nurtiantoro Adji
Department of Electrical Engineering
and Information Technology
Gadjah Mada University
Yogyakarta, Indonesia
mna@ugm.ac.id

Abstract—**These days, biometric authentication methods begin growing rapidly as one of promising authentication methods, besides the conventional authentication method. Almost all biometrics technologies require some actions by user, which are the user needs to place funds on the scanner to set the fingers or the hand geometry detection. The user shall stand still in a fixed position in front of the camera for iris or retina identification purpose. The face recognition method has several external advantages compared to the other biometric methods because this method can be done passively without explicit action or should be held by the user since the face image can be obtained by the camera from a certain distance. This method can be especially useful for mission and supervision.**

This research would develop and implement the face recognition system consists of four stage process. The four-stage process were face detection process using skin color detection and Haar Cascade algorithm, alignment process that contains face features normalization process, feature extraction process, and classification process using LBPH algorithm. Furthermore, the process would be continued to current system replacement with the system that has been built.

The experimental results show that the system can recognize the faces captured automatically by the camera accurately.

Keywords— ***biometric identification, skin detection, Haar Cascade, LBPH***

I. Introduction

The identification process to determine the presence of a person in a room or building is currently one of the routine security activities. Every person who will enter a room or building must go through several authentication processes first, that later these information's can be used to monitor every single activity in the room for a security purpose. Authentication process that is being used to identify the presence of a person in a room or building still vary. The process varies from writing a name and signatures in the attendance list, using an identity card, or using biometric methods authentication as fingerprint or face scanner.

Biometric-based authentication technique becomes one of the most promising methods [1, 2]. Nevertheless, the biometric authentication method used is still lacking and takes a relatively long time. The fingerprint scanner requires the user to place a finger on the scanner, and the face scanner method requires the user to set up their face position according to the position of the scanner.

This research will propose the authentication process based on face recognition using a web camera or CCTV

II. Related Works

The current biometric methods begin to evolve into one of the promising authentication methods compared to conventional authentication methods. The conventional authentication process is done by writing the name, address, and signature, or assigning someone by giving access to a physical or virtual realm using a password, PIN (Personal identification number), smart card, plastic card, token, key, etc [3]. Password and PIN are difficult to remember and in several cases, those are easy to steal or suspect. One of the biometric authentication methods is by using face recognition method.

Research on face recognition process has been done for a quite long time and continue to be developed until now. According to Viola and Jones [4], there are 3 important keys for object detection in machine learning. The first one is the image representation that able to create object features being detected in a short period of time. The second one is the algorithm based on AdaBoost that select important features in an object. The last one is to build a classifier according to a cascade that can override the background object in a short period of time.

In 2003, Viola and Jones [5] resumed their research based on their past research in 2001 about face detection not in an upright position but at an angle of 60°. They used a decision tree for the face detection and got satisfactory results.

Lienhart and Maydt [6] conducted a research based on the research conducted by Viola and Jones before. They included Haar-like features to the detector and obtained a 10% error reduction compared to the previous research and after the optimization, the error reduction increased up to 12.5%.

One of the deficiencies that still existed in face detection method is the heavy computation during the classifier training. This problem was overcome by Minh-Tri Pham and Tat-Jen Cham [7] who conducted a research to reduce the time required for training using statistical principles. The results obtained were quite significant in reducing the required computational time.

Face recognition approach by encoding face microstructure using learning based encoding method was conducted by Cao et al. [8]. They used unguided learning methods to learn an encoder from the training sample, which automatically gets a better trade-off between discriminatory power and invariant. Furthermore, a PCA (Principal component analysis) is used to obtain a compact face descriptor. The results of the research were able to recognize the face up to 84.45%.

According to the researchers described above, this research would develop a prototype a face recognition system that can be used in real time. This research would be divided into two stages. First, the face detection process is done by using skin color detection algorithm and Haar Cascade. Later, the eye position identification will be done as one of the feature extraction processes. The face recognition and classification process were simulated using the LBPH (Local Binary Patterns Histograms) algorithm.

978-1-5386-4739-4/18/$31.00 ©2018 IEEE

III. Proposed Method

The proposed face recognition system is shown in Fig. 1. The process began by capturing the face image using the camera. Afterward, the face detection process using skin color detection and face motion tracking was performed. This process also carried out localization process of eyes, lips, and face borders positions. Furthermore, the face alignment process was performed, including the face size normalization process and variations of lighting on the face. The next step was the face feature extraction, which later would be used in the matching process.

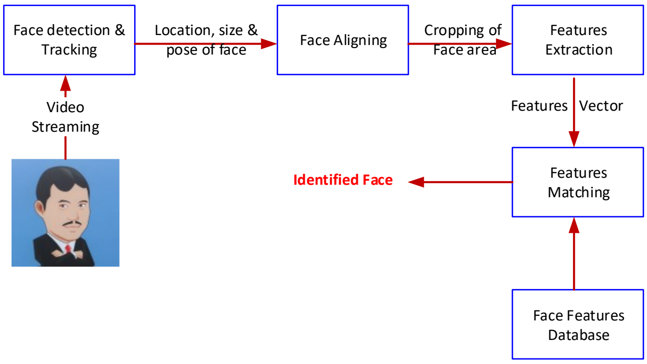


Fig. 1 Proposed System

A. Face Detection and Tracking

The camera captured images will be stored in the memory buffer for the further skin color detection process. This process aimed to detect the skin color of an image captured by the camera. The obtained image generally had the RGB format. To simplify the skin color detection process, the RGB format was converted to a YCrCb format to separate the intensity of Y using colors (chromaticity) expressed in two variables, Cr and Cb. In modeling the skin colors, only Cr and Cb information were used, thus the change effect in lighting intensity could be minimized. The saturated area of the light caught by the camera had stable Cr and Cb value, thus the values of Cr and Cb were a reliable information for the color classification process. Converting the RGB format to YCrCb used the following equation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.29900 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500000 \\ 0.500000 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where Y is the luminance (color intensity), Cb is the blue component, and Cr is the red component [9]. This method was also used to eliminate the background image which usually has different colors with the face skin color.

The next step is the conversion process from RGB format to Grayscale was performed. The face detection process was performed to determine the face location using Haar Cascade algorithm, as shown in the flow diagram in Fig. 2.

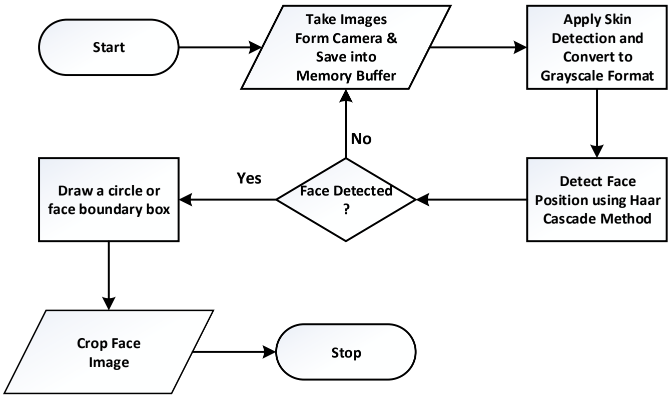

Start
Take Images Form Camera & Save into Memory Buffer
Apply Skin Detection and Convert to Grayscale Format
No
Yes
Draw a circle or face boundary box
Face Detected ?
Detect Face Position using Haar Cascade Method
Crop Face Image
Stop

Fig. 2 Face Detection and Tracking Process.

If the face image had been detected, it would draw a box that covers the face as an ROI (Region of Interest). This ROI box would be used to crop the image, thus only the face image would be processed by the next process to save the computation time. Fig. 3 shows the results of the user face detection process, marked with a white box image that would follow the user's face movement, or in the other words was able to track the movement of the user's face.

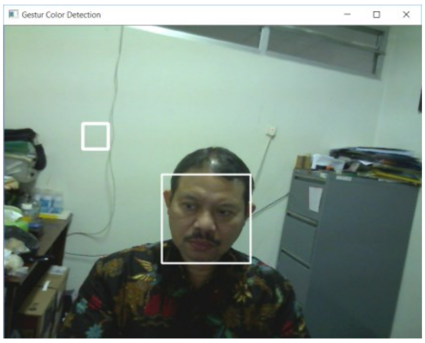


Fig. 3 Face detection and tracking result with ROI.

B. Features Extraction and Matching

This process began by aligning the ROI face input from the previous face detection process as shown in Fig. 4. The alignment process consisted of normalizing and adjusting the face size due to the variations of the ambient light. This process was necessary thus in the next process there would be no too far result difference because of the detected facial size, due to the different distances between the face and the camera and difference variations of the environmental light intensity during capturing the image.

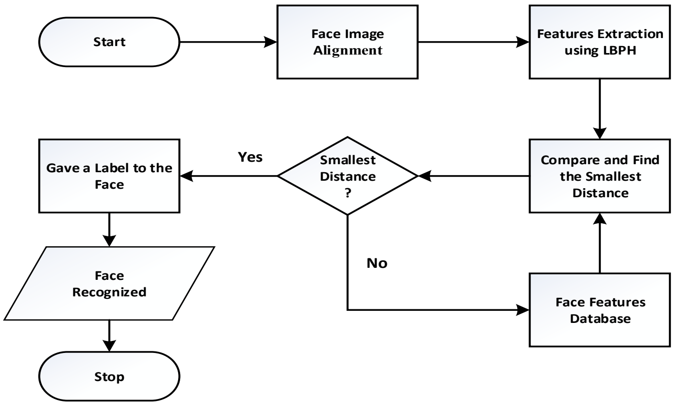


Fig. 4 Face alignment, feature extraction, and matching process.

The feature extraction was performed using LBPH algorithm. The detected face would be compared to all faces in the database to find the most similar face to the detected face. The database was stored using CSV file format to show the names and directories of the faces that exist in the database.

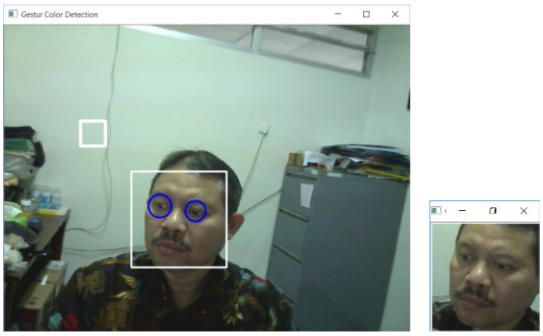


Fig. 5. Feature extraction to detect eyes position and face area.

Fig. 5 shows the feature extraction results could detect the user's eye position, marked with a blue circle symbol. Later, by using the distance between the center of the circle combined with the position of user's nose and mouth position, the identification and matching process would be performed.

IV. Result and Discussion

This section will be discussing the system performance testing process after the implementation was completed. This face recognition system was developed using Cpp software on Visual Studio 2013. The system performance testing was performed in 2 stages namely, face detection stage and face recognition stage.

A. *Face Detection*

The face detection process was performed using Haar Cascade algorithm. The initial stage of this test was to determine the performance of the face detection process within frontal conditions or face to face with the camera. The system indicator could detect the face captured by the camera by the appearance of a green box enclosing the detected face, as shown in Fig. 3.

The next process is to determine the system capability to detect faces that were not directly facing the camera, or in a condition where the face made a certain angle to the camera as shown in Fig. 6. The testing was done several times by making various face angle to the camera until the system could not detect the face anymore. According to the test results obtained, the face tilt maximum angle that could still be detected by the camera was about 30 °. Thus, if the face angle of the camera has exceeded 30 °, the face can no longer be able to be detected by the system.

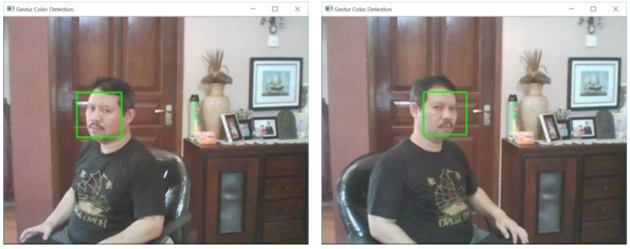


Fig 6. The results of face detection in conditions form a certain angle.

The next testing was to determine the system capability to detect the head position or face position tilted to the left or to the right. The testing was done with various angles, as shown as Fig. 7. According to various angle differences performed either to the left or to the right, the system could detect the face up to 30 °. If the face angle of the camera has exceeded 30 °, the face can no longer be able to be detected by the system.

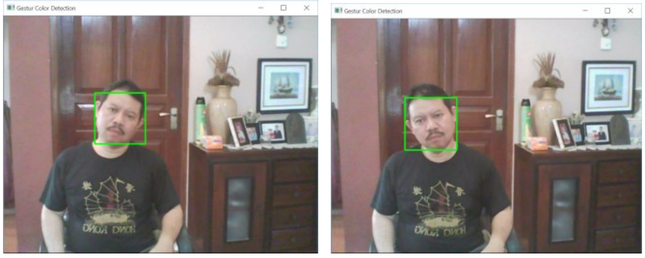


Fig. 7 Face detection at head position to form angle to left and right.

The next step was to determine the system capability to detect several faces with a various number of faces and different brightness of the image. The test was performed by using an image from a photo contained several people images with various variations of lighting. There were faces that used glasses or without glasses as shown in Fig. 8.

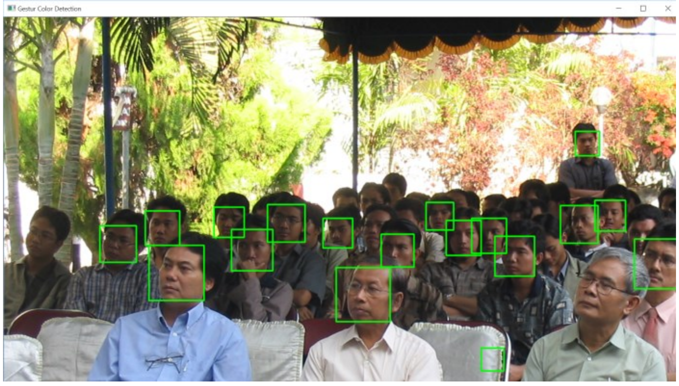


Fig. 8 Results of detection of many faces.

B. *Face Recognition*

The face recognition process was performed using LPBH algorithm because of its smaller computation load, thus it is relatively fast and can be used for the real-time recognition process. The LBPH concept is to not look the whole image as a high dimensional vector but to only review the local features of the important objects. The extracted object features only have low dimensions, for example in face recognition case, it will only review the face, eye, and mouth features.

The basic idea of LBPH is to make a summary of the local image structure by comparing each pixel with the neighbor pixel by taking a pixel as the center and developing (threshold) of the neighbor pixel's value. If the intensity value of the center pixel is greater or equal to its neighbor, it will be marked with a value of 1 and if it will not be marked with a value of 0. This process can generate the binary values for each pixel, such as 11001111. Thus with 8 pixels surrounding the center pixel, there would be 28 possible combinations called as Local Binary Pattern or often referred as LBP code. The example of LBP operators used a 3 x 3 pixels neighbors can be illustrated as in Fig. 9.

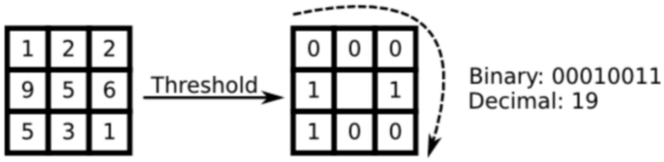


Fig. 9 Local Binary Pattern with 3x3 neighboring pixels.

In general, the LBP operator is given by the equation:

$$LBP(x_c, y_c) = \sum_{P=0}^{P-} 2^P s(i_p, i_c)$$

where (x_c, y_c) is the center pixels with intensity *ic*, and *ip* is the intensity of the neighbor pixels. S is a symbol function defined as:

$$s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise \end{cases}$$

Below is the testing result to get the eye and mouth features of a person's face marked with a blue circle for the eye position and a green circle for the mouth position, as shown in Fig. 10. The face identification process was performed by extracting the face features which is, in this case, was the position of the eyes and the mouth. The structure position, the eyes, and mouth position between one person and another is unique, thus it could serve as a feature for identification and classification or recognition process.

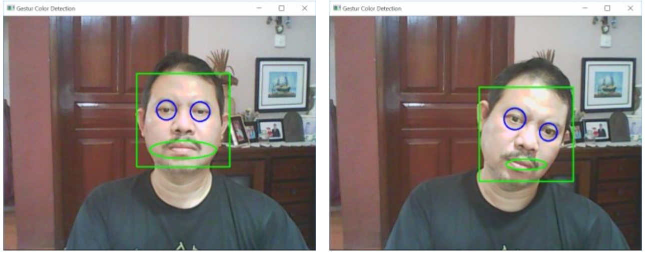


Fig. 10 Extraction results of eye and mouth features.

To be able to perform the recognition or classification process, it is necessary to create a database of faces that contain some different face positions you want to recognize. For each person, a minimum of 15 different face poses would be stored in the database, as shown in Fig. 11.

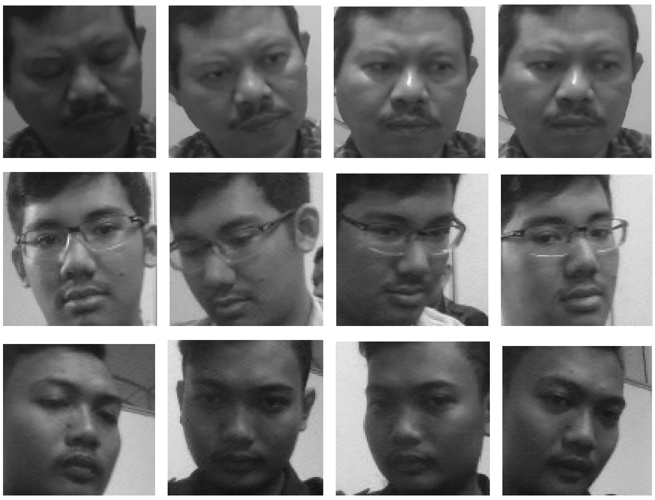
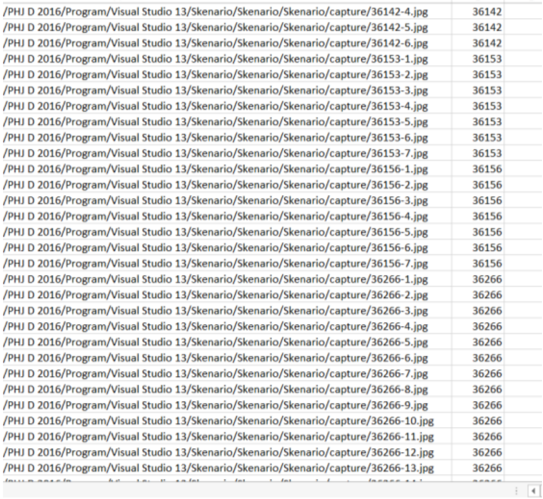


Fig. 11 Example of faces with different pose stored in database.

The face database will be collected in a specific folder that later would be called by the application to be used to be compared with the face captured by the camera. The results of the comparison would determine whether the face is common with the face stored in the database, or in the other words whether the face can be recognized or not.

For each face in the database, it would be labeled as a number indicating the identity of a person. Afterward, the face file location in the folder and the label would be arranged in a matrix stored in CSV format. The face recognition application would read the face file and its label through the matrix as shown in Fig. 12. The first column consisted of face image file location path stored in database folder, while the second column contained the identifiers of each face. For one person, it needs to store different various face positions files with the same label or identity.



/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36142-4.jpg	36142
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36142-5.jpg	36142
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36142-6.jpg	36142
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-1.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-2.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-3.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-4.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-5.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-6.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36153-7.jpg	36153
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-1.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-2.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-3.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-4.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-5.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-6.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36156-7.jpg	36156
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-1.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-2.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-3.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-4.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-5.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-6.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-7.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-8.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-9.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-10.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-11.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-12.jpg	36266
/PHJ D 2016/Program/Visual Studio 13/Skenario/Skenario/Skenario/capture/36266-13.jpg	36266

Fig. 12. Examples of face and label database matrices.

The example of the face identification is shown in Fig. 13. The red box enclosing the face indicates that the faces are detectable and the text above the box indicates the identification result, which in this case the identity is 12544. The system could detect and recognize a face though with a complex background image. In this case, Haar Cascade face detection algorithm could detect the position of the face precisely, furtherly to cut the face area according to the size of the box surrounding the face. The size of the box will vary according to the size of the face pixels captured by the camera. For the recognition process, only pixels in the box that will be processed with LBPH algorithm, thus the computing becomes smaller, lighter, and faster.

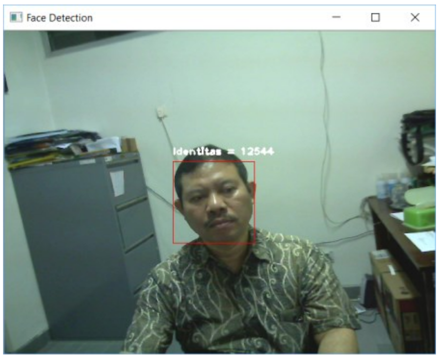


Fig. 13. Face identification result.

Fig. 14 shows the face identification process inside the classroom using CCTV camera. Out of 10 people who were inside the room, the system was only able to detect faces from 3 people. This was because CCTV camera resolution used was relatively low, which was about 640 x 480 pixels and the distance between the face and the camera was quite far, between 3 to 6 m. Thus, the face size that caught by the camera was relatively small, which meant the number of pixels on the face area was also relatively small. This made it more difficult to extract the face features properly, so the level of recognition also decreased.

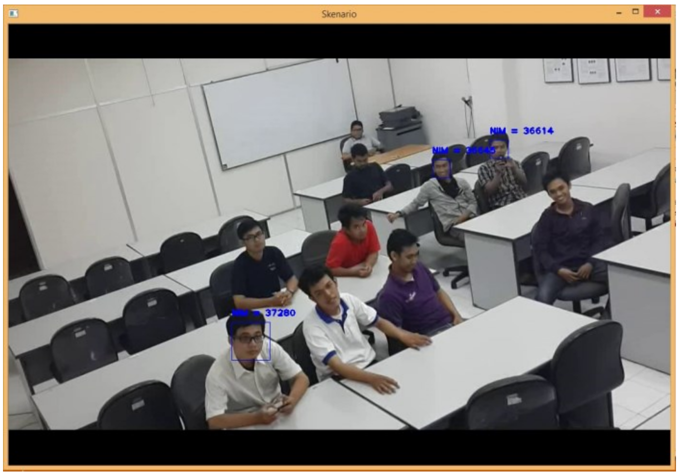


Fig. 14. Face identification in the classroom using CCTV camera.

Table 1 shows the face recognition accuracy result percentage with various light intensities ranging from 7 lux, 15 lux, until 24 lux. The accuracy rate of recognition was quite high, achieving a mean of 98.2% for high illumination intensity (24 lux), and 94.7% for lighting intensity of about 7 lux. Thus, the variations in light intensity are not very influential on face recognition accuracy.

TABLE 1 THE EFFECT OF LIGHT INTENSITY VARIATION TO FACE RECOGNITION ACCURACY.

Video Sample	**Light Intensity**		
	24 Lux	***15 Lux***	***7 Lux***
	Level of Face Recognition Accuracy (%)		
1	99	100	99
2	100	100	100
3	100	98	97
4	97	95	90
5	100	98	100
6	96	90	82
7	96	99	95
Average	**98.2**	**97.1**	**94.7**

This is the result of using the YCrCb color system which creates a separation between the intensity element (Y) and the chrominance element (Cr and Cb). Thus, the change of the environment light intensity only affects the intensity component of Y and less affects the chrominance component. Therefore, the recognition accuracy level does not have much difference.

The testing results of the distance effect of the face to the camera are shown in Table 2. The testing was performed by adjusting the camera distance to the face from 40 cm to 90 cm.

TABLE II. TESTING OF EFFECT OF FACE DISTANCE VARIATION TO CAMERA ON ACCURACY.

Video Sample	**Distance (cm)**			
	40	***60***	***80***	***90***
	Level of Face Recognition Accuracy (%)			
1	99	99	100	100
2	100	96	0	0
3	100	100	64	33
4	97	100	24	0
5	100	75	100	0
6	96	99	66	0
7	96	98	49	100
Average	**98.2**	**95.4**	**67.2**	**58.3**

The farther the distance of the face to the camera, the detection accuracy will also decrease. This is because the further distance of the camera to the face, the size of the face area caught by the camera will also be smaller, which means the number of pixels in the face area becomes less. The number of pixel on the face area will affect the number of features that can be extracted. The fewer the number of pixels in the face area or the smaller the resolution of the face, the fewer extractable features will be. Thus, the detection and face recognition accuracy will decrease even more.

CONCLUSION

Based on the observation, testing, and discussion results can be drawn several conclusions as follows. The system can recognize and identify the face well with an accuracy of 98.2 %, at a face distance 40 cm from the camera with adequate lighting (24 lux). The face recognition accuracy is still quite good amidst the variation of lighting intensity, with a minimum of 7 lux lighting intensity and variation of the distance between the face and the camera, between 40 cm to 90 cm.

REFERENCES

[1] S.-H. Lin, "An Introduction to Face Recognition Technology," *Informing Science Special Issue on Multimedia Informing Technology,* vol. 3, no. 1, 2000.

[2] J. Wayman, A. Jain, D. Maltoni and D. Maio, Biometric System, Technology, Design and Performance Evaluation, Springer, 2005.

[3] R. Jafri and H. R. Arabnia, "A Survey of Face Recognition Techniques," *Journal of Information Processing Systems,* vol. 5, no. 2, pp. 41-68, 2009*.

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE,* vol. 1, 2001.

[5] P. Viola and M. Jones, "Fast Multi-view Face Detection," Mitsubishi Electric Reseach Laboratory, 2003.

[6] R. Lienhart and J. Maydt, "An Extended set of Haar-like Features for Rapid Object Detection," in *2002 International Conference on Image Processing*, New York, 2002.

[7] M. Pham and T.J.Cham, "Fast Training and Selection of Haar Features using Statistic in Boosting-based Face Detection," in *2007 IEEE - International Conference on Computer Vision*, 2007.

[8] Z. Cao, Q. Yin, X. Tang and J. Sun, "Face Recognition with Learning-based Descriptor," in *Conference: Computer Vision and Pattern Recognition (CVPR)*, 2010.

[9] P. Hojoon, "A method for Controlling Mouse Movement using Real-Time Camera," Brown Computer Science, 2010.