Face recognition Using the Haar Cascade Classifier and Local Binary Patterns Histogram Algorithms to Detect and Identify Faces for Attendance

[1]Ni Kadek Ayu Purnaningsih
Student of Computer Engineering
Telkom University
Jawa Barat, Indonesia
ayupurnaningsih@student.telkomuniversity.ac.id

[2]Dr. Meta Kallista
[1]Computer Engineering
LecturerTelkom University
Jawa Barat, Indonesia
metakallista@telkomuniversity.ac.id

[3]Faisal Candrasyah Hasibuan
[2]Computer Engineering Lecturer
Telkom University
Jawa Barat, Indonesia
faicanhasfcb@telkomuniversity.ac.id

Abstract — In the era of globalization, especially in the field of education, student attendance tracking holds significant value for monitoring and managing participation within the teaching and learning process. Face detection and identification play a pivotal role in various modern technological applications, such as facial recognition and facial expression analysis. In the development of this system, a biometric approach using face recognition is employed, leveraging the Haar Cascade Classifier method for face detection in images, alongside the Local Binary Pattern Histogram (LBPH) method for facial identification through texture patterns. The system's implementation is conducted using the Python programming language and the OpenCV library. Testing is performed to recognize faces under diverse conditions, including variations in distance, light intensity, facial orientation, background, and accessories. Face detection and identification time range from 0.04 - 0.08 seconds, and a distance range of 30 cm - 150 cm.

Keyword — Face Recognition, Haar Cascade Classifier, Local Binary Pattern Histogram (LBPH), OpenCV.

I. INTRODUCTION

Face recognition is one of the biometric technologies that has been studied and developed by experts, as it utilizes facial recognition algorithms as identity data. The human face holds abundant information and distinct characteristics, making it widely used for personal identification. In addition to displaying emotions and attention, the face can also be employed for identifying an individual. One identification technique applied in biometric technology involves using the face as the primary parameter for attendance tracking systems in the realm of education [1]. Human face detection is also one domain within computer vision applications [2]. Heri Pratikno created a Realtime Face Recognition-Based Attendance System using a webcam with the PCA method. The device employs the Haar-Cascade Classifier algorithm with normal lighting, and individual has 25 stored images. The average recognition accuracy of this device reaches 88% [3]. Therefore, the attendance recording

The automatic attendance of students will expedite the attendance logging process. The Haar Cascade Classification algorithm is utilized for face detection. Each object will be analyzed by the facial detector, and subsequently, the detected faces will be validated. If successful facial recognition is achieved by the detector through the LBPH method, the attendance system will then automatically record the presence and store it in the database. Consequently, the attendance recording process becomes more optimized.

II. THEORETICAL STUDY

A. Haar Cascade Classifier

In this study, the face detection process employs the Haar Cascade Classifier method. Essentially, the term "Haar" refers to a mathematical function (Haar Wavelet) that takes the form of a rectangle. Initially, image processing only involved examining the RGB value of each pixel, but this method proved to be ineffective [4].

Subsequently, researchers Viola and Jones developed an approach to process images, giving rise to Haar-Like features. Haar-like features process images in boxes, where each box contains several pixels. These boxes are processed, producing differences in values that indicate areas of light and dark. These values serve as the basis for image processing [5].

When using the Haar Cascade method, there are several types of image formats that can be processed, one of which is grayscale. The Haar Cascade Classifier is a step to achieve more accurate results by calculating Haar Feature values in large quantities and repeatedly. The Haar Cascade Classifier workflow is as follows:

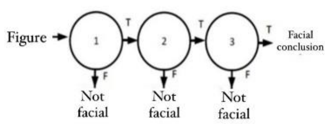


Fig. 1. Workflow of the Haar Cascade Classifier Method [5].

The Haar Cascade can be programmed to detect multiple objects, and the process involves determining the areas on the face with the highest probability. A face has skin and a certain level of pixel color on the skin. A segmentation technique is chosen for the pixel color on the face. Subsequently, it is validated with the Haar Cascade Classifier. If the pixels that are validated match their geometry, the system has identified the intended face. If they do not match, the system disregards it[5].

B. Local Binary Pattern Histogram (LBPH)

Face identification is an advanced stage in the face detection system, where the process of facial recognition employs template matching using LBPH (Local Binary Pattern Histogram). LBPH is a modern technique, formed by combining the LBP (Local Binary Pattern) algorithm with Histogram of Oriented Gradient (HOG). The facial image captured swiftly and in real-time by a webcam will undergo a comparison and matching process using histograms that have been extracted from the facial images stored in the database [7]. The operational principle of LBPH is illustrated in the following image:

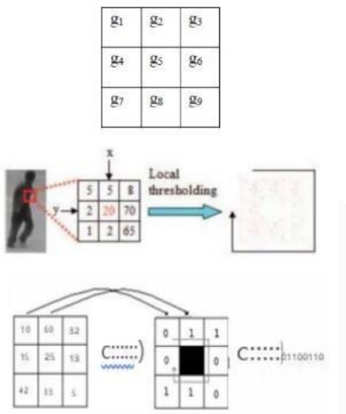


Fig. 2. Operation Principle of LBPH [7].

In the image above, the working principle of the Local Binary Pattern Histogram (LBPH) algorithm in image matching involves comparing the value of a pixel located at the center of an image with 8 pixel values surrounding it [5].

The pixel at the center is obtained by comparing its intensity with the intensities of the surrounding pixels. The value of the central pixel acts as a threshold for the surrounding eight pixels. In a matrix, the binary value at the center is compared with the values in the matrix surrounding it. If the value of the matrix at the center is higher than the surrounding values, it is assigned a binary value of 1; otherwise, it is assigned a value of 0. This process generates a binary pattern that describes the texture of the area under consideration. The resulting binary patterns are then used to create histograms, which represent the distribution of these patterns in the image. The LBPH algorithm uses these histograms to recognize and match facial patterns, enabling effective face recognition within the system. If the value of the matrix in the center is greater than the values around it, the surrounding matrix values will be assigned '1'; conversely, if the value of the center matrix is lower than the surrounding values, the surrounding matrix values will be assigned '0' [7].

The resulting 0s and 1s (8 binary values) are arranged in a clockwise manner, forming what is referred to as a threshold. From this arrangement of 8 binary values, they are converted into a decimal value [6]. The next step involves calculating histogram values to perform a comparison and matching of facial identities captured by the webcam with those stored in the database [7]. The equation to compute the threshold value is as follows:

$$D = \sqrt{\Sigma_i^n = 1\,(hist\ 1_i -\ hist\ 2_i)^2} \quad (1)$$

The value of D represents the comparison between the facial image stored in the database and the one captured by the camera or webcam.

C. Grayscaling

Grayscale is a type of image that deals with black and white color gradients, resulting in a grayscale effect. In this kind of image, colors are represented by intensity. In this context, intensity ranges from 0 to 255. A value of 0 represents black, while a value of 255 represents white. The conversion from a regular (RGB) image to a grayscale image is carried out using the formula [3]:

Grayscale = (0.299 * R) + (0.587 * G) + (0.114 * B)

In a nutshell, a use case is a series of scenarios combined for a common user goal [8].

D. Thresholding

Thresholding is used to control the level of grayness in an image. The thresholding process involves quantizing the image, and to apply thresholding to gray levels, the following formula can be used [9]:

"g(x,y) = " {1 if f(x,y) > T
{0 if f(x,y) ≤ T)

Explanation:
White if the gray level > T
Black if the gray level ≤ T

The thresholding process leads to a binary image, which contains only two gray level values: black and white. Generally, the goal of thresholding a grayscale image is to create a binary image.

E. OpenCV

The objective of this research is to create a system capable of counting the number of detected human faces in real time using the OpenCV library and Python. OpenCV (Intel Open Source Computer Vision Library) consists of at least 300 C functions, if not more. OpenCV is compatible with computers running on Windows or Linux operating systems. The OpenCV library is utilized to enhance the implementation of computer vision. This software offers a range of image processing functions, including image analysis and pattern recognition functions [10].

Python, a high-level programming language, functions as an interpreter-based language that can interact, is object-oriented, and is utilized across most systems, including Mac, Linux, and Windows. Its reputation as an easily mastered programming language stems from its clear syntax. Python can also be integrated with ready-to-use modules as well as efficient advanced data structures. The distribution of Python has been enhanced with various features, including a shell on the Linux operating system. OpenCV, in conjunction with Python, is employed to process images or videos (sequences of frames/images) according to their respective purposes, involving capturing images with a camera and then processing them on a computer [11].

III. METHOD

A. Flowchart of Attendance System Prototype

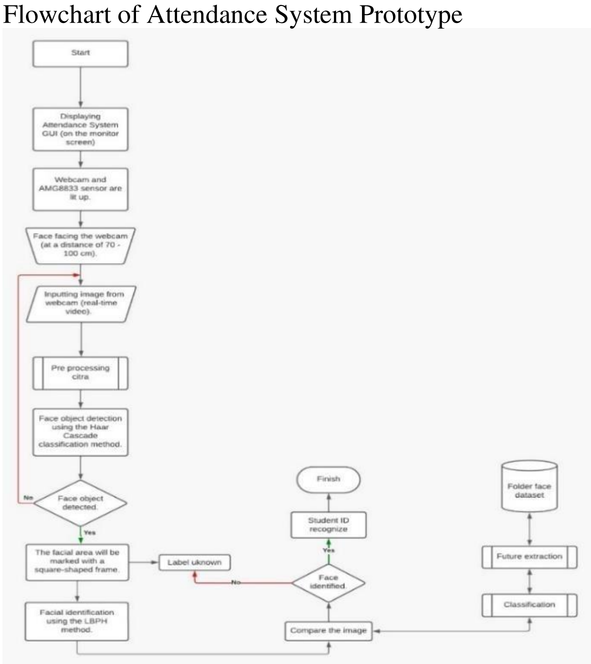


Fig. 3. Flowchart of Attendance System Prototype

Figure 3 represents the overall operation of the system. The images captured by the webcam are filtered using the Haar Cascade Classifier algorithm. If a face is successfully detected, the next step involves the system identifying the detected face. This process is carried out by the Local Binary Pattern Histogram (LBPH) algorithm. Subsequently, the system searches for the suitable threshold value interval on the face based on the stored image data in the database.

1. In the initial stage, prior to initiating the attendance process, the user operates the attendance system GUI on the monitor screen to activate the system along with input devices, namely the webcam.
2. Following the activation of the input device (webcam), the user is instructed to face the webcam from approximately 30 cm - 100 cm.
3. Afterward, the Haar cascade classifier algorithm will engage in detecting facial objects. If a successful facial detection occurs, the Local Binary Pattern Histogram (LBPH) algorithm will then work to match the detected face with the existing stored face database. During this stage, the algorithm will persistently operate; in cases where a face has not been detected, the process will be continually repeated until the face is recognized by the system.
4. Upon successful face detection, the face will be highlighted by a rectangular frame on the screen. Following this, the system will proceed to compare the recognized face with the database.
5. If a face is not detected, the system will display an 'Unknown' label on the screen.

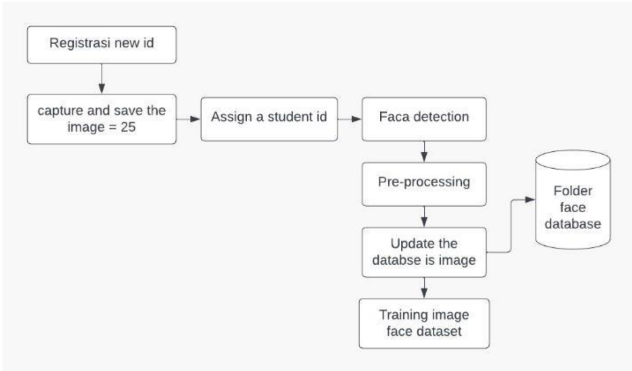


Fig. 4. Diagram Blok Facial data enrolment.

Figure 4 illustrates the system's process of training the dataset. Each ID will be registered, storing 25 facial pattern images per ID.

B. The accuracy of the algorithm Haar Cascade and LBPH

In identifying faces, this system employs the Haar Cascade Classifier algorithm, which is capable of rapid and real-time detecting a face. Meanwhile, for facial identification, the system utilizes the Local Binary Pattern Histogram (LBPH) algorithm. The system achieves accuracy under specific conditions. For instance, it can detect faces within a distance range of 30 - 100 cm from the webcam. For distances less than 30 cm, the system cannot detect and identify faces due to the proximity to the webcam. Similarly, for distances exceeding 150 cm, the system cannot detect and identify faces because the faces are too far from the webcam.

To function accurately, the system requires adequate lighting intensity. Diverse lighting intensity and facial positions are contributing factors to the accuracy of face detection and identification. This ensures that faces captured by the webcam can be detected effectively. Additionally, facial expressions, accessory usage, and background factors with interference or numerous objects significantly impact the webcam's accuracy. It is highly recommended to use a wall or screen, such as a green screen, to minimize these effects.

Table I. Algorithm Testing

Component	Algorithm
Details	Comparing, the system is capable of swiftly and real-time detecting and identifying a face
Measurement Method	Measuring the success of the algorithm in detecting and identifying faces within image segments that contain or involve faces
Testing	Performing classification on the method used to recognize faces quickly and accurately by utilizing an image processing library.

C. System Accuracy

Table II. Face Detection and Identification Results Based on Distance

No	Face Distance	Result	Description
1	10 cm	Not Detected	Too Close
2	30 cm	Not Detected	Too Close
3	50 cm	Detected	Sufficient
4	80 cm	Detected	Sufficient
5	100 cm	Detected	Sufficient
6	150 cm	Detected	Far Enough

Table III. Face Detection and Identification Results Based on Light Intensity

No	Condition of the Room	Result
1	Without Light	Detected
2	Normal	Detected
3	Additional Flashlight from Below	Detected Slightly Slower
4	Additional Flashlight from the Right	Detected Faster
5	Additional Flashlight from the Left	Detected Faster
6	Additional Flashlight from the Fron	Detected Faster

Table IV the lighting conditions significantly affect face detection and identification. Face detection would be faster if we used a flashlight for illuminating the surroundingsof the room.

Table V. Face Detection and Identification Results Based on Face Positions

No	Face and Body Positions	Result
1	Front View	Detected
2	Slightly Shifted to the Left	Detected
3	Slightly Shifted to the Right	Detected

D. Abbreviations and Acronyms

Table VI. Abbreviations

No	Abbreviation	Meaning
1	LBPH	Local Binary Pattern Histogram
2	LBP	Local Binary Pattern
3	HOG	Histogram of Oriented Gradients
4	OpenCV	Intel Open Source Computer Vision Library

IV. RESULTS AND DISCUSSION

A. The system can detect and identify faces. This test is conducted to determine whether the system can successfully detect and identify faces, such as when the webcam detects a face object and then identifies the personbehind the face.

Fig. 5. Testing Dataset Acquisition

Figure 5 register the user's facial pattern into the system by capturing a total of 20 - 25 facial images per person.

Fig. 6. Training Set Testing

Figure 6 The facial pattern training phase. This step is performed after registering the face, allowing the system to ensure that newly registered images are free from defects. The system rechecks whether they have been converted to grayscale, and any damage present in any of the images will become apparent during the training process.

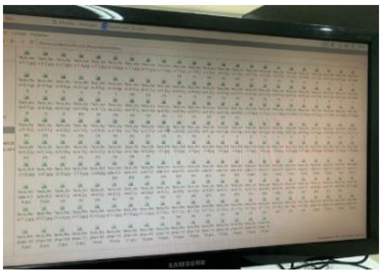


Fig. 7. Storage of Dataset Folder

Figure 7 The user's facial dataset stored in the directory/home/pi/Absensi/Source_Raspi/face,indicates successful facial pattern registration.

Fig. 8. Testing the Attendance Process

Figure 8 displays information based on the registered data.

B. Testing Based on Distance

Table VII. Testing Based on Distance

No	Face Distance	Result	Image
1	10 cm	Not Detected	-
2	30 cm	Not Detected	-
3	50 cm	Verified	
4	80 cm	Verified	
5	100 cm	Verified	
6	150 cm	Verified	

Represents the testing phase, where the attendance process is demonstrated. This stage showcases the system The table VII is a distance testing ranging from 30 – 150 cm. This testing aims to determine the optimal distance for detecting and identifying faces.

C. Testing Background Impact on Validation Speed

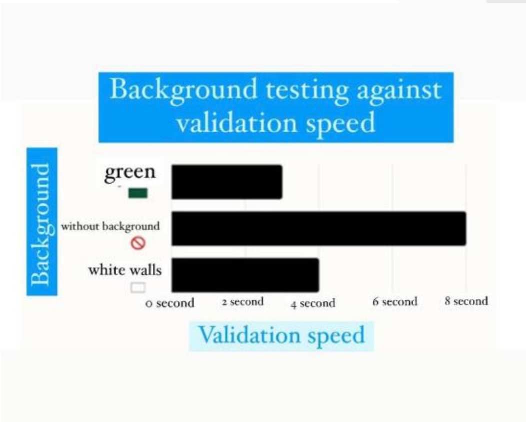


Fig. 9. Testing Background Impact on Validation Speed

In figure 9 a plain green background exhibited the fastest validation speed compared to others, taking only 3 seconds, while a white wall background showed a validation speed of approximately 4 seconds. On the other hand, validation without a background took the longest time, around 8 seconds.

D. Testing Various Facial Expressions

Table VIII. Testing Various Facial Expressions

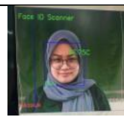
No	Expressions	Result	Image
1	Flat	Identified	
2	Slight Smile	Identified	
3	Wide Smile	Identified	
4	Angry	Identified	

Table VIII, testing conducted to determine whether the system can detect and identify faces based on different facial expressions.

E. Testing Various Accessories

Table IX. Testing Various Accessories

No	Aksesoris	Result	Image
1	Wearing Glasses	Identified	
2	Without Glasses	Identified	
3	Wearing a Hat	Not identified	
4	Using a Face Covering	Not identified	

Table Table IX, accessories used can affect the system in detecting and identifying faces, such as wearing hats and face coverings which can impact the system's facial identification.

F. Testing Based on Face Positions

Table X.Testing Based on Face Positions

No	Face Positions	Face Status	Image
1	Facing Forward	Identified	
2	Slightly Left	Identified	
3	Slightly Right	Identified	
4	Looking Upwards	Identified	
5	Looking Downwards	Not Detected	

Table X, face position greatly influences the system's ability to detect and identify faces. Faces positioned upwards and faces positioned downwards are not detected by the system.

G. Failure Test Experiment

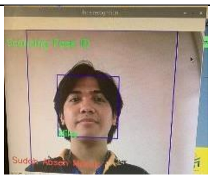
Table XI. Failure Test Experiment

Name Dataset	trials	Success	Failed	Level of success
Iful	40	38	2	95%
Aas	40	36	4	90%
Ayu	40	29	11	72.5%
Rayhan	40	22	18	55%
Success Rate				**78.125%**

Tablle XI, based on the testing, it can be concluded that the success rates, both individually and overall, vary. The datasets 'iful' (Success rate: 95%), 'Aas' (Success rate: 90%), 'Ayu' (Success rate: 72.5%), 'Rayhan' (Success rate: 55%), and the overall success rate is 78.125%."

H. The Subject Changes with the Object Beta Testing Test

Table XII. The Subject Changes with the Object Beta Testing Test

Result	Images in the Dataset	Images during Testing
Not Detected		
Detected		
Not Detected		
Detected		

Based on Table XII, it represents the results of testing the face detection system's ability to handle variations in facial changes. There are several findings and insights that can be extracted from this table. Testing on the subject named "Mike" indicates that a change from regular hairstyle to long bangs results in the system failing to detect the face. This might be due to drastic changes in facial features that the detection algorithm relies upon. Testing on the subject named "Ipul" shows that the system is still capable of detecting the face even with changes in hairstyle, whether it's short bangs or slightly shorter bangs. This suggests a level of tolerance towards minor variations in hairstyle. The test results on the subject "Ipul" wearing a head-covering jacket demonstrate that the system can still detect the face.

However, errors occur in detecting the name, indicating that head accessories can influence name recognition. From the conducted testing, it can be concluded that the system possesses the ability to detect faces with different hairstyles, particularly in smaller variations.Drastic changes in hairstyle, such as changing to long bangs, can cause the system to lose its ability to detect the face. The use of head accessories, like the head-covering jacket, still enables the system to detect the face. However, the presence of errors in name recognition suggests an impact on the identity recognition process.

I. Beta Testing Test

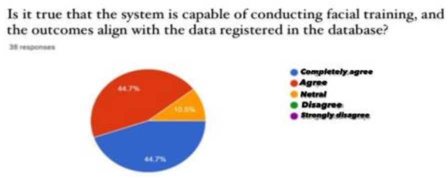


Fig. 10. Facial training process questionnaire

Figure 10 is the result of a questionnaire that has been distributed to respondents regarding questions about a system capable of conducting facial training, and the results align with the data that has been registered in the database.

The outcome of respondents' feedback includes 10.5% selecting neutral, while 44.7% expressed agreement, and 44.7% stated that they strongly agree.

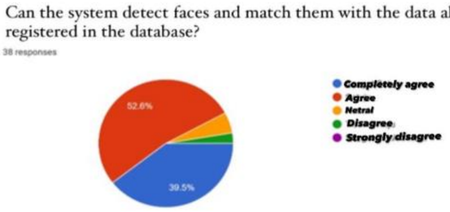


Fig. 11. Face detection process questionnaire

Through a questionnaire distributed to the respondents in the figure 11, regarding the question about the system's ability to detect faces and align with the data registered in the database, the results showed that 52.6% agreed, 39.5% strongly agreed, while 5.3% of respondents chose neutral, and 2.6% chose to disagree.

V. CONCLUSION

In the effort to establish an efficient attendance system within an educational environment, the author opts for the employment of the Haar Cascade and Local Binary Patterns Histogram (LBPH) methods. The Haar Cascade Classifier excels in swiftly and responsively detecting faces, making it well-suited for recording student attendance in time-constrained scenarios. Its ability to overcome variations in pose and lighting renders it a practical solution within diverse school settings. On the other hand, LBPH offers the capability of recognizing individual identities based on unique facial features. The synergy of both methods allows the system to efficiently detect faces using Haar Cascade, while LBPH ensures that attendance is captured with high accuracy. However, it is imperative to remember that in implementing these methods, the security of student data and privacy must remain a primary concern. By weighing the benefits and limitations of both approaches, an attendance system integrating Haar Cascade and LBPH can become a valuable tool in creating an organized and effective educational environment.

REFERENSI

[1] Munawir, L. Fitria, and M. Hermansyah, “Implementasi Face Recognition pada Absensi Kehadiran Mahasiswa Menggunakan Metode Haar Cascade Classifier,” *JURNAL NASIONAL INFORMATIKA DAN TEKNOLOGI JARINGAN*, vol. 4, no. 2, Mar. 2020.

[2] Paul Viola and Michaels J. Jones. (2001). Rapid Jones, Rapid Object Detection using Detection using boosted Cascade of Simple Features. IEEE Features. IEEE CVPR.

[3] A.J. Zenita, S. Budi, B. Agus. (2021). “Pengukur Suhu Tubuh Otomatis Berbasis Identifikasi Objek Menggunakan Metode Haar Cascade Clasifier”. [Online]. Available: https://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/view/4172.

[4] Angga Wahyu Wibowo, Aisyatul Karima, Wiktasari, Amran Yobioktabera, dan Sirli Fahriah. (2021). "Pendeteksian dan Pengenalan Wajah Pada Foto Secara Real Time Dengan Haar Cascade dan Local Binary Pattern Histogram". [Online]. Available: http://digilib.unila.ac.id/26828/4/skripsi_tanpa_bab_pembahasan.pdf, 2021.

[5] L. R. Febrin, Haryatmi. Emy, "Penggunaan Metode Haar Cascade Classifier dan LBPH Untuk Pengenalan Wajah Secara Realtime". [Online]. Available: https://jurnal.uisu.ac.id/index.php/infotekjar/article/view/4714/pdf.

[6] Raspberrypi.com. (2019). Raspberry Pi 4 Tech Specs. Diakses pada 31 Januari 2023, dari https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

[7] A. W. wibowol, A. Karima, W. A. Yobioktabera and S. Fahriah, "Pendeteksian dan Pengenalan Wajah pada Foto Secara Real Time dengan Haar Cascade dan Local Binary Pattern Histogram," vol. 9, no. 1, 2020.

[8] R. S. Pressman, Rekayasa Perangkat Lunak Pendekatan Praktis, Yogyakarta: Andi, 2012.

[9] R. (n.d.). Menentukan Thresholding. Retrieved from http://rubik.web.id

[10] Purnamawati S., R. F. Rahmat., dan Santana M., 2015, Aplikasi Pendeteksi Wajah Manusia Untuk Menghitung Jumlah Manusia, Lentera, Volume 15, juli 2015.

[11] Zein A., 2018, Pendeteksi Kantuk Secara Real Time Menggunakan Pustaka OpenCV dan Dlib Python, Sainstech, Volume 28 Nomor 2, Juli 2018, ISSN: 1410 - 7104