Journal of Information System and Technology Research

journal homepage: https://journal.aira.or.id/index.php/jistr/

Low-Cost CCTV for Home Security With Face Detection Base on IoT

Muhammad Akbar Syahbana Pane[1], Khairul Saleh[2], Andi Prayogi[3], Rahmad Dian[4], Ratu Mutiara Siregar[5], Raden Aris Sugianto[6]

[1,3,4,5,6] Prodi Sistem Teknologi Informasi, Institut Teknologi Sawit Indonesia, Nort Sumatera, Indonesia
[2] Jurusan Teknik Elektro, Sekolah Tinggi Teknologi Immanuel, Nort Sumatera, Indonesia

ARTICLE INFO

Article history:
Received Jan 1, 2024
Accepted Jan 31, 2024
Available online Jan 31, 2024

Keywords:
Face Detection,
CCTV,
IoT,
Python

ABSTRACT (10 PT)

Monitoring is an indispensable aspect of home surveillance, and with the advent of internet technology, it has become a crucial security measure. Despite the prevalence of CCTV cameras in the market, many still rely on analog and traditional technologies, particularly coaxial wires. This reliance results in additional expenses for wiring the CCTV system. Not only is this approach more costly, but the installation process also becomes more complex, as the picture data cable and control signal cable cannot be integrated. In response to these challenges, the objective of this project is to develop a security system capable of real-time object movement detection. This innovative system utilizes a webcam camera connected to a Raspberry Pi. The outcome of this study holds the potential to revolutionize home security by creating a low-cost CCTV system that can be monitored remotely through the Internet of Things. leveraging modern technologies like the Raspberry Pi and webcams, this project aims to eliminate the need for expensive and cumbersome coaxial wiring. The integration of these components allows for a more streamlined and cost-effective solution for home surveillance. Furthermore, the implementation of Internet of Things (IoT) connectivity enables users to remotely monitor their premises. This connectivity offers convenience and accessibility, allowing users to keep an eye on their homes from anywhere with an internet connection. The synergy of cost-effectiveness, real-time monitoring, and IoT connectivity positions this project as a promising advancement in the realm of home security. the development of a low-cost CCTV system with real-time object movement detection using a Raspberry Pi and webcam is a significant stride towards enhancing home surveillance. This project addresses the shortcomings of traditional CCTV systems, offering a more accessible and efficient solution for users who prioritize cost-effectiveness and remote monitoring capabilities.

Corresponding Author:
Muhammad Akbar Syahbana Pane,
Prodi Sistem Teknologi Informasi, Institut Teknologi Sawit Indonesia
Jl. Rumah Sakit H., Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20371
akbarsyahbana@yahoo.co.id

1. INTRODUCTION

In today's world, life requirements and technology are advancing fast. The wireless world is becoming increasingly significant due to highly dynamic demands that are neither geographically or temporally constrained. Wireless technology is becoming more prevalent in various industries, including manufacturing and communications. [1]. The wireless network is becoming more prevalent in various applications, including industry and communications. With the advancement of time, it has become necessary for the community to preserve security in the area in which they live. The advancement of world technology is accelerating, as may be seen in the advancement of technology. Closer Circuit Television (CCTV) is a non-fatigued monitoring instrument that continually monitors the region in which it is installed[2]. Facial Recognition is a type of computer technology that uses information about a person's facial traits to identify him or her. Facial Recognition is a biometric method that is continuously evolving to meet the needs of modern civilization [3]. Facial Recognition is commonly used for authentication, security access systems, and personal identity authentication.

In previous research, it was explained that security monitoring still uses infrared-based technology which has weaknesses in terms of control distance and also in previous research, it also has weaknesses in that monitoring areas cannot recognize people who enter the area[1].

The Internet of things (IoT) is a web of embedded physical and electronic devices, sensors, actuators, and network connectivity that enables these items to collect and exchange data[4][5]. It is the fastest-growing wireless communication technology. As an extension of the Internet, the concept of a network that uses the Internet network to extend the benefits of continuously connected Internet connectivity for various reasons, including identity, location, tracking, monitoring, data exchange, and remote control. Monitoring is a component of security measures; home monitoring can be accomplished over an internet network with the IoT idea. By integrating CCTV equipment into an IoT system, users can remotely monitor the region's security surrounding their home.

2. RESEARCH METHOD

The research approach is one of creating and installing tools, in which the theory addressed is realized in the form of a tool that is immediately useful in daily life. The research created a monitoring camera that has the ability to recognize faces, a system built using a Raspberry Pi using machine learning methodology. This method uses a large database of people whose facial images are first created in the dataset.

2.1 Additional Hardware

To create CCTV using Raspberry Pi and the IoT concept, it is necessary first to understand the components that go into the monitoring camera. The components are as follows:

Table 1. Component Names

No.	Component Name
1	Raspberry Pi 3 Model B+
2	Webcam Logitech
3	Adaptor USB 2.5 A
4	Kabel USB
5	Kabel HDMI
6	Smartphone/PC

2.2 Diagram of a Raspberry Pi-Based Cctv Monitoring System

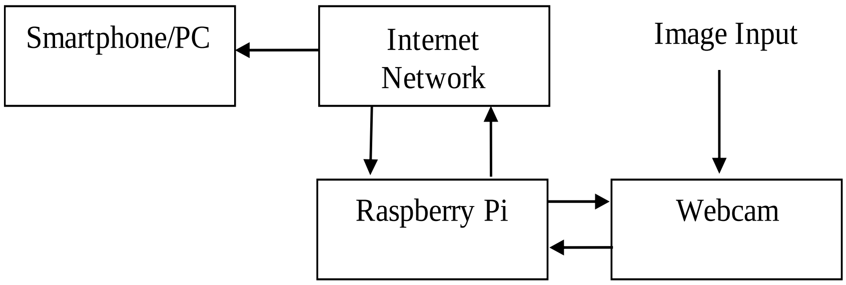


Figure 1. Schematic representation of a CCTV monitoring system

The figure describes the system configuration and its inputs and outputs. The input comes from a webcam camera that observes the surrounding location in this arrangement. Then the input image from the webcam will be analyzed by the Raspberry Pi and will perform image processing on the image relatively able to distinguish faces from the images inputted by the webcam. The output of facial recognition results from the Raspberry Pi will be transferred in real-time via the internet network and received by the Smartphone/PC anywhere and anytime as long as the smartphone/PC is connected to the internet network.

2.3 Algorithm for Recognized Faces

When coupled with the Python application software, the Raspberry Pi can perform face recognition algorithms directly but also requires the assistance of face detection algorithms that seek to identify actual items. Python's initial task for object identification is to perform face detection, followed by face recognition. Face recognition requires the webcam's camera to recognize a person's face using a database that has been loaded onto the Raspberry Pi's dataset. Additionally, the database for the Raspberry Pi dataset is comprised of pictures or webcam capture results.
The Raspberry pi is a small, inexpensive computer around the size of a credit card that is relatively new on the market. The Raspberry pi platform has gained popularity in recent years; it may be used for various electronics applications and projects and tasks performed

by a desktop PC, such as browsing, text editing, and gaming. Additionally, it supports high-definition video and includes an Ethernet port, allowing for easy connectivity with the most popular boards. Additionally, we can add Wi-Fi access simply by connecting the Wi-Fi dongle into one of the USB ports[6].

2.3.1 Facial Detection

Detecting objects or faces is a simple task for humans but difficult for computer vision-based systems. Computer vision science attempts to replicate the way people see (Human vision)[7]. However, this copycat process necessitates an extensive study[8]. The most often used technique for detecting faces (or any other item) is the "Haar Cascade classifier." Paul Viola and Michael Jones published a successful object detection approach called "Fast Object Detection Using Boosted Cascade of Simple Features" in 2001[9]. The Haar Cascade classifier is a machine-learning technique that involves training a cascade function on many positive and negative photos and finding objects in other images.

The Haar algorithm detects faces using statistical approaches. This approach makes use of a sample of haar-like features. This classifier uses images that are fixed in size (usually 24x24 in size). Haar's face detection algorithm works by applying a 24x24 sliding window approach to the entire image and determining if any portions of it are shaped like faces or not. Haar is also capable of scaling, which enables it to detect larger or smaller faces than the picture used in the classifier[8].

2.3.2. Haar-like feature

A Haar-like feature is a technique that is frequently employed in object detection. Haar is based on the Haar Wavelet, a box-shaped mathematical function that operates similarly to the Fourier function[8]. Haar-like features are rectangular features (square functions) that provide information about a particular image or image. Haar-like features operate on the premise of object recognition based on the feature's simple value rather than the pixel value of the item's image. This method has the benefit of being faster, as it is based on the number of pixels in a square rather than on each pixel value in a picture. Vehicle object detection in this work is a version of Viola and Jones's Haar-like features approach for face detection, which Lienhart later developed. Viola and Jones propose a method for detecting an object that combines four primary keys[10]:

a. The Haar feature is a simple square feature.
b. Fast feature detection with an integral image
c. The AdaBoost approach of machine learning
d. Combination of many features using a cascade classifier

The detection results are less accurate when only one function performs a Haar-like feature. The greater the detection filter level, the more precisely an object is recognized, but the procedure takes longer. A cascade classifier[11] is used to process many Haar-like features[12]. The Haar-Wavelet is a square wave (dark interval and light interval) compared to the average of the two-pixel values. The comparison satisfies the Haar feature requirements if the average intensity value exceeds the threshold[25]. This method is performed discretely for moving images such as video by sampling the video at a specific frame rate. The following figure 2 illustrates the numerous variants of the Haar-like feature:

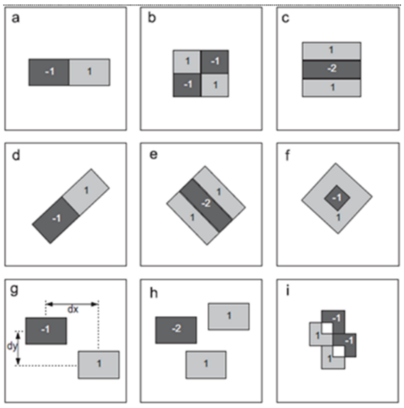


Figure 2. Variation in the Haar Square with Rectangle Standard Weight

Image caption 2:
a, b: Haar features proposed by Papageorgiou et al.
c: Viola and James' proposed Haar feature
d, e, f: Variations of the Haar features proposed by Leinhardt
g, h: Lietal decomposition of Haar-like features
i: Haar-like Viola and James feature to capture diagonal structure in object appearance

The formula for calculating the feature value in Figure 2 is shown in Figure 3

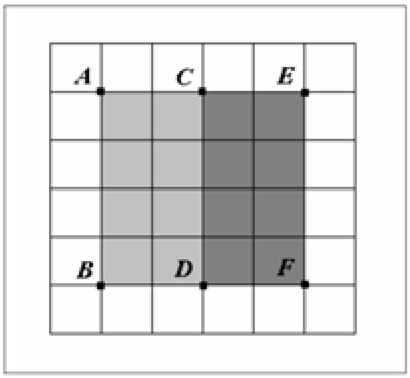


Figure 3. The Haar-like Square Feature

Feature Value(ABFE) = Number of Pixel Values(ABDC)-Number of Pixel Values(CDFE) (1)

If the Feature Value (ABFE) is greater than the threshold, the condition is considered to be met. As illustrated in Figure 2, if a feature is declared ineligible, no items are discovered in the ABFE area, and the square area is repositioned; however, if the ABFE square meets the features, the following feature rule is performed. When all of the feature requirements are met, the ABFE square is said to have an object. The following figure illustrates the process of detecting cars using the Haar-like feature:

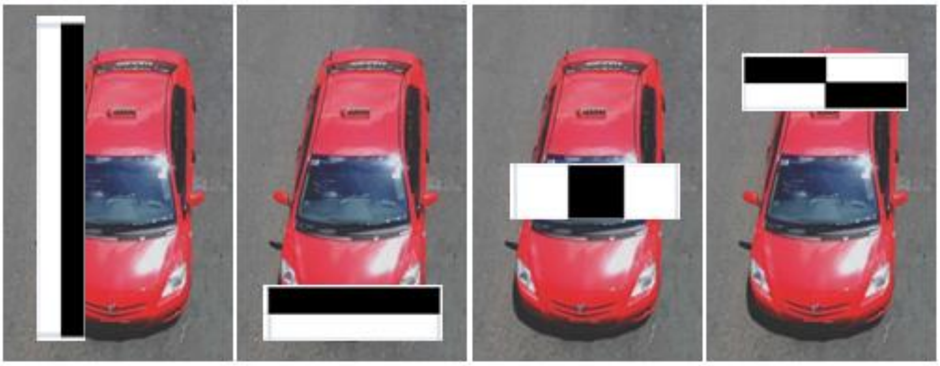


Figure 4. Vehicle Detection with Haar-like Feature

2.3.3. Integral image

A component RGB value is assigned to a digital image (a combination of red, green, and blue). As can be observed from the RGB value, the grayscale value (degree of gray) is determined using the following formula (2)[13]:

Grayscale_pixel = 0.2989R + 0.5870G + 0.1140B (2)

For example, if a pixel has the color values R=100, B=100, and G=100, the grayscale value calculated using formula two above is 99.99. In Figure 5, the following image has been transformed to grayscale:

Figure 5. Difference between Original Image and Grayscale

2.3.4. Local Binary Patterns Histograms

Local Binary Patterns Histograms (LBPH)[14] is a variant of Local Binary Patterns (LBP) that was initially defined in 1994 [15]. LBPH is a technique for analyzing the texture of a surface that uses statistical models and structures. LBPH compares the gray values of adjacent pixels. With a dimension of 3x3, the principal LBPH operator employs eight pixels in neighbor In from a middle pixel Ic. Here, face detection will be used. The algorithm requires many positive images (face images) and negative images (without faces) to train the classifier[23]. Then, features must be extracted from it. The good thing is that OpenCV comes with trainers and detectors pre-installed. To train a classifier on any object, such as a car or a plane, use OpenCV to accomplish this.

The LBPH algorithm utilizes the following four parameters:

a. Radius: utilized to construct the circular local binary pattern, the radius represents the radius surrounding the center pixel. Typically, it is set to 1.
b. Neighbors: the number of sample points used to create the circular binary pattern in the immediate area. Bear in mind that the greater the number of sample points included, the greater the computing cost. Usually, it is set to 8.
c. Grid X: the number of horizontally aligned cells. The greater the number of cells in the grid, the larger the dimensionality of the resulting feature vector. Usually, it is set to 8.
d. Grid Y: the vertical dimension's cell count. The greater the number of cells in the grid, the larger the dimensionality of the resulting feature vector. Typically, it is set to 8[16].

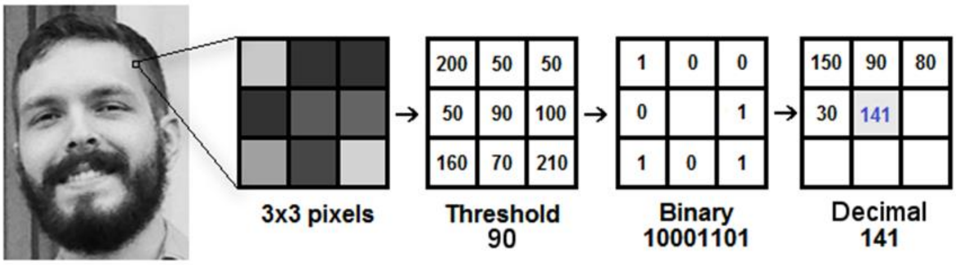


Figure 6. LBPH Threshold Process

According to Figure 6, the LBPH process can be described as follows. Suppose we have a grayscale image of a face and wish to retrieve a portion of it as a 3x3 pixel window. Additionally, this can be expressed as a 3x3 matrix, including the intensity of each pixel (0-255) [17]. Then, the canter value of the matrix that will be used as the threshold must be determined. This value will be used to establish the eight neighbors' new values[18]. It will create a new binary value for each neighbor of the core value (threshold). Then, assign a value 1 to any value equal or greater than the threshold and 0 to any value less than the threshold. At this point, the matrix will only contain binary values (ignoring the central value). Each binary value from each location in the matrix must be combined into a new binary value (e.g., 10001101). Then convert this binary value to a decimal value and put it to the matrix's center value, the original image's pixel[24]. After this method (LBP technique) will have a new image that more accurately portrays the original image's qualities. Equation 3 provides an explanation for LBP:

$$\boldsymbol{LBP_{P,R}(X_C, Y_C) = \sum_{p=0}^{P-1} S(g_p - g_c) 2^{\mathcal{P}}} \quad (3)$$

Where

$$\boldsymbol{S(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}} \quad (4)$$

Explanation

X_C, and Y_C : The pixel center coordinates are neighboring each other.
$\mathcal{P}$: Circular sampling point.
P : Number of sampling points.
g_p : Gray value from $\mathcal{P}$.
g_c : The average pixel value of neighbor and center values.
S (x) : thresholding Operator

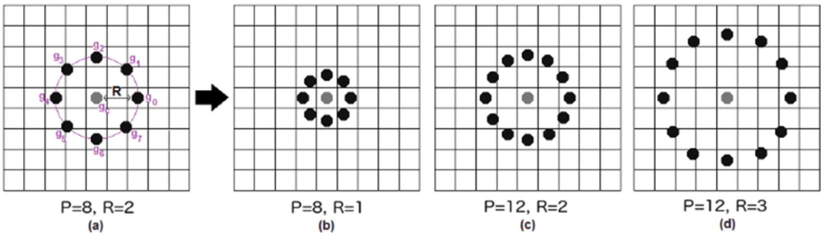


Figure 7. Local Binary Patterns of Variance

On Local Binary Pattern, the operator has a label denoted by the characters P and R. P is the number of neighboring pixels used in the computation. In contrast, R denotes the radius between the center and adjacent pixels.

2.4 Configuration of a Real VNC Server on a Raspberry Pi

Execute monitoring using the IoT idea, and the VNC Server software must be configured. Select License from the VNC Server status menu at the first stage, then sign in to the RealVNC account. Enter the account email and password in the sign-in box with the VNC account, as shown in figure 8, and the VNC Server will configure the Raspberry Pi automatically[19].

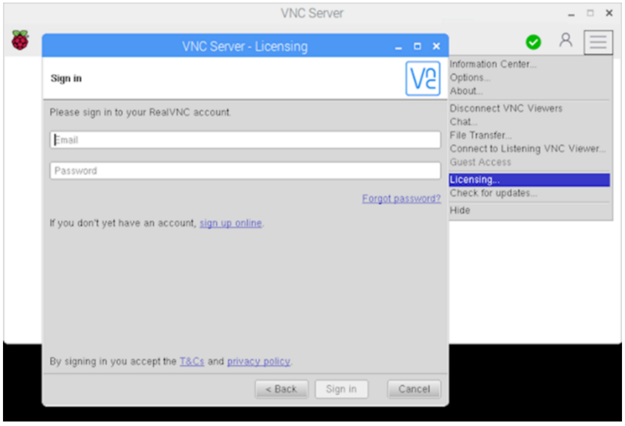


Figure 8. Logging into the VNC Server

Launch VNC Viewer on the Smartphone/PC that will be used to operate the raspberry pi and log in using the same account as on the Raspberry Pi's VNC server[20]. Connect to Raspberry Pi will appear automatically under the team name in the VNC Viewer window. Simply press or double click to connect VNC Viewer to the Raspberry Pi's VNC Server, as illustrated in Figure 9.

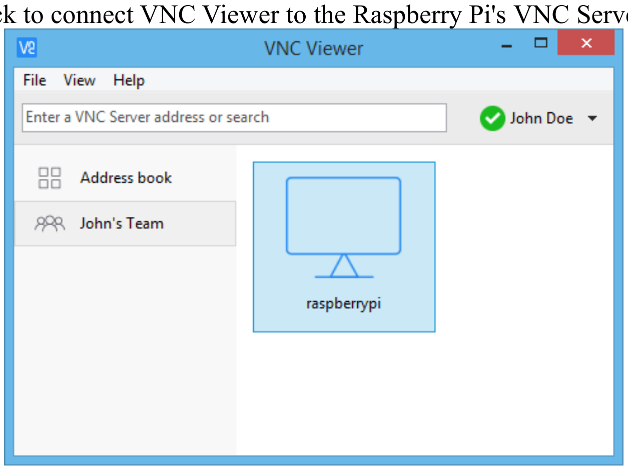


Figure 9. VNC viewer on a smartphone/PC

3. RESULTS AND DISCUSSION

In this section, using the Internet of Things system, objects will be viewed from a smartphone or a PC, and an analysis of face detection capabilities on CCTV will be presented in this part.

3.1 Prototype of a low-cost CCTV system with face detection

Raspberry Pi is a small computer with an operating system based on Rasbian OS[2]. Due to its small size, it is effortless to use it for various purposes, one of which is to pair it with a webcam that functions as a monitoring camera capable of monitoring specific areas and performing various functions. Additionally, facial Recognition is included[3]. The monitoring camera depicted in Figure 4.1 is a prototype built using a Raspberry Pi and a webcam.

Figure 10. Prototype of a low-cost CCTV system with face detection

3.2 Performance of software

Python 3.5.3 was used to construct a monitoring camera with real-time facial detection; this software utilizes an OpenCV package to accomplish the face recognition process using the LBPH method. To show the output from the webcam camera during monitoring, the Raspberry Pi can be used in conjunction with a Smartphone/PC that is linked to the Internet through Wi-Fi and has pre-installed software, as well as the Raspberry Pi. VNC Viewer is the program used to do Remote Desktops from the Raspberry Pi to the Smartphone/PC. Figure 11 shows the VNC Viewer window when an application executes on a PC.

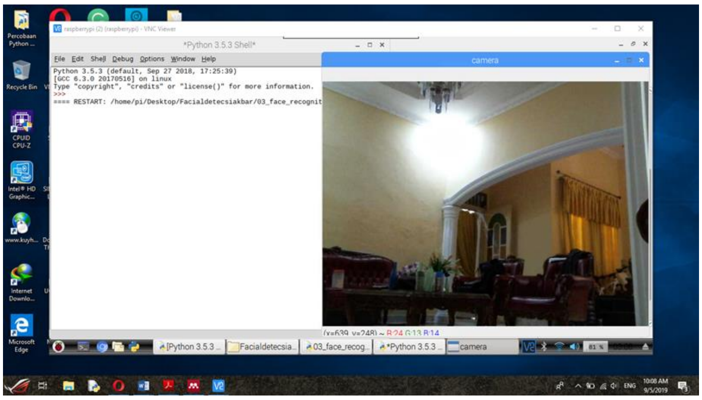


Figure 11. While program execution, the program's display

3.3 Analysis of Face Detection Capabilities

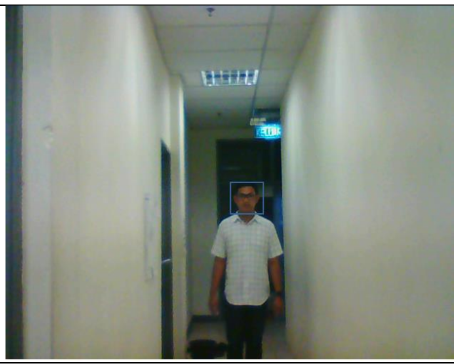
This section will explain how CCTV can do face detection using a webcam. The analysis will be conducted by comparing CCTV and humans to the distance between CCTV and humans, as shown in Table 2.

Table 2. Comparison between HAAR and LBP

Distance	HAAR	LBP
1 M	Detected	Detected
2 M	Detected	Detected
3 M	Detected	Detected
4 M	Detected	Detected
5 M	Detected	Detected
6 M	Detected	Not Detected

Table 2 shows that the capacity of surveillance robots to detect faces using two different ways with a 640 x 480 camera resolution results in a different detection rate than the existing distance. Haar cascade increases face detection performance even further, capable of detecting human face objects up to a distance of 6.5 meters. In contrast, the LBP cascade extends face detection capabilities only to a maximum of 5.45 meters. The maximum face detection results obtained using the Haar cascade and LBP cascade methods are listed in Table 3.

Table 3. Maximum Detection Results

Haar cascade detection	LBP Cascade detection
	

Facial Recognition can be used throughout the CCTV monitoring process. Recognized faces have been registered and have a database on a previously produced dataset; in this study, two IDs have been registered, namely ID1: Akbar and ID2: Reza. Following that, analyze the robot's capacity to detect faces. In Figure 4.5, the robot can distinguish faces with ID1 and ID2 at a distance of 40cm.

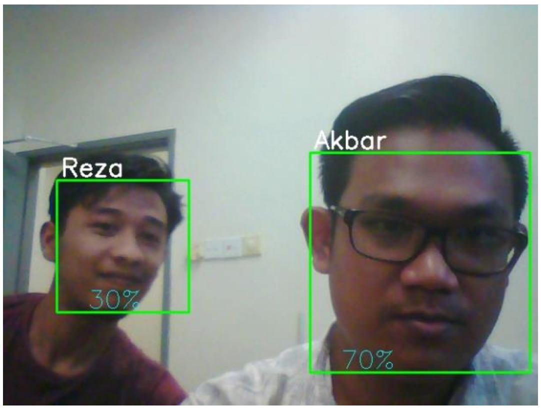


Figure 12. Face Recognition on CCTV

Distance-based facial recognition tests were also performed. CCTV cameras with a maximum distance of 2 meters and an interior resolution of 640 x 480 can conduct facial recognition services. CCTV can recognize people's faces at a distance of two meters, with ID1 being "Akbar," as displayed in Figure 13.

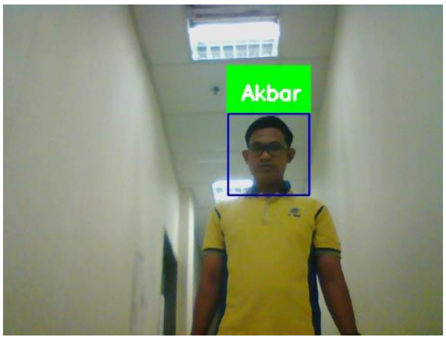


Figure 13. Face Recognition with a Distance of 2 meters

At distances greater than 2 meters and in interior situations with webcam cameras with a resolution of 640 x 480, the robot cannot recognize faces and instead recognizes ID as "unknown," as illustrated in Figure 14.

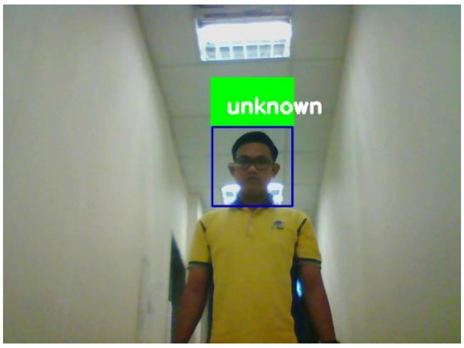


Figure 14. Face Recognition with a Distance of More than 2 meters

Experiment with two different individuals, one registered in the dataset and the other not. Face Recognition on CCTV indicates that the ID1 result is "Akbar," but the others cannot recognize the face and show the ID result is "unknown," as displayed in Figure 15.

Figure 15. Face Recognition with two People

4. CONCLUSION

The implementation of Internet of Things (IoT) connectivity enables users to remotely monitor their premises. This connectivity offers convenience and accessibility, allowing users to keep an eye on their homes from anywhere with an internet connection. The synergy of cost-effectiveness, real-time monitoring, and IoT connectivity positions this project as a promising advancement in the realm of home security. In conclusion, the development of a low-cost CCTV system with real-time object movement detection using a Raspberry Pi and webcam is a significant stride towards enhancing home surveillance. This project addresses the shortcomings of traditional CCTV systems, offering a more accessible and efficient solution for users who prioritize cost-effectiveness and remote monitoring capabilities.The design of CCTV to monitor areas using the Internet-based IoT has been successfully implemented using a Raspberry Pi and a webcam with a compact size and low cost. The LBPH Algorithm is successfully implemented on the Raspberry Pi and can perform facial Recognition on individuals already registered in the dataset on the Raspberry Pi.

References

[1] Q. Akariman, A. N. Jati, and A. Novianty, “Face recognition based on the Android device using LBP algorithm,” *ICCEREC 2015 - Int. Conf. Control. Electron. Renew. Energy Commun.*, pp. 166–170, 2015.
[2] S. Mallu, “PENDETEKSIAN GERAKAN MENGGUNAKAN INTERNET PROTOCOL CAMERA BERBASIS WEB,” *J. Ilm. Teknol. Inf. Terap.*, vol. I, no. 3, pp. 9–14, 2015.
[3] C. Liu, “The development trend of evaluating face-recognition technology,” *Proc. - 2014 Int. Conf. Mechatronics Control. ICMC 2014*, no. August 1994, pp. 1540–1544, 2015.
[4] N. A. Mathew and A. Client, “IoT based Real Time Patient Monitoring and Analysis using Raspberry Pi 3,” *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput.*, pp. 2638–2640, 2017.
[5] N. Patil, S. Ambatkar, and S. Kakde, “IoT Based Smart Surveillance Security System using Raspberry Pi,” pp. 344–348, 2017.
[6] P. I. Y. Sheikh, H. C. Chavan, J. S. Vyawhare, S. B. Mahajan, and R. S. Raut, “Wi-Fi Surveillance Robot Using Raspberry Pi,” no. April, pp. 25–29, 2016.
[7] M. Gutensohn, N. S. Session, and M. Gutensohn, “Developing a Natural User Interface and Facial Recognition System With

OpenCV and the Microsoft Kinect," pp. 1–6, 2018.
[8] W. S. Pambudi, B. Maria, N. Simorangkir, J. T. Elektro, U. I. Batam, and D. Obyek, "Facetracker Menggunakan Metode Haar Like Feature," *J. Teknol. dan Inf.*, vol. 2, no. 2, pp. 142–154, 2012.
[9] A. Mahmudi, "Deteksi Senjata Tajam Dengan Metode Haar Cascade Classifier Menggunakan Teknologi Sms Gateway," *Matics*, vol. 1, no. 1, pp. 27–30, 2014.
[10] L. W. Alexander, S. R. Sentinuwo, A. M. Sambul, T. Informatika, U. Sam, and R. Manado, "Implementasi Algoritma Pengenalan Wajah Untuk Mendeteksi Visual Hacking," *E-Journal Tek. Inform.*, vol. 11, no. 1, 2017.
[11] S. Al-Aidid and D. Pamungkas, "Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram," *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 62–67, 2018.
[12] R. Puri and A. Gupta, "CONTOUR, SHAPE, AND COLOR DETECTION USING OPEN CV – PYTHON," no. May, pp. 2–4, 2018.
[13] P. Pasumarti and P. P. Sekhar, "Classroom Attendance Using Face Detection and Raspberry-Pi," *Int. Res. J. Eng. Technol.*, vol. 5, no. 1, pp. 167–171, 2018.
[14] M. Phankokkruad, P. Jaturawat, and P. Pongmanawut, "A real-time face recognition for class participation enrollment system over WebRTC," *Eighth Int. Conf. Digit. Image Process. (ICDIP 2016)*, vol. 10033, no. August 2016, p. 100330V, 2016.
[15] W. S. M. Sanjaya, D. Anggraeni, K. Zakaria, A. Juwardi, and M. Munawwaroh, "The design of face recognition and tracking for human-robot interaction," *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua, no. c, pp. 315–320, 2018.
[16] N. Sontakke, S. Kulthe, R. Gaikwad, and T. Ladkat, "Automatic Face Recognition Techniques using LBPH," no. 3, pp. 1912–1916, 2018.
[17] A. Rahim, N. Hossain, T. Wahid, and S. Azam, "Face Recognition using Local Binary Patterns (LBP)," *Glob. J. Comput. Sience Technol. Graph. Vis.*, vol. 13, no. 4, pp. 469–481, 2013.
[18] N. Stekas and D. Van Den Heuvel, "Face recognition using local binary patterns histograms (LBPH) on an FPGA-based system on chip (SoC)," *Proc. - 2016 IEEE 30th Int. Parallel Distrib. Process. Symp. IPDPS 2016*, pp. 300–304, 2016.
[19] M. A. S. Pane, P. Ehkan, K. Saleh, and M. Irwanto, "A Mobile Surveillance Robot Over The Wifi Network Using Atmega 8," vol. 20, no. 5, pp. 20–26, 2018.
[20] N. Sakali and G. Nagendra, "Design and Implementation of Web Surveillance Robot for Video Monitoring and Motion Detection," vol. 7, no. 2, pp. 4298–4302, 2017.
[21] Huu-Quoc Nguyen, Ton Thi Kim Loan, Bui Dinh Mao, and Eui-Nam Huh, "Low cost real-time system monitoring using Raspberry Pi," *2015 Seventh Int. Conf. Ubiquitous Futur. Networks*, pp. 857–859, 2015.
[22] K. Damodhar, B. Vanathi, and K. Shanmugam, "A Surveillance Robot For Real Time Monitoring And Capturing Controlled Using Android Mobile," *Middle-East J. Sci. Res.*, vol. 24, pp. 155–166, 2016.
[23] S. P. Richard, S. S. R.U.A, and N. M. E.I, "Aplikasi Pengenalan Wajah untuk Sistem Absensi Kelas Berbasis Raspberry Pi," J. Tek. Inform., vol. 15, no. 3, pp. 179–188, 2020.
[24] Y. Zhou, D. Liu, and T. Huang, "Survey of face detection on low-quality images," Proc. - 13th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2018, pp. 769–773, 2018.
[25] R. S. Deepthi and S. Sankaraiah, "Implementation of mobile platform using Qt and OpenCV for image processing applications," 2011 IEEE Conf. Open Syst. ICOS 2011, pp. 290–295, 2011.