Implementation of Face Recognition in People Monitoring Access In-and-Out of Crystal Dormitory Universitas Klabat

Penerapan Face Recognition Pada Pemantauan Orang Dalam Akses Masuk Dan Keluar Asrama Crystal Universitas Klabat

Jacquline M.S. Waworundeng*[1], Raycle Raynold Inzaghi Suwu[2]
[1,2]Jurusan Informatika, Fakultas Ilmu Komputer
Universitas Klabat, Jl. A. Mononutu, Airmadidi, Minahasa Utara, Sulawesi Utara, Indonesia
e-mail: *[1]jacquline.morlav@unklab.ac.id, [2]s11910054@student.unklab.ac.id

Abstract

Crystal Dormitory is a dormitory in Universitas Klabat that accommodates some male students. This study aims for monitoring and minimize the entry of foreigners into the dormitory which affects the comfort and safety of boarding students. Using the prototyping model, the systems can observe the faces of students who live in the dormitory, as well as foreigners who enter the dormitory. The system performed face recognition to recognize a person's face when the person's face data has been stored in the dataset. The hardware in this system used a Raspberry Pi 3 which is integrated with a webcam and monitor to detect and recognize human facial images. With machine learning libraries namely TensorFlow, OpenCV, Dlib, and Haar Cascade, combine with Python programming, the system can detect and recognize human facial images. If the system detects an unfamiliar human face, then the image of that person's face will be sent via the Telegram application using the auto-send feature to notify about the unidentified person.

Keywords—Machine Learning, OpenCV, Python, Raspberry Pi, Telegram.

Abstrak

Asrama Kristal merupakan asrama di kampus Universitas Klabat yang menampung beberapa mahasiswa laki-laki. Penelitian ini bertujuan untuk pemantauan dan meminimalisir masuknya orang asing ke dalam asrama yang berdampak pada kenyamanan dan keamanan santri asrama. Peneliti membuat sistem untuk mengamati wajah siswa yang tinggal di asrama, serta orang asing yang masuk ke asrama. Sistem melakukan pengenalan wajah untuk mengenali wajah seseorang ketika data wajah orang tersebut telah disimpan dalam dataset. Perangkat keras pada sistem ini menggunakan Raspberry Pi 3 yang terintegrasi dengan webcam dan monitor untuk mendeteksi dan mengenali citra wajah manusia. Dengan menggunakan library machine learning seperti Tensorflow, OpenCV, Dlib, Haar Cascade, dikombinasikan dengan pemrograman Python, sistem mampu mendeteksi dan mengenali gambar wajah manusia. Jika sistem mendeteksi wajah manusia yang tidak dikenal, maka gambar wajah orang tersebut dikirimkan melalui aplikasi Telegram dengan menggunakan fitur kirim otomatis yang memberikan pemberitahuan tentang orang tidak dikenal tersebut.

Kata kunci— Machine Learning, OpenCV, Python, Raspberry Pi, Telegram.

1. INTRODUCTION

The dormitory is a place to live and also a place to interact with fellow residents. It should be able to meet the needs of its residents, especially in terms of comfort, security, and privacy. Comfort and safety are the human rights of each dormitory resident who must be respected [1]. Not only the right to security in a psychological sense but also the right to security of property. The dormitory should not allow or only limit people who need to enter the dormitory, for reasons of security and convenience for the residents, so that unwanted things won't happen in the dormitory [2]. To minimize the occurrence of unwanted things in the dormitory which can disturb the comfort and cause anxiety for the residents, security in the dormitory needs to be improved [3]. By utilizing current technology that is more advanced, it is hoped that it can maintain order and security in the dormitory [4].

In this modern era, technology can do many things. Among them is technology that can recognize human faces. There are several related research of face recognition. For autonomous monitoring systems, research [5] combines face recognition and motion detection utilizing Raspberry Pi and the Internet of Things (IoT). Another study [6] on facial recognition uses biometrics and liveness detection to enable security systems to distinguish between fake and real identities when combining Raspberry Pi with a mobile device or tablet as the display device. In research [7], explore real-time face recognition using Raspberry Pi and libraries such as Dlib, face recognition libraries, and OpenCV library. Research [8], focused on smart city law enforcement services which implement face recognition using a tiny portable wireless camera mounted on a police officer's uniform that can capture a video stream, then passed to Raspberry Pi for face detection and recognition. Research [9], proposed real-time face recognition on video surveillance systems with, Python programming, Haar Cascade classifier, OpenCV library, and Raspbian OS integrated with hardware Raspberry Pi and Pi camera module. These related researches show the use of face recognition which consist of programming the hardware with Raspberry Pi, applied in various environment for monitoring and security system.

Before using face recognition, security systems in dormitories may have been limited to traditional methods such as direct monitoring by dormitory staff. While these methods can provide a basic level of security, they are vulnerable to abuse, theft or burglary, and possible access by unauthorized parties. In addition, monitoring individuals who enter or leave manually can also be time-consuming, and allows for mistakes or misidentification of dormitory occupants. Based on this problem, researchers conducted research through IoT and face recognition to recognize someone's identity that has been validated with the person's data. The system works with face recognition using a computer algorithm [10]. The system can recognize a person's face when the person's face data has been stored in a dataset.

Face recognition is a technology for scanning human faces using a webcam, which aims to record and monitor the activity recorded by the device. By using the webcam, the system can see the situation in our environment without having to observe it directly [11]. The system can see what is happening in the environment as long as the environment can be recorded by a webcam [12]. The scanned facial image will be processed in the OpenCV library and also Dlib. OpenCV and Dlib are Python libraries that focus on computer vision [13]. OpenCV itself was developed by Intel Corporation using the C and C++ programming languages [14]. With the help of the library from OpenCV, face recognition operations are more precise in detecting human faces [15]. Implemented with OpenCV, in this case, it requires a microprocessor, namely the Raspberry Pi. The Raspberry Pi 3 Model B specifications are suitable and compatible with image processing [16].

Webcams that are integrated with the Raspberry Pi microprocessor can run the face recognition feature as one of the methods found in OpenCV as a library [17]. Furthermore, Webcam can monitor people entering and leaving the dormitory. If the webcam detects a face that

is not in the dataset, then the image is sent to the head of the dormitory via the Telegram application.

2. LITERATURE STUDY

2.1 Internet of Things (IoT)

IoT refers to a type of network to connect anything with the Internet based on stipulated protocols through information sensing equipment to conduct information exchange and communications to achieve smart recognition, positioning, tracking, monitoring, and administration [18]. IoT is a revolutionary concept that connects physical objects around us via the internet network. IoT creates an ecosystem where various electronic devices, vehicles, sensors, and even household appliances can communicate with each other and share information automatically without human intervention. This concept involves the application of advanced technologies such as sensors, wireless devices, voice recognition systems, and artificial intelligence (AI) to enable the sending and receiving of data between objects connected to the internet network. In our research, the system was set up with Raspberry Pi which is connected to the internet and can send notifications through the Telegram application.

2.2 Raspberry Pi

Raspberry Pi is a microprocessor or single-board computer [19]. This tool is the same size as a credit card [20]. Despite its small size, Raspberry Pi can work like a functioning computer. Raspberry PI is equipped with a system-on-a-chip (SoC). Raspberry Pi uses an SD card for booting and data storage in storage. Raspberry Pi has many models and types. What distinguishes each model and type is the specifications, starting from the processor, network port, Bluetooth, wireless, number of USB ports, and others. The main operating system of the Raspberry PI uses Debian GNU/Linux and Python as the programming language. Raspberry Pi launched an operating system called Raspbian as a Linux distribution [21].

2.3 Face Recognition

Face recognition is a biometric system that can identify a person from digital images and live video (real-time) [22]. The way face recognition can identify and recognize people's faces, the method is with an algorithm that can see and take the special characteristics of an object specifically for human faces [23]. In the field of security systems, face recognition is one of the technologies that is often implemented. With face recognition, we can minimize direct physical contact with other people [24].

Face recognition technology cannot always provide accurate results. In face recognition, several problems and errors can be caused. Errors in recognizing a face have many factors, including changes in position, different lighting levels, and facial expressions. The evaluation parameters used for errors in face recognition are the False Acceptance Rate (FAR) and the Face Rejection Rate (FRR) [25].

The False Acceptance Rate (FAR) is an instrument to measure the possibility of an error occurring in a biometric system. FAR may experience errors in identifying and recognizing input images. Images that are outside the database, which are detected as images that are in the database are errors in recognizing the identity of the image. Conversely, another error is an image that is in the database, is not recognized and identified as an image outside the database [26]. A False Rejection Rate (FRR) is an error in the biometric system. Likewise, the False Acceptance Rate (FAR) can experience errors in identifying and recognizing input images, where images that are in the database that should be recognized become unrecognized [27].

2.4 Machine Learning

Machine learning is a part of artificial intelligence that is often used to solve several problems [28]. The goal of machine learning is to make hardware and software think, like the human brain. The application of machine learning in face recognition focuses on identifying human facial images. With the help of machine learning the system can identify human facial images automatically, especially people who access and enter the Crystal dormitory.

2.4.1 TensorFlow

TensorFlow is a machine learning library to train models developed by Google. TensorFlow is very helpful in developing models such as object tracking, object detection, and object recognition [29]. TensorFlow is used by developers to create deep learning projects at scale. With TensorFlow, the data obtained can be trained and classified easily and efficiently. TensorFlow can help and simplify the creation of neural networks that work the same as the human brain [30].

2.4.2 Open-Source Computer Vision Library *(OpenCV)*

OpenCV is a library of Python used for face recognition in real-time or directly. The performance of OpenCV itself is very fast and the computation is lighter. For monitoring used in dormitories, OpenCV is needed because the identified facial images are taken in real-time.

2.4.3 Dlib

Dlib is a library whose job is to solve machine learning-based problems. In developing artificial intelligence (AI)-based programs, Dlib is a library that must be used. The Dlib library uses the C++ programming language [31]. The benefit of using C++ implemented in the Dlib library is that the process of machine learning is very time-consuming. With the C++ programming language, coding execution is faster than other programming languages [32]. The main characteristic of the Dlib library is that it has a special purpose in analyzing and processing facial images using the facial landmark method [33]. With the help of facial landmarks, the process of recognizing a person's facial image is easier. Dlib is used in this study to anticipate unrecognizable facial images because some faces are closed or blocked. Dlib can recognize the image of a person's face even though part of the face is covered by recognizing every coordinate on the face.

2.4.4 Haar Cascade

Haar Cascade is a machine-learning algorithm to detect faces. The Haar Cascade algorithm is often used in object detection methods due to its advantages in fast computing [34]. In face recognition technology, real-time detection is needed so that the data information obtained is accurate and fast [35]. Haar Cascade is a method for detecting an object in real-time created by Paul Viola and Michael Jones in 2001 [36]. The Haar Cascade algorithm is very beneficial for deep learning technology, especially face recognition [37]. With the help of the Haar Cascade algorithm, faces can be identified accurately and quickly with the help of fast computations and depending on the number of pixels in an image [38].

The number of people who enter the dormitory at the same time can be more than one, therefore to process a person's facial image in real-time and quickly, the Haar Cascade algorithm is used so that the facial image capture process is precise and accurate.

2.5 Python

Python is a popular programming language in technology development related to machine learning. Python categorizes as a high-level programming language (easier for humans to understand). The reason for the popularity of Python itself so that many people use it is in terms of its ease of use [39]. Python's simple syntax and efficient data structure make it easier for

programmers to work on their coding. In addition to a simple syntax, the flexibility of Python itself can help work on and operate on almost all platforms: Mac, Linux, and Windows [40].

With increasingly advanced technological developments, deep learning and AI are becoming a new trend in the world of computer science [41]. Python is one of the programming languages used in matters related to machine learning, deep learning, and also in AI [42]. The Python language is used to call libraries such as OpenCV, TensorFlow, and Dlib. The library is used to process facial images contained in the dataset, which then go to the training stage.

2.6 Telegram

Telegram is a super-fast, simple, free, and secure messaging app [43]. With mobile and web-based messenger, Telegram provides end-to-end encryption of privacy services, where messages sent are guaranteed to be safe from data exploits.

The role of Telegram in this research is when the facial images ware processed do not match the facial images in the dataset, then the facial images are considered as foreigners entering the dormitory. Telegram account of the head of the dormitory receives a message from the system that there is a stranger face unrecognized had entered the dormitory.

3. RESEARCH METHOD

In this study, researchers developed face recognition with IoT using the prototype software development model. This method has stages such as Communication, Quick Plan and Modeling, Quick Design, Construction of Prototype, Deployment Delivery & Feedback [44]. The prototype method is a suitable method for developing prototype systems such as face recognition. The face recognition system is dynamic and always changes over time. By using the prototype method, a dynamic system that is more flexible and efficient is developed.

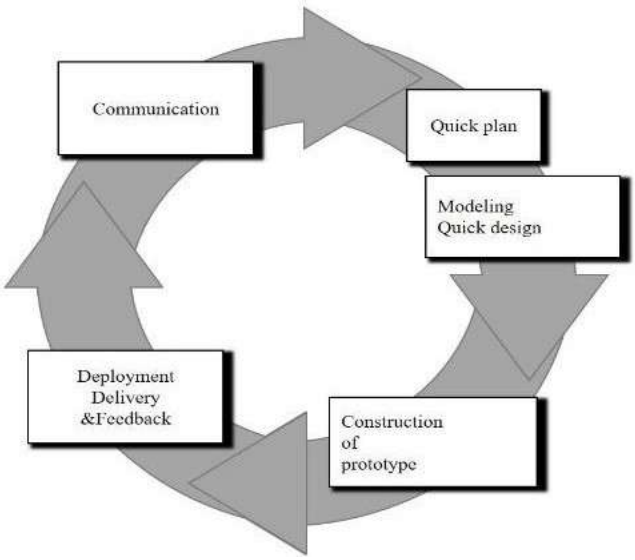


Figure 1 Prototype Model [45]

The stages of the prototype method are in Figure 1 which is applied to this study as follows.

1. Communications
   At this stage, the researcher conducted interviews with the head of the dormitory to obtain approval and information about the system being built and took pictures of facial images with students living in the crystal dormitory as a dataset.
2. Quick Plan and Modeling Quick Design

By using the requirements that have been obtained at the communication stage, the researcher makes a plan for each requirement. This will be the basic design of the prototype.

3. Construction of prototype
   Construction of the prototype is the prototype development stage by assembling each unit and tool component. After the tool assembly process is complete, then the program is created, which is used for the process of data collection and system development.
4. Deployment Delivery & Feedback
   At this stage, the prototype that has been made is tested by the system to test the functionality of the prototype made, whether the system meets the requirements, and whether the purpose of building the system is following the plan.

In this study, researchers used the Prototype method which is an iterative process, which works continuously. In the Prototype system development method, the required objectives of the research can be processed quickly, because the research object is directly involved. From the results of this design, testing, and evaluation will be carried out later, whether the system created is by the plan. The data used in this study is collected from students who live in dormitories by taking pictures in the form of facial photos. Subject shots must cover the entire face, and shots taken from the front using a webcam.

3. RESULTS AND DISCUSSION

This section describes the results and discusses the system design, system implementation, and system testing.

3.1 System Design

Figure 2 shows the system design of face recognition.

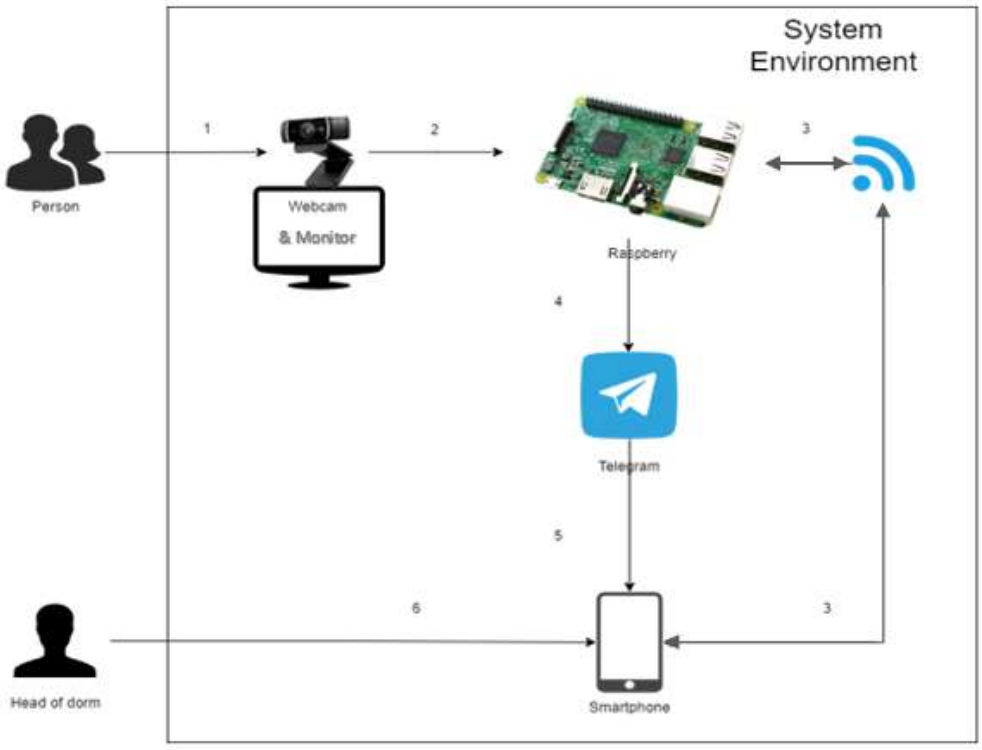


Figure 2 System Design

The flow is explained as follows.

1. The webcam that has been integrated with the raspberry cam detects the image of a person's face.
2. The results of the facial image taken by the webcam are processed on the Raspberry Pi.
3. Raspberry Pi needs internet connection to send push notifications to the Telegram application. The smartphone of the head of the dormitory connected to internet to receive data in his Telegram account.
4. If the system detects an unrecognized facial image, the facial image is captured and then sent via the telegram application using the auto-send feature of the telegram bot.
5. A picture message is sent via telegram to the head of the dormitory, as notification of unknown person were detected.
6. The head of the dormitory can see messages sent via Telegram.

Figure 3 shows the use case diagram of the system with the interaction of the user (admin). The use case diagram visualized the interaction of each process that occurs in the designed system. The use case diagram explains the flow and process between a person's facial image to a facial image sent via telegram in the development of a face recognition system.

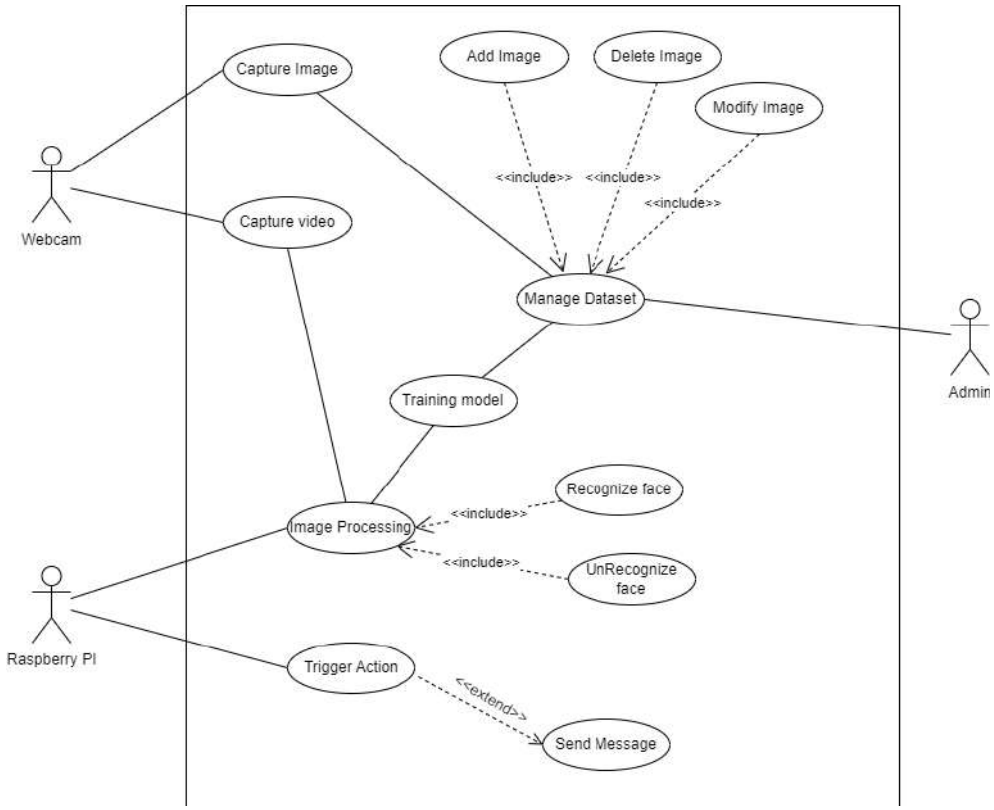


Figure 3 Use Case Diagram

3.2 System Implementation

The development stage refers to the process of creating program code using the Python language. The program code is made according to the purpose of this system so that the features designed in the system are following the needs and requirements of the user. The researcher executed the program code using a text editor from Raspbian (The Geany). The following are the stages of the programming process.

Figure 4 and Figure 5, show the command to install library OpenCV and Dlib in Raspberry Pi. Figure 6 shows the program code for how to create a dataset in the form of facial

image images using the OpenCV and dlib libraries on the Raspberry Pi. Generating the program code uses the haar cascade algorithm as an algorithm to identify facial images, which will later be successfully captured facial images will be entered into the ‘dataset’ directory. The images that are in the 'dataset' directory are then trained on each image.

```
sudo apt-get install python-opencv
```

Figure 4 OpenCV Library Installation

```
sudo apt-get install python3-dlib
```

Figure 5 Dlib Library Installation

```python
import cv2
import os

cap = cv2.VideoCapture(0, cv2.CAP_V4L2)

# Create a Haar cascade classifier for face detection
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Create a directory for the dataset if it doesn't exist
if not os.path.exists('dataset'):
    os.makedirs('dataset')

# Capture 10 images of faces and save them to the dataset directory
count = 0
while count < 100:
    ret, frame = cap.read()
    if ret:
        # Convert the frame to grayscale for face detection
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Detect faces in the grayscale frame using the face cascade classifier
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

        # If a face is detected, capture and save the detected face
        if len(faces) > 0:
            # Get the largest detected face
            (x, y, w, h) = max(faces, key=lambda f: f[2] * f[3])
            face = frame[y:y+h, x:x+w]
            cv2.imshow('Detected face', face)
            cv2.imwrite('dataset/raycle{}.jpg'.format(count+1), face)
            count += 1

        # Display the webcam feed with the detected faces
        cv2.imshow('Webcam', frame)
        if cv2.waitKey(1) == 27:  # Press Esc to exit
            break
    else:
        print("Error: Failed to capture frame")
        break

cap.release()
cv2.destroyAllWindows()
```

Figure 1 Dataset Program Code

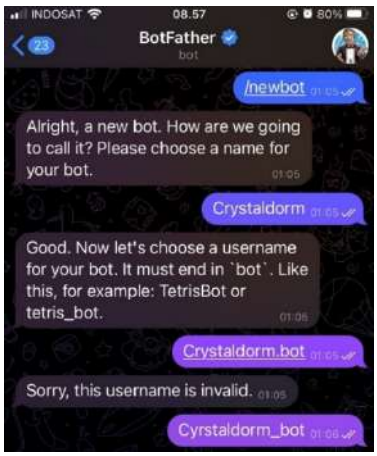


Figure 2 Creating a Telegram Bot

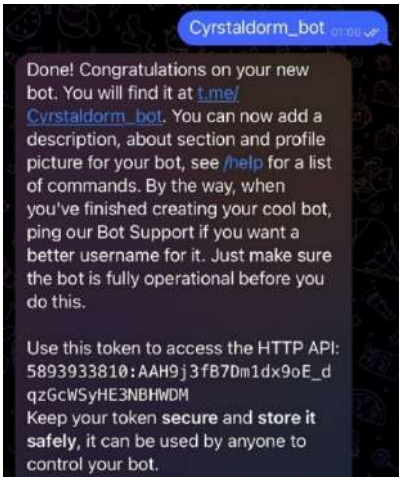


Figure 8 Token Telegram Bot

Figure 7 and Figure 8, show the stage for creating and configuring a telegram bot using BotFather. At first, create a new bot on BotFather, then the bot will be named crystaldorm_bot. After completing the bot creation stage, the BotFather will provide a token to access the API from the telegram bot. Furthermore, the token is used in the program code to perform the auto-send feature using the telegram bot as shown in Figure 9.

```
41  # Set up the Telegram bot
42  bot = telegram.Bot(token='6156244223:AAH3tY4GZZ9RZ-EhlAFvvp7wW4KoIZEpe30')
43  chat_id = '-941741798' # Replace with your chat ID
```
Figure 3 Telegram bot token and Chat Id

```
import cv2
import dlib
import os
import numpy as np
import telegram  # Import the Telegram module

# Load the detector and predictor
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

# Load the images from the dataset directory
dataset_dir = "dataset"
images = []
labels = []
for filename in os.listdir(dataset_dir):
    if filename.startswith("raycle") or filename.startswith("agung") or filename.startswith("jovan"):
        img = cv2.imread(os.path.join(dataset_dir, filename))
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = detector(img_gray, 1)
        for face in faces:
            landmarks = predictor(img_gray, face)
            face_img = img_gray[face.top():face.bottom(), face.left():face.right()]
            if face_img.size == 0:
                continue
            face_img_resized = cv2.resize(face_img, (100, 100))
            images.append(face_img_resized)
            if filename.startswith("raycle"):
                labels.append(0) # 0 represents "raycle" label
            elif filename.startswith("agung"):
                labels.append(1) # 1 represents "agung" label
            elif filename.startswith("jovan"):
                labels.append(2) # 2 represents "jovan" label

# Train the model
model = cv2.face.LBPHFaceRecognizer_create()
model.train(images, np.array(labels))

# Open the webcam
cap = cv2.VideoCapture(0, cv2.CAP_V4L2)

# Set up the Telegram bot
bot = telegram.Bot(token='6156244223:AAH3tY4GZZ9RZ-EhlAFvvp7wW4KoIZEpe30')
chat_id = '-941741798' # Replace with your chat ID

# Start the detection loop
async def main():
    while True:
        ret, frame = cap.read()
        if ret:
            img_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            faces = detector(img_gray, 1)
            for face in faces:
```
Figure 4 Face Recognition System Code

```
            landmarks = predictor(img_gray, face)
            x1, y1 = face.left(), face.top()
            x2, y2 = face.right(), face.bottom()
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            # Predict the label of the face
            face_img = img_gray[y1:y2, x1:x2]
            if face_img.size == 0:
                continue
            face_img_resized = cv2.resize(face_img, (100, 100))
            label, confidence = model.predict(face_img_resized)
            if confidence > 70:
                if label == 0:
                    cv2.putText(frame, "reycle", (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
                elif label == 1:
                    cv2.putText(frame, "agung", (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
                elif label == 2:
                    cv2.putText(frame, "unknown", (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0), 2)
                    # Send a message to the Telegram bot
                    unknown_img = cv2.imencode('.jpg', face_img_resized)[1].tobytes()
                    # Use an asyncio coroutine to send the photo message
                    async def send_telegram_message():
                        await bot.send_photo(chat_id=chat_id, photo=unknown_img, caption="Unknown detected!")
                    asyncio.run(send_telegram_message())

        # Show the resulting image
        cv2.imshow('Face Recognition', frame)
        # Stop the loop if the user presses the 'q' key
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

# Release the capture and destroy the windows
cap.release()
cv2.destroyAllWindows()
```

Figure 11 Face Recognition System Code (2)

```
            elif label == 2:
                cv2.putText(frame, "unknown", (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0), 2)
                # Send a message to the Telegram bot
                unknown_img = cv2.imencode('.jpg', face_img_resized)[1].tobytes()
                # Use an asyncio coroutine to send the photo message
                async def send_telegram_message():
                    await bot.send_photo(chat_id=chat_id, photo=unknown_img, caption="Unknown detected!")
                asyncio.run(send_telegram_message())
```

Figure 5 Auto Send Telegram Code

Figure 10 and Figure 11, show the program code for detecting and recognizing datasets in the form of facial image using the Dlib and OpenCV libraries on Raspberry Pi. The program code is intended so that the system can recognize faces whose facial images have been entered into the dataset and after that training is carried out on each image. If the detected face cannot be recognized by the system will be labeled 'unknown', then the image of the person's face will be captured and will be sent using the auto-send features (as in the code in Figure 12), to Telegram application through the Telegram bot.

3.3 Hardware Implementation

Hardware implementation is related to the hardware design process, from each existing unit and component. The following are the hardware components required for the system to work. Figure 13 show Raspberry Pi 3 model B which is a microprocessor that is used as a single-board computer to operate the designed system.

Figure 13 Raspberry Pi 3 Model B

Figure 14 Webcam

Figure 14 shows the webcam 1080p resolution as a device for taking pictures as a dataset and also for detecting and recognizing the image of a person's face. The quality of the camera could affect the detection process and 1080p resolution is capable to perform the face recognition. Figure 15 show Raspberry Pi, Webcam, and monitor the implementation of the face recognition system.

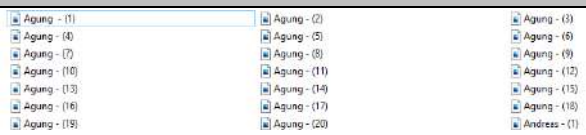
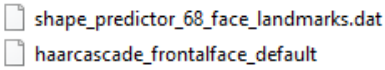
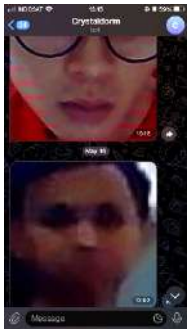
Figure 15 Hardware configuration in the face recognition system

3.4 System Testing

Testing is carried out by researchers to know whether the application is functioning properly or not. The main focus of this research as a whole is by implementing face recognition with Python libraries such as OpenCV and Dlib in the process starting from taking a person's facial image as a dataset, facial image training, face recognition, and also auto-sending it to the head of the dormitory using the telegram application using a webcam as the main focus tool to take an image of a person's face.

Based on the results of testing the system and its hardware, it was obtained that in sufficient light the system with the webcam 1080p resolution was able to detect faces with less than or maximum distance of 700 cm, which is considered a good range of detection. The system could not detect faces other than human faces. Table 1 show the results obtained from the final system testing.

Table 1 Final Testing of the system

System	Description	Validation	Results
Dataset	Taking 20 facial images as a dataset.		Success
Training	The system studies the images in the dataset with the existing training files.		Success
Face Detection and Recognition	Can detect a person's face and can also distinguish the image of a stranger's face that is not in the dataset.		Success
Face Recognition test with obstacle	Doing trials, on whether the system can recognize partially covered faces.		Success
Auto Send	An image of an unrecognized person's face is sent to the head of the dormitory using the Telegram application.		Success
Recognition and Detection of	The system cannot detect and recognize faces		Success

Object other than human	other than human faces		

4. CONCLUSION

This research builds a face recognition system using the Python programming language with several libraries namely *TensorFlow, OpenCV, Dlib,* and *Haar Cascade.* The face recognition process carried out consists of taking a dataset in the form of facial image images directly from a Raspberry Pi using a webcam and monitor which will also be used as a tool to detect facial images of people who enter the dormitory. Based on the testing and design that has been done, the results of this study can be concluded that the face recognition system built is capable of detecting and recognizing the faces of dormitory residents with sufficient accuracy, and can send messages automatically to the head of the Crystal dormitory when an unfamiliar face is detected. With this system, it is hoped that it can improve dormitory security and facilitate supervision of dormitory residents.

5. FUTURE RESEARCH

From the results of the research, there are suggestions for further system development considering the results of this study there are still deficiencies that can allow for further development. Here are some suggestions:

1. Using a microprocessor that has more adequate specifications such as Raspberry Pi 4, so that the face recognition process is more effective.
2. Using a webcam that has a higher specification as a tool for recording and monitoring, or using a Raspberry Pi cam.
3. For the design of this system, is not only done in the Crystal dormitory but can be used in all dormitories at Klabat University.

ACKNOWLEDGMENT

The author would like to thank Crystal Dormitory and Fakultas Ilmu Komputer Universitas Klabat, in support in this research.

REFERENCES

[1] Setiawati, A., 2020, "Perlindungan Hak Asasi Manusia Bagi Pengguna / Penghuni Bangunan Gedung (The Urgency on the Certificate of Occupancy as Human Rights Protection for Building User/Occupant)," *Teras Law Rev.*

[2] Munawwaroh, H., 2019, "Hak Keamanan Menurut Pasal 29-35 UU No. 39 tahun 1999 Perspektif Maqashid Syariah," *Ijtihad J. Huk. dan Ekon. Islam*, doi: 10.21111/ijtihad.v13i1.3230.

[3] Wulandari, R., 2016, "Analisa Kaitan Desain Asrama Dengan Perilaku Penghuni Melalui Studi Analisa Konten Penelitian Sejenis," Jurnal Idealog, *Desain Prod.*, 2016.
[4] Yang, Y., and Ye. L., 2021, "Dormitory Management System Based on Face Recognition," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, doi: 10.1007/978-3-030-69066-3_31.
[5] Balogh, Z., Magdin, M., Molnár, G., 2019, "Motion Detection and Face Recognition using Raspberry Pi, as a Part of, the Internet of Things", Acta Polytechnica Hungarica. 16. 2019-167. 10.12700/APH.16.3.2019.3.9. 2019. [Online]. Available: https://www.researchgate.net/publication/333773980_Motion_Detection_and_Face_Recognition_using_Raspberry_Pi_as_a_Part_of_the_Internet_of_Things
[6] Srihari, K., Ramesh, R., Udayakumar, E., Gaurav, D., 2020, "An Innovative Approach for Face Recognition Using Raspberry Pi."., Artificial Intelligence Evolution [Internet]. [Online]. Available from: https://ojs.wiserpub.com/index.php/AIE/article/view/62
[7] Ambre, S., Masurekar, M., Gaikwad, S., 2020, "Face Recognition Using Raspberry PI", In: Gunjan, V., Zurada, J., Raman, B., Gangadharan, G., (eds) Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough. Studies in Computational Intelligence, vol 885. Springer, Cham. [Online]. Available: https://doi.org/10.1007/978-3-030-38445-6_1
[8] Sajjad, M., et al., 2020, "Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities", Future Generation Computer Systems, Vol. 108, p. 995-1007. [Online]. Available: https://doi.org/10.1016/j.future.2017.11.013.
[9] Ahmad, A. H., et al., 2021, "Real time face recognition of video surveillance system using haar cascade classifier." Indonesian Journal of Electrical Engineering and Computer Science 21, p. 1389-1399.
[10] Prima, A.N., Prabowo, C., and Rasyidah., 2020, "Sistem Absensi dengan OpenCV Face Recognition dan Raspberry Pi," *JITSI J. Ilm. Teknol. Sist. Inf.*, doi: 10.30630/jitsi.1.2.12.
[11] Adriansyah, A., et.al., 2014, "Rancang bangun Dan Analisa CCTV Online Berbasis Raspberry Pi," *J. Sinerjgi*.
[12] Pratomo, I., Sakinah S., and Affandi, A. , 2018, "Evaluasi Jaringan Closed Circuit Television (CCTV) di Institut Teknologi Sepuluh Nopember (ITS)," *Appl. Technol. Comput. Sci. J.*, doi: 10.33086/atcsj.v1i2.858.
[13] Wibowo, A. P. W., 2016 "Implementasi Teknik Computer Vision Dengan Metode Colored Markers Trajectory Secara Real Time," *J. Tek. Inform.*
[14] Sobral, A., 2013,"BGSLibrary: An OpenCV C++ Background Subtraction Library," *IX Work. Visao Comput.*
[15] Kurniawan, L. M., 2015, "Metode Face Recognition untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang," *Sci. J. Informatics*, doi: 10.15294/sji.v1i2.4027.
[16] Hartono, H. V. E. A., et.al., 2020, "Perbandingan Performansi Mini PC 1.2 GHz dan 1.5 GHz sebagai Server Layanan Video Call menggunakan Codec H264," *J. Jartel J. Jar. Telekomun.*, doi: 10.33795/jartel.v10i2.93.
[17] Nagoriya. H., 2020, "Attendance System using Face Recognition utilizing OpenCV Image Processing Library," *Int. J. Res. Appl. Sci. Eng. Technol.*, doi: 10.22214/ijraset.2020.6297.
[18] Kumar, N., 2021, "Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges", Amazon Digital Services LLC.
[19] Barnes, R., 2015, "Raspberry Pi Projects Book," *Pearson Educ. India*. [Online]. Available: https://www.raspberrypi.org/magpi-issues/Projects_Book_v1.pdf.
[20] Jolles, J. W., 2021, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists," *Methods in Ecology and Evolution*. doi: 10.1111/2041-210X.13652.
[21] Budiawan, I., and Andriana, A., 2015,"Pengujian Pengenalan Wajah Menggunakan Raspberry Pi," *J. Otomasi Kontrol dan Instrumentasi*, doi: 10.5614/joki.2014.6.2.6.
[22] Huang, T., Xiong, Z., and Zhang, Z., 2011., *Face Recognition Applications*.
[23] A. Agrawal and S. Samson, 2016, "A Review on Feature Extraction Techniques and General Approach for Face Recognition," *Int. J. Comput. Appl. Technol. Res.*, doi:

10.7753/ijcatr0503.1008.
[24] Putra, S., Fitri, I., and Ningsih, S., 2021, "Absensi Pengenalan Wajah Menggunakan Menggunakan Algoritma Eigenface Berbasis Web," *J. Appl. Informatics Comput.*
[25] Rahman, J. Ul., Chen, Q., and Yang, Z., 2020, "Additive Parameter for Deep Face Recognition," *Commun. Math. Stat.* doi: 10.1007/s40304-019-00198-z.
[26] Kim, J. K., and Kim, Y. B., 2018 "Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates," doi: 10.21437/Interspeech.2018-1581.
[27] Setiawan, W., 2016, "Sistem Pengenalan Wajah Dengan Metode Face Features," *J. Ilm. SPEKTRUM*.
[28] Kubat, M., 2017,"An Introduction to Machine Learning," *An Introd. to Mach. Learn.*, pp. 1–348, 2017, doi: 10.1007/978-3-319-63913-0.
[29] Sutjiadi, R., and Pattiasina, T. J., 2020, "Deteksi Objek menggunakan Dashboard Camera untuk Sistem Peringatan Pencegah Kecelakaan pada Mobil," *J. Teknol. Inf. dan Ilmu Komput.*, doi: 10.25126/jtiik.2020712520.
[30] Arsal, M., Wardijono, B. Agus., and Anggraini, D., 2020, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *J. Nas. Teknol. dan Sist. Inf.*, doi: 10.25077/teknosi.v6i1.2020.55-63.
[31] Suwarno, S., and Kevin, K., 2020, "Analysis of Face Recognition Algorithm: Dlib and OpenCV," *J. INFORMATICS Telecommun. Eng.*, doi: 10.31289/jite.v4i1.3865.
[32] Mortensen, M., and Langtangen, H. P., 2016, "High performance Python for direct numerical simulations of turbulent flows," *Comput. Phys. Commun.*, doi: 10.1016/j.cpc.2016.02.005.
[33] Bachtiar, F. A., and Wafi, M., 2021., "Komparasi Metode Klasifikasi untuk Deteksi Ekspresi Wajah Dengan Fitur Facial Landmark," *J. Teknol. Inf. dan Ilmu Komput.*, doi: 10.25126/jtiik.2021834434.
[34] Kamarudin, N., *et al.*, 2019, "Implementation of haar cascade classifier and eye aspect ratio for driver drowsiness detection using Raspberry Pi," *Univers. J. Electr. Electron. Eng.*, doi: 10.13189/ujeee.2019.061609.
[35] Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C., 2019, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *IEEE Access*, doi: 10.1109/ACCESS.2019.2914929.
[36] Viola, P., and Jones, M., 2001 "Rapid object detection using a boosted cascade of simple features," doi: 10.1109/cvpr.2001.990517.
[37] Pratamasunu, G., et.al., 2020, "Pengenalan Wajah Mahasiswa Universitas Nurul Jadid pada Video Menggunakan Metode Haar Cascade dan Deep Learning," *Kecedasan Buatan, Komputasi dan Teknol. Inf.*
[38] Septyanto, M. W., et.al., 2020, "Aplikasi Presensi Pengenalan Wajah Dengan Menggunakan Algoritma Haar Cascade Classifier," *Telematika*, doi: 10.31315/telematika.v16i2.3182.
[39] Edwardo, T.O., 2018, "Penggunaan Python untuk Data Mining," *Sch. Comput. Sci. Binus Univ.*
[40] Raharjo, B., 2019, *Mudah Belajar Python Untuk Aplikasi Dekstop Dan Web Edisi Revisi*.
[41] Nurfita, R. D., and Ariyanto, G., 2018, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari," *Emit. J. Tek. Elektro*, doi: 10.23917/emitor.v18i01.6236.
[42] Akhinov, I. A., and Cahyono, M. R. A., 2021, "Pengembangan Smart Home System Berbasis Kecerdasan Buatan untuk Memanajemen Konsumsi Energi Rumah Tangga dengan Pendekatan Finansial," *JSAI (Journal Sci. Appl. Informatics)* doi: 10.36085/jsai.v4i1.1218.
[43] Telegram, Telegram FAQ, 2023. [Online]. Available: https://telegram.org/
[44] Pressman, R.S., and Maxim, B.R., 2019. Software Engineering: A Practitioner's Approach, New York, McGraw-Hill Education.