Journal of Intelligent & Fuzzy Systems 40 (2021) 8021–8031
DOI:10.3233/JIFS-189624
IOS Press



# Face recognition in an intelligent door lock with ResNet model based on deep learning

Suphawimon Phawinee[a], Jing-Fang Cai[b], Zhe-Yu Guo[b], Hao-Ze Zheng[b] and Guan-Chen Chen[b,*]
[a]*Department of Computer Science, Kasetsart University, Bangkok, Thailand*
[b]*Department of Mechanical and Electro-mechanical Engineering Tamkang University, Taiwan (ROC)*



**Abstract**. Internet of Things is considerably increasing the levels of convenience at homes. The smart door lock is an entry product for smart homes. This work used Raspberry Pi, because of its low cost, as the main control board to apply face recognition technology to a door lock. The installation of the control sensing module with the GPIO expansion function of Raspberry Pi also improved the antitheft mechanism of the door lock. For ease of use, a mobile application (hereafter, app) was developed for users to upload their face images for processing. The app sends the images to Firebase and then the program downloads the images and captures the face as a training set. The face detection system was designed on the basis of machine learning and equipped with a Haar built-in OpenCV graphics recognition program. The system used four training methods: convolutional neural network, VGG-16, VGG-19, and ResNet50. After the training process, the program could recognize the user's face to open the door lock. A prototype was constructed that could control the door lock and the antitheft system and stream real-time images from the camera to the app.

Keywords: Face recognition, intelligent lock, ResNet, deep learning

## 1. Motivation

Life in modern society is increasingly affected by Internet of Things (IoT) and artificial intelligence. Although technology has advanced, numerous problems still exist. Numerous household appliances are now designed with IoT in mind. However, most home door locks have retained the conventional mechanical system. Conventional door locks provide excellent security but lack convenience because a key is required for access. Hence, we explored improvements to this type of door lock. In addition to using a key, biological characteristic recognition (both contact and noncontact) can also be applied. Contact recognition is widely applied in a variety of products. Its limitation is that users must approach the machine to be detected, which is inconvenient. Noncontact recognition includes sound, face, and iris recognition, among which face recognition is the first choice based on cost and reliability.

Improving the door lock not only involves changing its mechanism but also raising its value by using IoT. Therefore, this project focused on the combination of IoT and face recognition on a door lock. Face recognition should be used in the smart home. Access is controlled wirelessly through a mobile application (hereafter, app), and the camera module can sense knocking or damage. The app informs the user and finally records the situation.

## 2. Literature review and discussion

### *2.1. Door lock*

Some developments in smart door locks include the One Password password manager authentication

*Corresponding author. Guan-Chen Chen, Department of Mechanical and Electro-mechanical Engineering Tamkang University, Taiwan, ROC. E-mail: gcchen@mail.tku.edu.tw.

ISSN 1064-1246/$35.00 

Fig. 1. KF-320 face recognition lock.

remote key, mobile phone fingerprint identification and AES encryption, allowing users to verify their identity through fingerprints and mobile devices, and server-end authentication. Another example is the application of active remote keyless control and algorithms for encryption to achieve high-security certification [1]. Therefore, this project selected several commercially available smart door locks for comparison.

The KF-320 face recognition lock is depicted in Fig. 1. Unlocking is achieved through face recognition, swiping, or using a key. The recognition distance is 30–70 cm. This lock is suitable for wooden doors, stainless steel security doors, iron doors, and pure copper doors.

The ASUS smart door lock (Fig. 2) can be unlocked with a password, near-field communication, remotely, and a key. The device is used with a door handle; therefore, the user must first confirm whether the door has a two bolt hole. When fewer than 200 door locking–unlocking cycles remain, a weak current display reminds the user to replace the battery. This lock has an antitheft system: an alarm will sound when someone attempts to open the door without authorization.

The August smart door lock is depicted in Fig. 3. This lock can only be unlocked with the app, and visitors can enter or exit the home on a specific date and at a specific location. With the August app and the phone's Bluetooth activated, even without taking out the phone, the door will automatically unlock when it finds the phone is nearby and automatically lock at the set time. This lock is suitable for general latch-type sub locks. Moreover, it can also be equipped with a camera doorbell and a smart keyboard to ensure safety.


Fig. 2. ASUS smart door lock.

The EF-15 face recognition smart lock, developed by Tairong Intelligent Technology (Fig. 4), can be unlocked with face recognition, a password, a magnetic card, and a key. It locks immediately after the door closes and is fireproof and waterproof. The triangular geometry enables high-precision analytical recognition; therefore, it is not affected by makeup, and even twins can be distinguished. Infrared LED lights next to the lens allow the device to function in the dark.

Most current smart door locks combine a variety of unlocking methods. Although most of the door locks have antitheft devices, the function becomes invalid if the lock is destroyed. Therefore, the current project proposes the placement of a vibration module on the smart door lock. When it is forcibly destroyed, the vibration module will be activated, which triggers the buzzer and notifies the user through the app.

Fig. 3. August smart door lock.

Fig. 4. EF-15 face recognition smart lock.

## 2.2. Facial detection methods

With the advent of the IoT era, facial detection and recognition techniques are becoming increasingly critical. Higher speed and accuracy have become imperative. A wide array of facial detection methods exist, with each presenting advantages and disadvantages. Different detection methods have been analyzed and attempts made to improve them. The most prominent methods are compared next.

### 2.2.1. Haar cascade in OpenCV

Since the launch of Viola and Jones in 2001, the facial detector based on the Haar cascade has been the most advanced technology in the field [2].

Its advantages are as follows:

- It works almost in real time on the CPU.
- It employs simple architecture.
- It detects different proportions of faces.

Its disadvantages are as follows:

- The high probability of false predictions (the main disadvantage).
- It is unsuitable for nonfrontal images.
- It does not work under occlusion.

### 2.2.2. Deep neural networks in OpenCV

This model is included in version 3.3 of OpenCV. It is based on the single shot multibox detector, and the ResNet-10 architecture is used as the backbone to train this model using images from the network.

It has the following advantages:

- It is the most accurate method of all.
- It runs in real time on the CPU.
- It is compatible with different facial directions, such as top, bottom, left, right, and side.
- It is able to work under severe occlusion.
- It detects faces of various sizes

### 2.2.3. Histogram of oriented gradients face detector in Dlib

This is a widely used face detection model based on histogram of oriented gradients (HOG) features and support vector machine. This model consists of five HOG filters: front, left, right, front–left, and front–right.

This model has the following advantages:

- It has the fastest CPU.
- It is suitable for front and slightly nonfrontal faces.
- It is a more lightweight model compared with others.
- It works under small occlusions.

This model has the following disadvantages:

- Small faces cannot be detected because the minimum face size for training is $80 \times 80$ (the main

drawback). Therefore, users must ensure that their face size is larger than that in the app.
– The bounding box typically excludes a part of the forehead and even a part of the chin.
– It does not work well under severe occlusion.
– It is unsuitable for side and extreme nonfrontal views, such as looking up or looking down.

#### 2.2.4. *Convolutional neural network in Dlib*

This method uses the Maximum Margin Object Detector with the functionality of a convolutional neural network (CNN). The training process for this method is straightforward and little data are required to train the custom object detector.

This method has the following advantages:

– It is suitable for different facial directions.
– It can adapt to occlusion.
– It works fast on the GPU.
– The training process is straightforward.

This method has the following disadvantages:

– The CPU speed is slow.
– Small faces are not detected because the minimum face size for training is 80 × 80. Therefore, users must ensure that the face size is larger than the face size in the app. However, you can train your face detector with a smaller face size.
– The bounding box is even smaller than the HOG detector.

Deep neural network (DNN)-based detectors overcome all the shortcomings of Haar-based detectors, as presented in Fig. 5, without affecting any of the advantages offered by Haar. Therefore, DNN-based face detectors are the first choice for OpenCV. Although they are slower than the Dlib HOG-based face detector discussed previously (Fig. 6), for home security, accuracy takes precedence over speed; therefore, DNNs in OpenCV was selected for the current project.

### 2.3. *Analysis of the multiple algorithms of OpenCV in face recognition and applicability of neural networks*

OpenCV provides various algorithms for identification, including built-in Eigenfaces, Fisherface, and local binary pattern (LBP) histogram [3].

Currently, CNNs are more common in deep learning, and VGG-16, VGG-19, and ResNet are image classification models of ImageNet used to compare and analyze.

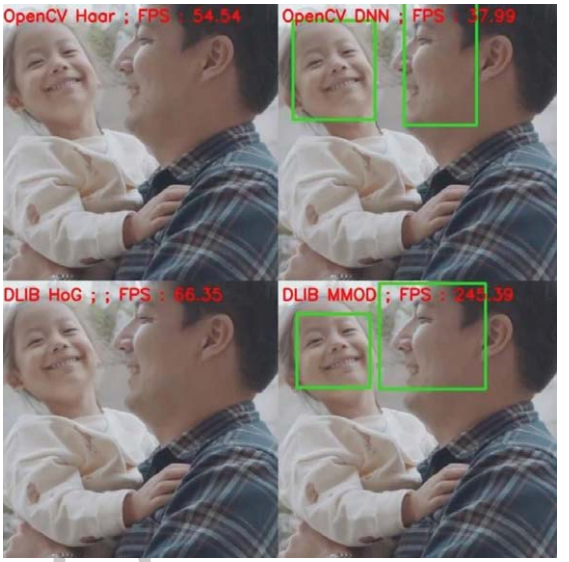



Fig. 5. DNN detection technology.

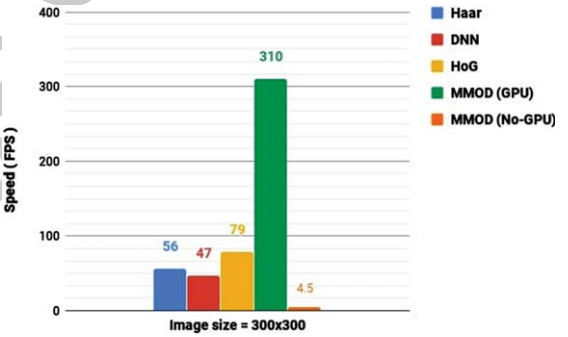



Fig. 6. Detection technology speed comparison.

#### 2.3.1. *Introduction to OpenCV built-in algorithms*

*2.3.1.1. Eigenfaces.* EigenFaces is a statistical feature-based method that approaches face images as random vectors and uses statistical methods to identify different facial feature patterns. The workflow is illustrated in Fig. 7. Eigenfaces identifies the basic elements of the face image distribution from a statistical point of view (i.e., the feature vector of the covariance matrix of the face image sample set) to approximate the face image. These feature vectors are called feature face.

Feature face is superior to other methods in terms of efficiency because feature face is calculated fast, and a numerous faces can be processed in a short time. However, a problem exists in the actual

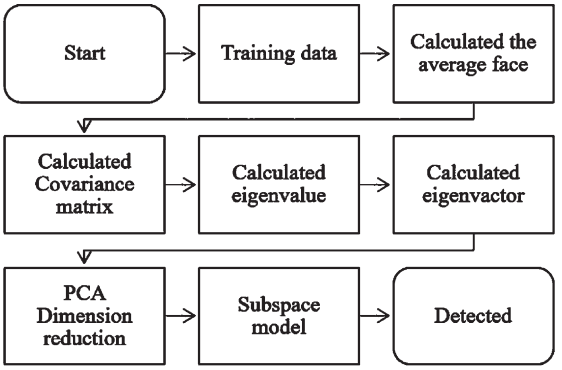



Fig. 7. Eigenfaces workflow.

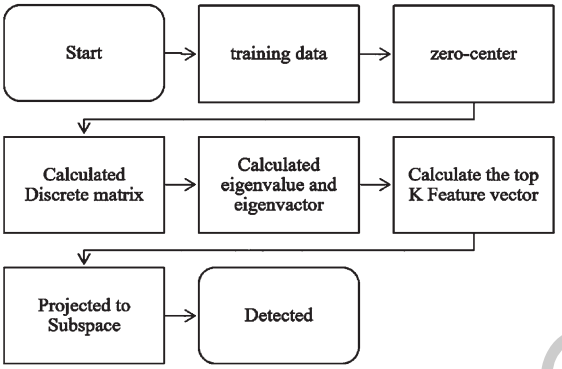



Fig. 8. Fisherface workflow.

use of feature face: in different lighting conditions and imaging angles, the recognition rate is greatly reduced. Therefore, the use of feature faces requires user identification using a frontal image under uniform lighting conditions.

*2.3.1.2. Fisherface.* Fisherface is a linear discriminant analysis (LDA)-based face recognition algorithm. The workflow is illustrated in Fig. 8. Both LDA and principal component analysis (PCA) are used to classify the original data after dimension reduction [4]. PCA is used to remove the original data set. The redundant dimension makes the variance of each dimension of the projection subspace as large as possible (i.e., the entropy is as large as possible). LDA finds those discriminative dimensions through data dimensionality reduction to ensure the projection of raw data in these dimensions is as distinct as possible from different categories.

*2.3.1.3. LBP.* LBP focuses on each pixel, judges the relationship between the gray value of the surrounding pixels, and performs binary encoding to obtain the LBP encoded image of the whole image, and then divides the LBP image into gradx × grady regions. The LBP coded histogram of each region is first obtained and then the LBP coded histogram of the entire image is obtained [5]. The workflow is presented in Fig. 9.

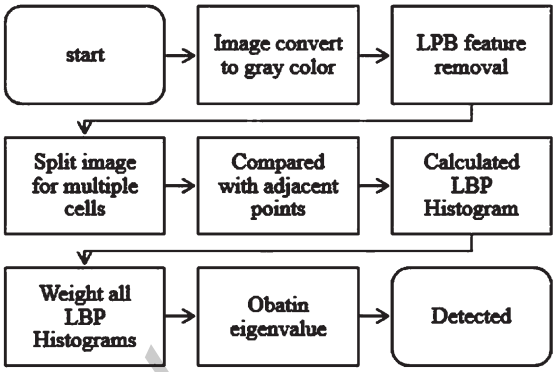



Fig. 9. LBP workflow.

The purpose of face recognition is achieved by comparing LBP coded histograms of different face images. The original LBP operator is defined as a threshold value in the window of 3 × 3, and the gray value of the adjacent 8 pixels is compared with the pixel value of the adjacent 8 pixels. If the surrounding pixel value is greater than or equal to the central pixel value, then the position of the point is marked as 1, otherwise it is 0. Thus, the 8 points in the 3 × 3 neighborhood are compared to produce an 8-bit binary number. That is, the LBP value of the center pixel of the window is obtained, and this value is used to reflect the texture feature of the area.

### 2.3.2. *Comparison and analysis of CNN and VGG-16, VGG-19, and ResNet, including in ImageNet*

*2.3.2.1. CNN.* CNN is a feedforward neural network whose artificial neurons can respond to a part of the surrounding cells in coverage and exhibit excellent performance for large image processing. A CNN consists of one or more convolutional layers and a fully connected layer at the top, as well as associated weights and pooling layers. This structure enables the CNN to exploit the two-dimensional structure of the input data [6]. In addition, CNNs require fewer parameters than other deep learning structures, as indicated in Fig. 10.

The convolutional layers extract the relevant features from the image, and the fully connected layers

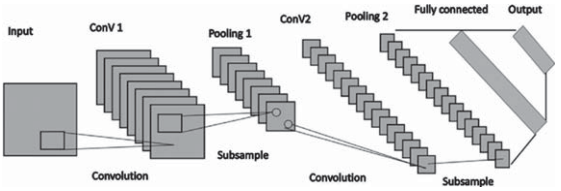



Fig. 10. CNN structure.

will focus on using these features to classify the images well [7]. The current project will use four convolution layers and two fully connected layers.

*2.3.2.2. Image classification model in ImageNet. 2.3.2.2.1. VGG-16 and VGG-19:* VGG is the abbreviation for Visual Geometry Group of Oxford University, UK. It enables the use of more hidden layers and the use of numerous picture in training, and it improves accuracy to 90%.

VGG-16 has 16 layers, including 13 convolutional layers and 3 fully connected layers, whereas VGG-19 has 19 layers, including 16 convolutional layers and 3 fully connected layers. The structure is illustrated in Fig. 11. The first few layers only use a $3 \times 3$ convolution kernel to increase network depth, and the number of neurons per layer is in turn reduced by max pooling. The last three layers are two fully connected layers with 4,096 neurons and one softmax layer. The "16" and "19" indicate the numbers of network layers in the network that must be updated with weight.

*2.3.2.2.2 ResNet:* Because of the convergence of deep networks, ResNet differs from traditional sequential network architectures, such as VGG, in that it incorporates an identity mapping layer ($y = x$ layer), which allows the network to increase depth without degrading. Figure 12 illustrated a building block. When the input goes through two weight layers and adds to the input to form a microarchitecture module, then ResNet is ultimately composed of several microarchitecture modules.

The problem of face recognition, be it face verification or face identification, is that it must remove the feature value with "discrimination" on the person's face. The capture of facial features values is the core concern. Considering the introductory information, this project used CNN and VGG-16, VGG-19, and ResNet in ImageNet to compare and analyze.

The advantages of deep learning include its use of unsupervised or semisupervised feature learning and layering. The efficient feature extraction algorithm replaces the manual acquisition feature. This method can achieve a higher recognition rate, and the

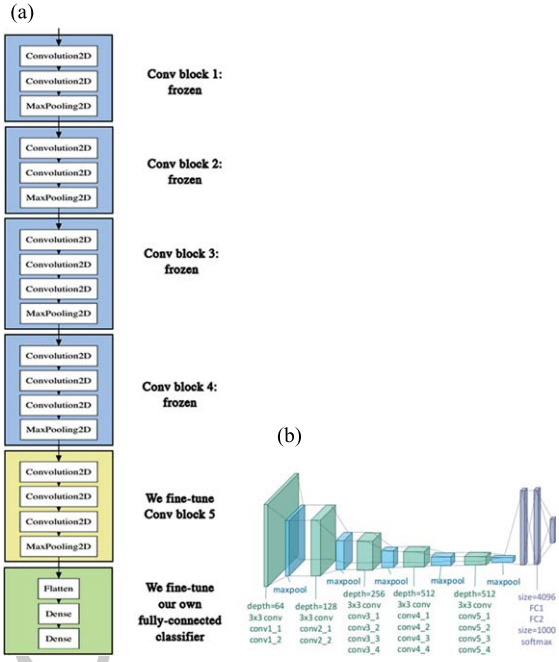



Fig. 11. VGG structure [8]. (a) VGG16 structure diagram. (b) VGG19 structure diagram.

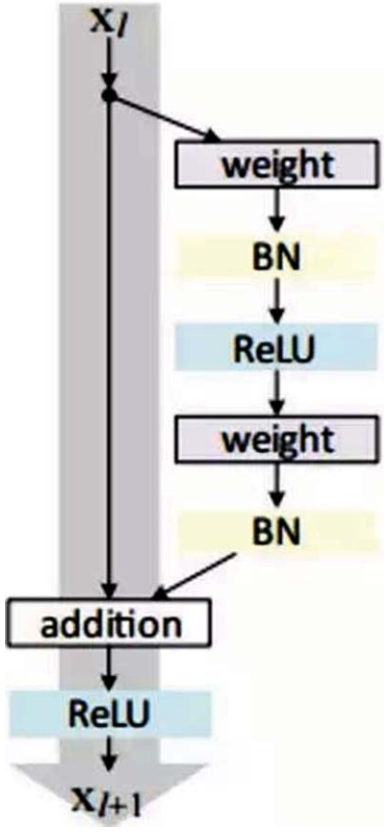



Fig. 12. ResNet's residual network.

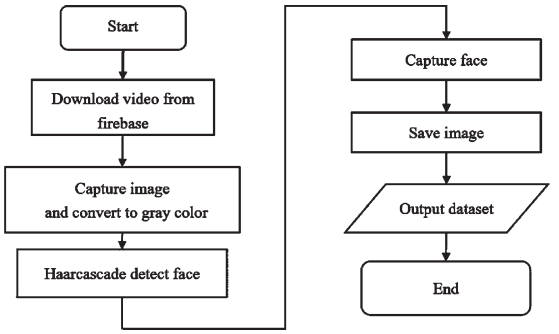



Fig. 13. Flow of database creation.

CNN module in ImageNet has a strong generalization ability, which can greatly reduce training time.

## 3. Research methods and steps

### *3.1. Face recognition*

#### *3.1.1. Face detection and building the dataset*

In terms of face detection, this project used Haar cascade in OpenCV. Although a literature review suggested that Dlib has more advantages than Haar cascade, with Dlib it is extremely difficult to cut face images during execution. Dlib requires the setting of more variables than Haar cascade to crop out recognizable faces to construct a database [9].

An established database was used as a training set. This project used training models, such as CNN, VGG-16, VGG-19, and ResNet, each model required different inputs. Because Dlib requires more input than Haar cascade, this project opted to use Haar cascade.

The process of establishing the database is illustrated in Fig. 13. First, facial videos were downloaded from Firebase. Each video was divided into 100 image pieces that were then separated into the training folder and validation folder and converted to grayscale. Haar cascade was used to detect the face. The final storage process and the establishment of the database are revealed in Fig. 14.

#### *3.1.2. Training model*

The training process (Fig. 15) used CNN modules including VGG-16, VGG-19, and ResNet in ImageNet.

First, the established database was loaded and the data generation program for training and validation was set up. That was followed by the building

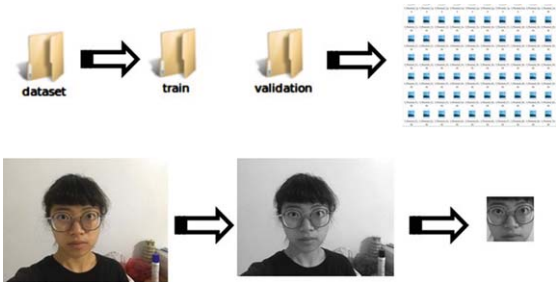



Fig. 14. Folder creation and image processing.

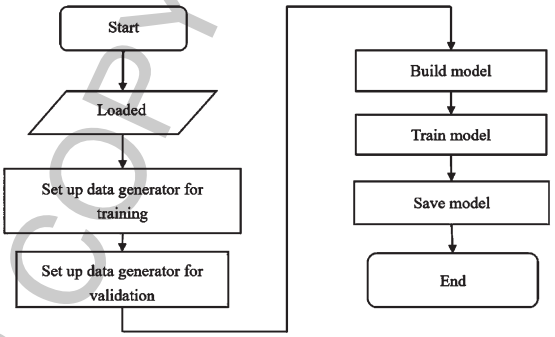



Fig. 15. Model training flow.

of the pretraining model and finally training and storage.

#### *3.1.3. Training result and testing*

The execution of VGG-16 and VGG-19 did not work as envisioned because the project was executed on Raspberry Pi. Raspberry Pi 3 $B+$ model has a capacity of only 1 GB but VGG-16 and VGG-19 have a capacity of 500MB that is a large network and hungry for memory. It will need more than 1GB of RAM to hold all the weights.; therefore, they could not run. In terms of the ResNet model, ResNet50 was chosen due to capacity constraints(99MB) [10].

The data display used Keras' built-in finite module "fit" to compare the recognition accuracy of CNN, VGG-16, VGG-19, and ResNet50. The results are listed in Table 1.

The testing used 20 epochs for the training model. The accuracy of the model was calculated through the training process at the last epoch. The difference in prediction accuracy was calculated through the face recognition process; it relates to the approximate difference in accuracy between the recognized person and others in the training set.

The best model was selected by identifying the highest difference in prediction accuracy because a

Table 1
Comparison of CNN face recognition modules

| | Accuracy of model | Difference in prediction accuracy |
|---|---|---|
| CNN | 100% | 40% |
| VGG-16 | RAM overload | RAM overload |
| VGG-19 | RAM overload | RAM overload |
| ResNet50 | 100% | 80% |

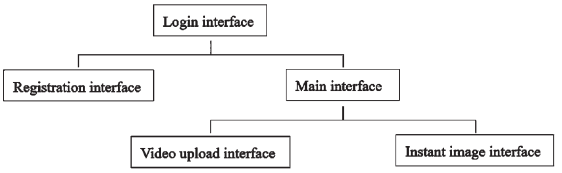



Fig. 16. App hierarchy.

greater difference in range means more confidence that the program accurately identifies who is in front of the camera. Resnet50's accuracy was 100% and the difference in prediction accuracy was approximately 80%; therefore, it was chosen for this project.

### 3.2. *Electromechanical integration between app and Raspberry Pi*

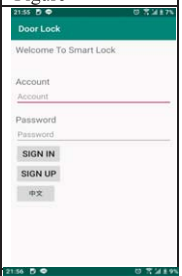
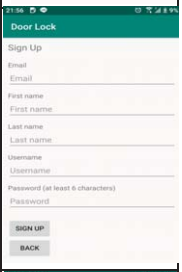
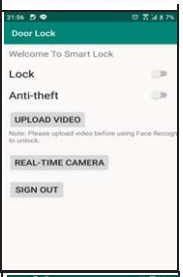
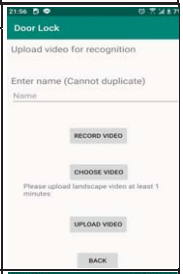
#### 3.2.1. *App interface*

To enable remote control, we designed an app with Android Studio, including the following elements: login, registration, video upload, instant image, and main interface (Fig. 16). The app's function is described in Table 2. To link the app to Raspberry Pi, we designed a back-end service platform through Firebase, which could not only store most of the data but also relay messages between the app and Raspberry Pi as a transporter.

#### 3.2.2. *App operation process*

The app has four main functions. The first is registration and login. The registration process is outlined in Fig. 17. After clicking the registration button on the login interface, users move to the registration interface. After inputting personal information, the user presses the registration button. Registration is then completed, and the data are uploaded to Firebase and stored. On the login interface, the user types the correct account and password to complete the login process. Second, the "switch button" on the app can control the lock and the antitheft system. When the switch button is clicked, the "locked" and "closed" originally set on Firebase changes to "unlocked" and

Table 2
APP interface and function introduction

| Interface | Function | Figure |
|---|---|---|
| Login interface | Log in<br>Enter the registration interface<br>Chinese and English interface switching |  |
| Registration interface | Sign up an Account |  |
| Main interface | Door lock switch<br>Antitheft system switch<br>Enter the video upload interface<br>Enter the instant image interface |  |
| Video upload interface | Shooting videos<br>Uploading videos |  |
| Instant image interface | Instant image<br>Alert (download thief photos) |  |

"open" (and vice versa). Raspberry Pi controls the door lock or the antitheft system after receiving the signal. The flow of the antitheft system is illustrated in Fig. 18. The third function of the app is video upload (Fig. 19). To unlock the door with face recognition, the user must upload a video in this interface. Users

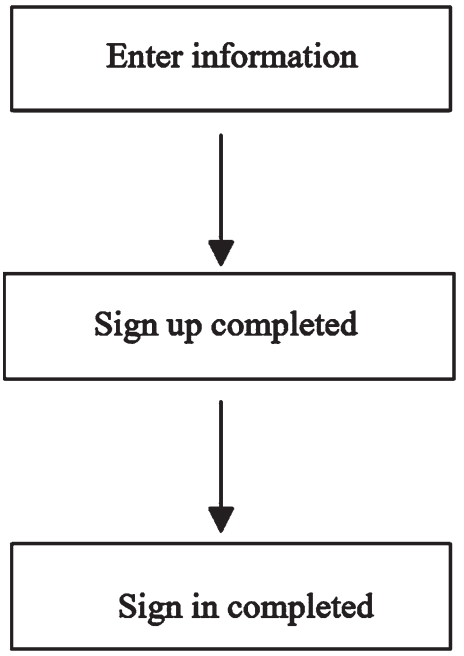



Fig. 17. Registration flow.

can start recording videos after entering their name. If a formatted movie already exists, the user can directly upload it to start the aforementioned model training. The last function is the live image, which includes an instant image and the alarm. The instant image enables users to directly view the situation outside the door through the Pi camera. As for the alarm, when the antitheft system is triggered, Raspberry Pi transfers the instant image to Firebase and then sends an alarm through the mobile phone. The user can download the photo of the person who intends to damage the door lock.

### *3.2.3. Connecting the various module parts of Raspberry Pi*

The various connected modules of the system are listed in Table. 3.

### *3.3. Door lock mechanism*

General household door locks mostly use conventional mechanical switches, and the configuration is mostly auxiliary locks with horizontal handles. The module mechanism can be used as part of the electronic lock through the transmission of a signal to the MG995 servomotor. The door lock mechanism consists of an auxiliary lock tongue, an external lock head, a servomotor, a motor mount, and a camera

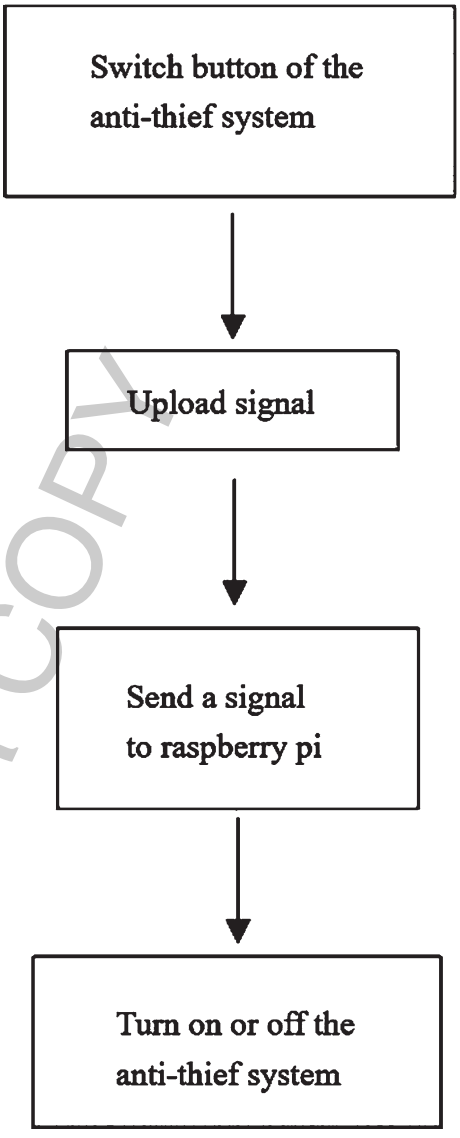



Fig. 18. Antitheft system flow.

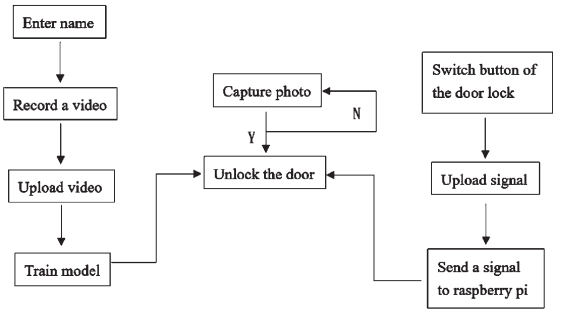



Fig. 19. Video upload and door locking flow.

Table 3
The module part list

| Name | Description |
|---|---|
| MG995 servomotor | Torque: 13 kg/cm<br>The servo control is applied to the motor. The sensor is installed on the motor and the target machine. The detection result is returned to the servo amplifier for comparison with the command value. The servomotor is controlled by a feedback signal. It controls positioning and motion speed. The main feature is that the speed can be precisely controlled. |
| SW-420 | Working voltage: 3.3–5 V,<br>When accepting the vibration, the switch is momentarily disconnected and the output terminal outputs a high level. |
| DuPont line | This is used to connect the circuit. |
| LED light | Informs the user of the state of the door lock. When it is open, the LED light is on; when it is locked, the LED light is off. |
| Resistance | Prevent LED lights from burning out. |
| Pi camera | Captures images. |

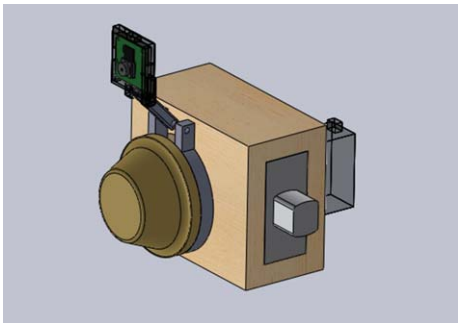


Fig. 19. Door lock mechanism combination (front).

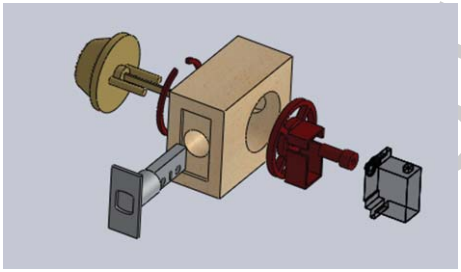


Fig. 20. Expanded view.

Fig. 21. Physical transfer bearing.

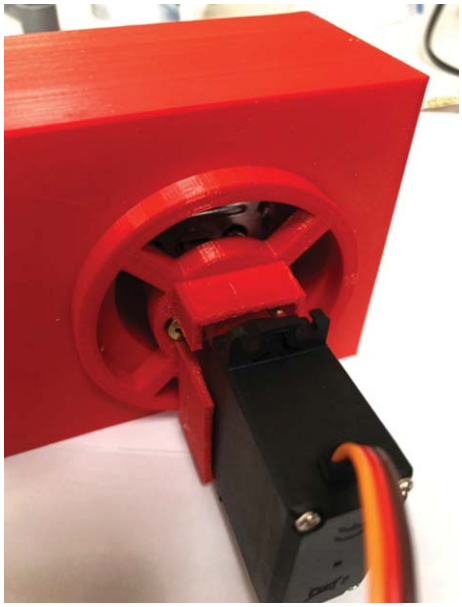


Fig. 22. Actual installation situation.

assembly (Figs. 20 and 21). Raspberry Pi serves as the core control. The servomotor of the electronic lock is given a signal to initiate the transfer bearing. The servomotor is reversed through the internal gear. A fixed angle (90°) is used to open the door. A 90° movement in the opposite direction locks the door. The prototype was constructed as illustrated in Figs. 22 and 23.

## 4. Conclusions

In this research, an electronic lock mechanism design and electromechanical integration were completed. The mechanism design integrated the door lock mechanism with the servomotor, used the signal output, and promoted the operation of the mechanism. In terms of the door lock electromechanical integration, the Raspberry Pi module circuit and the door lock were combined and the antitheft system was set up. Firebase was used for project design, data storage, and message transmission—as a dedicated space for placing the circuit and the tree. Raspberry Pi played an integral role in the overall hardware of the door lock.

Haar cascade was used for face detection and ResNet50, which exhibited 80% accuracy during testing, was used for training. The app performs the following tasks:

- Registration and login
- Face data upload
- Antitheft system alert
- Opening and closing of the door lock
- Instant image rendering

The use of "wisdom" on this page was replaced with "smart," which a more common expression for such devices. If the "wisdom" is part of the device's official name, please correct it. Your paper does not mention this term again; therefore, providing its abbreviation is unnecessary. Please review the insertion that was made here to enhance clarity. Please review this change in word choice and ensure that your intended meaning has been conveyed. Your paper does not mention this term again; therefore, providing its abbreviation is unnecessary. This was unclear. Please review the changes here and ensure that your intended meaning has been retained. Please review this change in word choice and ensure that your intended meaning has been conveyed.



## References

[1] S.M. Jhuo, Z.Y. Wu and X.L. Jiang, The Design of Keyless Entry System for Home Door Lock Application, June, (2009), pp. 1-20

[2] GitHub, (2018). opencv/opencv. [online] Available at: https://github.com/opencv/opencv/tree/master/data/haarcascades [Accessed 29 Mar. 2018]

[3] W.E. Miller, Utilizing Facial Recognition Software to Record Classroom Attendance, Auburn, Alabama May 5, 2018.

[4] P. Belhumeur, J. Hespanha and D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. ACM Digital Library (1997).

[5] N. Stekas and D. Heuvel, Face Recognition Using Local Binary Patterns Histograms (LBPH) on an FPGA-Based System on Chip (SoC). In: *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Chicago: IEEE*, 2016, pp. 300–304. ISBN 978-15090-3682-0. DOI: 10.1109/IPDPSW.2016.67

[6] Y.K. Wang, Y.X. Zheng and C.X. Lin, Classiface: Real-time Face Recognition Based on Multi-Task Convolution Neural Network, (2018), pp. 15-28.

[7] P. Kamencay, M. Benco, T. Mizdos and R. Radil, A New Method for Face Recognition Using Convolutional Neural Network, 2017. DOI: 10.15598/aeee.v15i4.2389

[8] F. Chollet, Building powerful image classification models using very little data, June, 2016 [The Keras Blog].

[9] A. Geitgey, Face Recognition Documentation, Jun, 05, 2019.

[10] Keras documentation, retrieved from https://keras.io/applications/