DEVELOPMENT OF AN INNOVATIVE SMART LOCK SYSTEM USING ARTIFICIAL INTELLIGENCE

Tran Thi Hien, Dao Thi Hang, Pham Van Phi
Nam Dinh University of Technology Education, Vietnam

ARTICLE INFORMATION

Journal: *Vinh University Journal of Science*
ISSN: *1859-2228*

Volume: 52
Issue: 4A
****Correspondence:***
vumanhquang04963@gmail.com

Received: *25 September 2023*
Accepted: *30 October 2023*
Published: *20 December 2023*

Citation:
Tran Thi Hien, Dao Thi Hang, Pham Van Phi (2023). Researching and developing a smart lock system applying artificial intelligence.
Vinh Uni. J. Sci.
Vol. 52 (4A), pp. 93-103
doi: 10.56824/vujs.2023a112

ABSTRACT

This paper focuses on researching and developing a smart security lock system using artificial intelligence, specifically facial recognition technology combined with the Internet of Things (IoT) to automatically recognize faces of accessing people.. The system is constructed with both hardware and software components, designed to verify the correct face stored in the database. The results show a high practical application potential, meeting the designed hardware and system programming objectives. The product can be used as an access control system based on facial recognition for government offices or industrial areas. Its advantages include cardless attendance tracking, fast and accurate verification, cost-effective installation, energy efficiency, compactness, and high efficiency.

Keywords: Artificial intelligence; face recognition; Raspberry Pi; smart lock; Internet of Things.

1. Introduction

Currently, the application of high technology in access control at organizations, companies, or industrial zones has become increasingly common. Integrated systems with artificial intelligence, computer vision, and advanced IoT technologies have proven to be very useful due to the advantages they offer. Instead of manual identification, which is time-consuming and labor-intensive, these new access control systems perform automatically with many new features and functions. The advantage of this system is that facial recognition does not require touching any hardware. Human faces are automatically detected using computer algorithms (Figure 1). Facial recognition is the first step of the facial recognition system.

Figure 1: *Facial Recognition Time Attendance Machine*

There are various methods for identifying individuals in the real world, such as using machine learning and deep learning. However, these methods have a significant drawback: they require the construction of large databases and the processing of vast amounts of data quickly and accurately, making the identification process time-consuming. The task at hand is to research and develop a program that uses an accurate identification method while requiring less computational time and data. To address this challenge, the authors propose a method that combines Viola-Jones face detection with Principal Components Analysis (PCA) feature extraction and Support Vector Machine (SVM) classification for face recognition [1-2]. PCA reduces unnecessary components, enabling fast and accurate computations. After PCA selects relevant features, SVM is used for classification and facial recognition. The research utilizes components of the Internet of Things (IoT), with the Raspberry Pi embedded computer serving as the primary control system. The Python programming language, along with several libraries, are employed to the fullest extent in building the facial recognition product [3-7].

Various facial recognition-based access control systems for tasks such as time attendance and personal identification have already been commercialized or are covered in contemporary research. These systems generally meet essential requirements, including accurate face recognition, comprehensive reporting, quick verification times, and user-friendly interfaces. However, they often come with downsides such as high costs, complex technology, high power consumption, and intricate designs that are challenging to maintain.

In this research, the authors aimed to construct and design a smart lock system that inherits the advantages of previously published solutions. Moreover, their product introduced a novel feature: remote door locking/unlocking through a web server. Furthermore, the finalized product is cost-effective, features straightforward installation technology, and satisfies the required functionalities. The performance of the system has been optimized through programming and design, facilitating easy replacement or maintenance when issues arise [8-11]. In this study, the authors successfully built a smart lock system based on Raspberry Pi, integrating the latest facial recognition techniques widely applied in artificial intelligence.

2. Hardware and software requirements

In this study, modern IoT devices are used in combination with some relevant programming libraries [2] that are widely used nowadays, including:

2.1. Raspberry Pi: is a very compact computer, which has integrated everything needed to be used as a computer (Figure 2). Its Broadcom BCM2835 SoC processor includes CPU, GPU, RAM, MicroSD Card slot, WiFi, Bluetooth, and 4 USB 2.0 ports. Raspberry Pi is a versatile device that can be used for electronics systems, computing setups, IoT projects, and more.

2.2. System components

a. Solenoid Lock LY-03: is an electrically operated door lock with the function of using a solenoid to lock and unlock (Figure 3). It is commonly used in smart homes, electronic cabinets, electric doors, and more. The lock operates on a 12/24VDC voltage and is typically a normally closed type known for its high quality and durability.

b. Module relay 5V with OPTO isolation for High/Low-Level trigger: This compact device includes OPTO and transistor isolation, enhancing safety when used with the main circuit board (Figure 4). The 1-Relay module with 5V High/Low-Level OPTO Isolation is utilized for switching high-power AC or DC electrical sources. The relay can be configured to activate on either a high or low-level trigger, depending on the jumper setting.

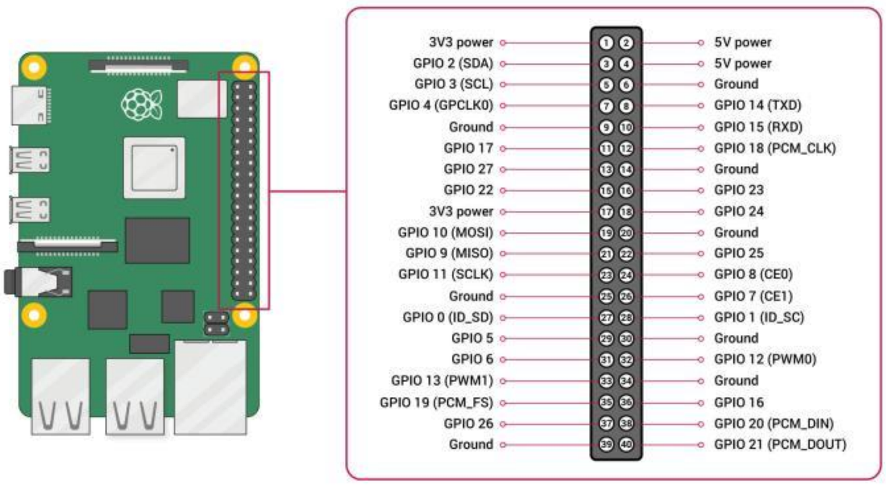

Figure 2. *The structural diagram of Raspberry Pi*

Figure 3: *Solenoid Lock LY-03*

Figure 4: *Module Relay 5V*

2.3. RPi.GPIO Library: This is a library package used to control the GPIO pins of the Raspberry Pi. This package provides a container with GPIO control functions. Additionally, RPi.GPIO supports the Python programming language and provides a module for controlling and pulsing all pins on the Raspberry Pi.

2.4. OpenCV Library: developed by Intel, is an open-source computer vision library with over 500 functions and more than 2500 optimized algorithms for image processing and related computer vision tasks. OpenCV is designed for optimal performance, utilizing the power of multi-core processors to perform real-time calculations, meaning its response speed is sufficient for typical applications. OpenCV is a cross-platform library, which means it can run on various operating systems, including Windows, Linux, Mac, iOS, and more.

2.5. Face_recognition Library: is a Python library. This is an extremely useful tool for small applications with accuracy that can go up to 99.38%. Although the recognition capabilities of face-recognition are limited by external conditions and depend on the image resolution, it is still the most powerful and useful library for face recognition projects.

2.6. Web Programming with Flask: Flask is a lightweight web application framework used for web development using the Python language. It is based on the Web Server Gateway Interface (WSGI), Toolkit, and Jinja2 Template Engine.

3. System analysis and design

3.1. The block diagram of the system

The block diagram of the system is as follows [3-8]:

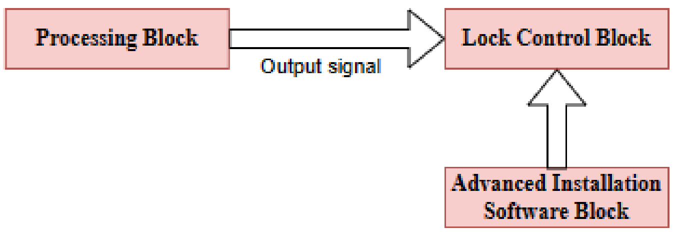

Figure 5: *Overall Block Diagram*

Detailed function of each block are as follows:

- **Processing Block:** Processes commands from the Face Recognition algorithm using Python programming language through the Raspberry Pi's CPU and generates a voltage signal on the GPIO pins to send to the next block.

- **Lock Control Block:** Receives input signals from the Processing block, and the Relay with opto-isolation circuit receives the signals and controls the lock through the NO pin of the component.

- **Advanced Installation Software Block:** Helps manage and control, customize according to the user's needs based on the application interface.

3.2. Methodology of Face recognition system

a. General layout: The face recognition system performs processing steps as described in Figure 6.

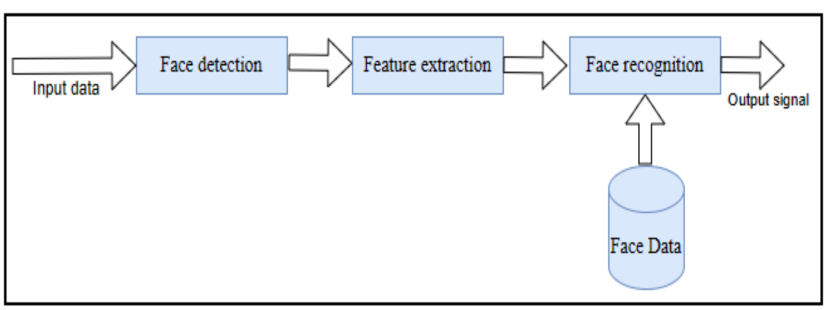

Figure 6: *Overall architecture of the face recognition system*

The project will have several directories as follows:

- ***dataset***: contains images of individuals (each person in a separate folder). When implemented in practice, it is necessary to add an ID for each person to control in case two or more people have the same name.

- ***build_dataset.py:*** used to create the dataset.

- ***encode_faces.py:*** encodes faces into 128-dimensional vectors.

- ***recognizer_faces_image.py:*** recognizes faces from a webcam video based on the dataset's encodings. Note that in this file, there is a part that checks for matches with faces in the dataset using the line:

matches = face_recognition.compare_faces(data["encodings"], encoding, 0.4.

The last parameter can be altered to better adapt if there are recognition errors (images not in the dataset being recognized as part of it).

- ***encodings.pickle:*** encodings generated from encode_faces.py will be saved to disk through this file.

b. Procedure to build the face recognition system

- Step 1: Create a dataset

Use *build_dataset.py* to construct the dataset. The dataset directory contains subdirectories for each person with names (and IDs if necessary), and each subdirectory contains images of the person's faces. Each image should contain only one face of that person (if there are multiple faces in an image, the implementation below will be more complex because it is required to to identify who is in the picture). The dataset is created using a webcam. Bring the person's face close to and away from the webcam, with different poses and expressions.

Run *build_dataset.py* and press the 'k' key to save images of each person. Each person should have at least 10-20 images for high accuracy. In addition to creating the dataset via the webcam, one can manually create the dataset or use Search APIs like Bing or Google. After creating the dataset with *build_dataset.py*, *run encode_faces.py* to create embeddings.

- Step 2: Create encodings for faces in the dataset

After creating the dataset, create embeddings (or encodings) for the faces in the dataset. The first thing to do is to extract face ROIs (avoid using the entire image as there may be a lot of background noise that affects the model's quality). Various methods can be used to detect and extract faces, such as Haar cascades, HOG + Linear SVM, deep learning-based face detectors, etc. Once you have face ROIs, pass them through an NN to obtain the encodings. In this case, we won't train the entire model to create encodings from scratch but rather reuse a pre-trained model (in the dlib library and integrated into face_recognition for ease of use) to generate face embeddings. Run encode_faces.py to save the encodings and names. The encodings and names are saved in the encodings.pickle file.

- Step 3: Face recognition in images

After obtaining encodings from datasets (obtained through a pre-trained model, via dlib and face_recognition). Use *recognize_faces_image.py* to recognize faces in images and *recognize_faces_video.py* for videos.

c. Connecting to Raspberry Pi: To use face recognition on CPU or embedded devices like Raspberry Pi, select the detection method as 'hog' in the *recognize_faces_image.py* file.

d. Building the Unlocking System on a Web interface: Building the unlocking feature on the web allows users to unlock the door from anywhere without having to recognize faces. The web interface is constructed using Flask and uses Authentication, requiring users to have an account to log in to the website. The website is designed to control the lock after user authentication. The structure includes login and registration pages, and a main interface page for control (Figure 7). To achieve this, web programming is required, specifically HTML and CSS, in addition to Python and Flask for authentication and control.

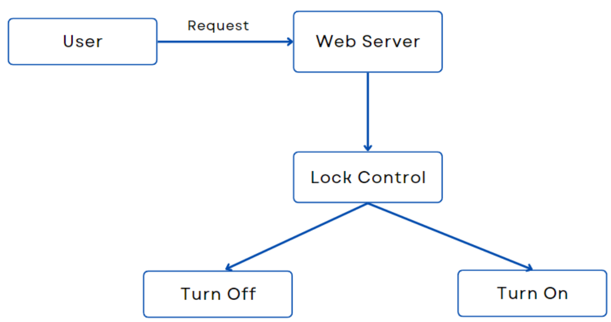

Figure 7: *Operation mechanism of the Web Server*

3.3. Hardware connection diagram

The hardware circuit connection diagram is shown in Figure 8.

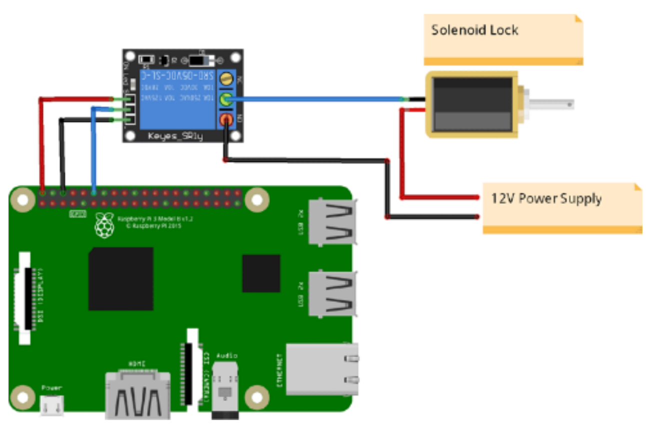

Figure 8: *System Circuit Connection Diagram*

Description of component connections: Power is supplied to the Relay and Solenoid Lock as described in the diagram. The DC+ port of the Relay is connected to pin number 2 of the Raspberry Pi for power supply, while the DC- port is connected to pin number 6 for ground. The IN port is connected to GPIO pin number 16, which is programmed in the Python code to transmit the signal of the recognized face. The actual connections are illustrated in Figure 9.

Figure 9: *Real device connection model*

3.4. Flowchart diagram algorithm

The hardware (Figure 10) and software (Figure 11) operation algorithm flowcharts of the system can be described as follows [9] - [20]:

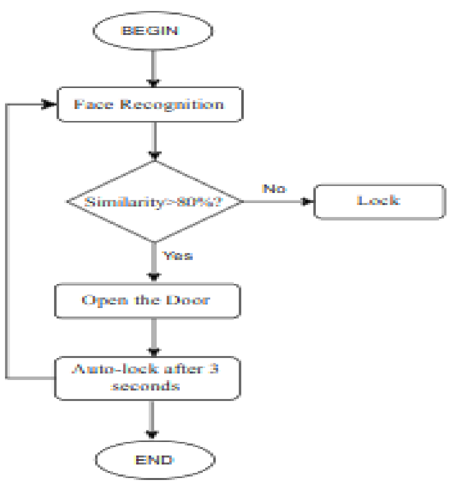

Figure 10: *Hardware algorithm flowchart*

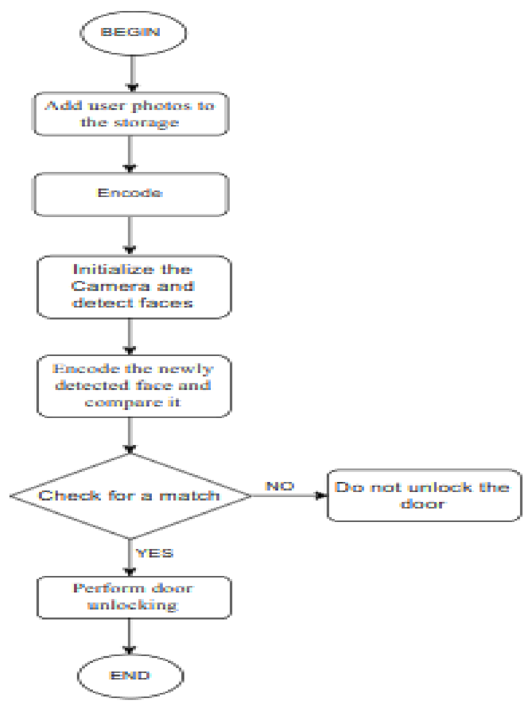

Figure 11: *Software algorithm flowchart*

4. Results and discussion

The designed applications have been run as shown in Figures 12, 13, 14, and 15:

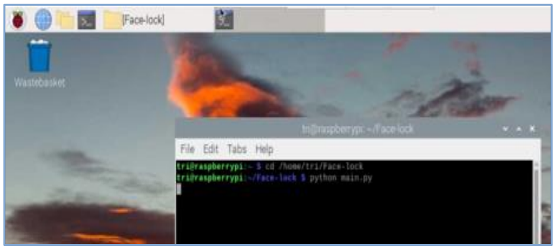

Figure 12: *Raspberry OS interface*

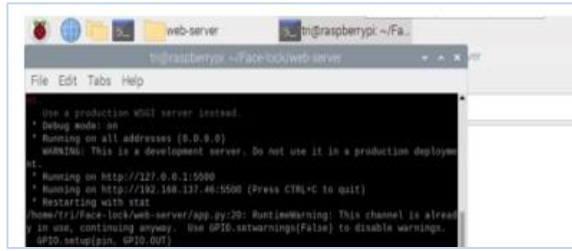

Figure 13: *Start the web server*

Figure 14: *Main interface of the website*

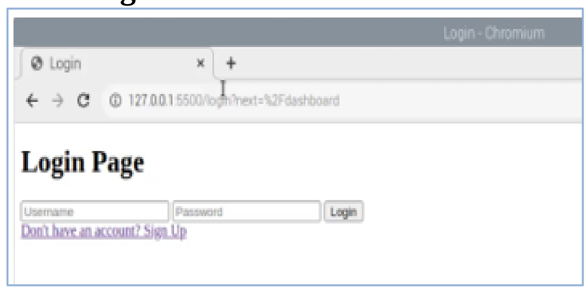

Figure 15: *Login page interface*

The results after execution are shown in Figure 16 and Figure 17:

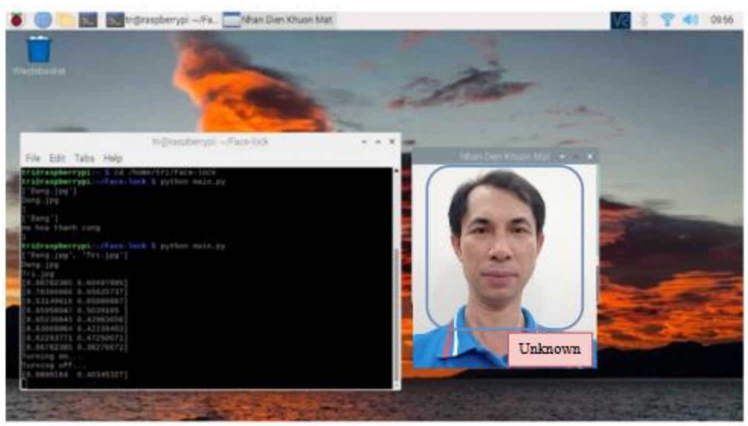

Figure 16: *The result returns "Unknown" when the face is not in the dataset*

Figure 17: *The result returns the "Name"*
if the person was already stored in the the dataset

In essence, the design of this research has met all the basic requirements set out. It correctly performs the unlocking process when a valid face is recognized. The successful development of a face recognition model on the Raspberry Pi has opened up many more practical applications, such as access control systems (timekeeping, attendance) based on face recognition for government agencies, companies, or industrial zones. This system exhibits many advantages in practice, including a compact design, high security, cost-effectiveness, easy installation and deployment, ease of maintenance, energy and manual labor savings. However, the recognition results of the system under specific conditions will vary due to various challenges such as processing speed, Raspberry Pi's memory, lighting conditions, face obstruction, image quality in real-world environments, etc. These are the issues that need to be addressed to effectively apply the system in practice.

5. Conclusion

The research has developed a complete face recognition system, including real-time face detection and identification with relatively high accuracy. The system performs well with fast detection times for images with straight-on facial shots and good image quality. This system can help managers monitor and look up information, update errors, and generate reports. However, there are still some limitations. Specifically, for individuals wearing glasses or backgrounds with many details, the detection is not accurate. For faces turned at steep angles or in low-light conditions, the program may not detect them. Future research could implement improved deep learning models to enhance the system's accuracy.

REFERENCES

[1] M. Zufar, "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time," *Jurnal Sains dan Seni ITS*, vol. 5, no. 2, pp. 2337-3520, 2016.

[2] K. Pulli, A. Baksheev, K. Kornyakov and V. Eruhimov, "Real-time computer vision with OpenCV," *Communications of the ACM*, vol. 55, no. 6, pp. 61-69, 2012. DOI: 10.1145/2184319.2184337

[3] S. H. Yoon and J. U. Kim, "IoT Open-Source and AI based Automatic Door Lock Access Control Solution," *International Journal of Internet, Broadcasting and Communication*, vol. 12, no. 2, pp. 8-14, 2020. http://dx.doi.org/10.7236/IJBC.2020.12.2.8

[4] H. H. Lwin, A. S. Khaing, and H. M. Tun, "Automatic Door Access System Using Face Recognition," *International Journal Of Scientific Technology Research*, vol. 4, no. 6, pp. 294-299, 2015.

[5] J. Mae, F. Rabbaanii, and R. B. Naza, "Door Access System Using Face Recognition," *4th International Conference on Eco Engineering Development 2020*, pp. 1-9, 2020. DOI: 10.1088/1755-1315/794/1/012127

[6] P. Kamencay, M. Benco, T. Mizdos and R. Radil, "A new method for face recognition using convolutional neural network," *Digital Image Processing and*

Computer Graphics, vol. 15, no. 4, pp. 663-672, 2017. DOI: 10.15598/aeee.v15i4.2389

[7] K. Venkatesh and B. Rajalingam, "IoT Based Home Automation Using Raspberry Pi," *Journal of Adv Research in Dynamical & Control Systems*, vol. 10, no. 07 - Special Issue, pp. 1721-1728, 2018.

[8] S. Mahindrakar and R. K. Biradar, "Internet of Things: Smart Home Automation System using Raspberry Pi," *International Journal of Science and Research (IJSR)*, vol. 6, issue 1, pp. 901-905, 2017. DOI: 10.21275/ART20164204

[9] M. Chauhan, M. Sakle, "Study & analysis of different face detection techniques," International Journal of Computer Science and Information Technologies, Vol. 5, no. 2, pp. 1615-1618, 2014.

[10] L. H. Hiep, "Research and building of the environmental monitoring system based on Android operating system", *TNU Journal of Science and Technology*, vol. 200, no. 07, pp. 125-132, 2019.

[11] H. Lwin, A. S. Khaing, and H. M. Tun, "Automatic Door Access System Using Face," *International Journal of Scientific & Technology Research*, vol. 4, issue 6, pp. 294- 299, 2015.

[12] A. Abraham and A. Usmani, "Face Recognition Based Attendance System," *OSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661*, ISSN: 2278-8727, volume 22, issue 1, Pp. 56 - 60, 2020.

[13] A. Singh and A. Gupta, "Automated attendance system with face recognition," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, issue 12, no. 2455-2143, pp 233-241, 2021. DOI: 10.33564/IJEAST.2021.v05i12.041

[14] L. H. Hiep and H. M. Viet, "Study to build an automatic measurement and warning system of alcohol concentration for vehicle drivers", *TNU Journal of Science and Technology*, vol. 225, no. 14, pp. 165-172, 2020. DOI: 10.1049/iet-smt.2019.0118

[15] M. F. Siddiqui and A. K. Jumani, "Face Detection and Recognition System forEnhancing Security Measures Using ArtificialIntelligence System," *Indian Journal of Science and Technology*, Vol. 13, No. 09, Pp. 1057-1064, 2020. DOI: 10.17485/ijst/2020/v013i09/149298

[16] L. H. Hiep and H. M. Viet, "Research on designing color-based product classification system applying digital image processing technology", *TNU Journal of Science and Technology*, vol. 226, no. 11, pp. 332-340, 2021.

[17] L. H. Hiep and H. M. Viet, "Study to build a control and monitoring system of poultry incubator based on Internet of Things", *TNU Journal of Science and Technology*, vol. 227, no. 08, pp. 20 - 28, 2022.

[18] L. H. Hiep, "Research and building a voice-controlled environmental monitoring device using Internet of Things", *Vinh University Journal of Science (VUJS)*, vol. 51, no. 2A, pp. 13-23, 2022. DOI: 10.56824/vujs.2022nt07

[19] P. Chakraborty and S. Rahman, "Automatic Student Attendance System using Face Recognition," *International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 - 8958*, volume 9 issue 3, pp. 93-99, 02/2020.

[20] L. H. Hiep, "Study to design of intelligent lighting control and monitoring systems", *TNU Journal of Science and Technology*, vol. 189, no. 13, pp. 99-105, 2018.

TÓM TẮT

NGHIÊN CỨU CẢI TIẾN HỆ THỐNG KHÓA THÔNG MINH ỨNG DỤNG TRÍ TUỆ NHÂN TẠO

Trần Thị Hiền, Đào Thị Hằng, Phạm Văn Phi
Trường Đại học sư phạm kỹ thuật Nam Định, Việt Nam
Ngày nhận bài 25/9/2023, ngày nhận đăng 30/10/2023

Bài báo trình bày kết quả nghiên cứu, xây dựng một hệ thống khóa bảo vệ thông minh ứng dụng trí tuệ nhân tạo, cụ thể là kỹ thuật nhận dạng khuôn mặt người kết hợp công nghệ mạng kết nối vạn vật thực hiện tự động việc xác thực khuôn mặt. Hệ thống được xây dựng gồm phần cứng và phần mềm với chức năng thực hiện việc xác minh đúng khuôn mặt đã được lưu lại trong cơ sở dữ liệu. Kết quả nghiên cứu cho thấy khả năng ứng dụng vào thực tế rất cao, thực hiện đúng các yêu cầu, mục tiêu như đã thiết kế phần cứng và lập trình hệ thống. Sản phẩm có thể được sử dụng như một hệ thống kiểm soát người vào ra dựa trên việc nhận dạng khuôn mặt dùng cho các cơ quan công sở hoặc tại các khu công nghiệp. Ưu điểm của hệ thống là thực hiện chấm công không cần dùng thẻ, thời gian xác thực nhanh chóng, chính xác, giá thành lắp đặt rẻ và dễ dàng, tiết kiệm điện năng, nhỏ gọn và hiệu quả cao.

Từ khóa: Trí tuệ nhân tạo; nhận diện khuôn mặt; Raspberry Pi; khóa thông minh; mạng kết nối vạn vật.