Techno-Science Research Journal 11(1) (2023) 30-36

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

Techno-Science Research Journal

Lock and Unlock Door with Face Detection using OpenCV, Python, and Arduino Board

Seangleng Ny[1*], Dona Valy[1,2], Phutphalla Kong[1]

[1] *Department of Information and Communication Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

[2] *Mechatronic and Informatiion Technology Research Unit, Research and Innovation Center, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

Received: 30 Agust 2022; Accepted: 04 May 2023; Available online: June 2023

Abstract: *Nowadays, smart home technology has become one of the leading IoT-based projects and as a result of that there are lots of new IoT-based products available in society that allow people to live more convenient and secure lives at home. Many people are aware of this technology, and the smart home application helps people to manage their schedules, home lighting, electricity bills, grocery lists, and also their home security. Today, face recognition is a well-established and popular process to keep homes safe. Here facial recognition helps detect and identify faces that we want to allow into our home. In this aims research, we propose an intelligent door system using an Arduino microcontroller with face recognition. It has two steps for extra security and a smart door system using face recognition. The first step is to find and predict whether the image is fraudulent or real. The fraud image is a face image taken from a screen smartphone, tablet, TV monitor, or computer laptop. If the system predicts a fraud image, it will lock the door, otherwise, if it predicts a real image, then the system will run the second step, the step of identifying the owner by face recognition. In the second step, we created two models for comparing the accuracy rate, Model 1 is a new model from the Convolutional Neural Network (CNN), and Model 2 uses a transfer learning model from the renet18 model. In both models, we get an accuracy rate on the training set of 95.83% for model 1 and 100.00% for model 2. In this second step, if the system determines the owner, the door will be unlocked in 30 seconds, and if not, it will be locked.*

Keywords: Smart Door; Machine Learning; Face Recognition

1. INTRODUCTION

Biometrics is unique to an individual and is used in many systems that involve security. In the face recognition approach, a given face is compared with the faces stored in the database to identify the person. The aim is to search out a face in the database that has the highest similarity to the given face. In this paper, an automatic door access system is proposed where the face is recognized by comparing it with faces in databases. When a face with the highest similarity is found, it is identified as an authorized user. The entire methodology will fall into two categories as face detection and face recognition. In the face detection classification between face versus non-face, the region is performed. To perform face recognition, features are extracted from the image obtained and compared with the authenticated database to identify and verify the person. Face recognition ensures security by restricting unauthenticated entry.

In the world of emerging technology, security become an essential component of day-to-day life. Information theft, lack of security , violations of privacy, etc. are the essential components that need to be protected. Using smart security systems for door locking and unlocking has become popular nowadays. This system is being adopted by many countries, and first-grade countries such as the USA, Japan, etc. already make use of it. With this, we will review other pieces of research that are related to our project with two keywords: smart doors and face recognition.

Raghu et al. [1] describes and designs systems for locking and unlocking the door using face recognition using the microcontroller Raspberry pi and Module GSM. The paper proposes to first scan faces and use face recognition to determine if the true system will be unlock the door, and if not, the system

[*] Corresponding author: Seangleng Ny
E-mail: nyseangleng@gmail.com; Tel: +855-97 733 2111

will send a message to the author. In this research paper, used Local Binary Pattern Histogram (LBPH) for face recognition with a dataset of 10 face images and an accuracy rate of 90%.

Saquib et al. [2] studied on face Recognition (Image Processing) based Door Lock using OpenCV, Python, and Arduino. The author describes and designs systems for locking and unlocking the door using face recognition using micro-controller Arduino Uno. This paper introduces an intelligent door system using an Arduino Uno micro-controller with face recognition. The face recognition method uses the built-in libraries in OpenCV and uses a single-owner dataset with 10 faces.

Nandan et al. [3] studied on smart voting system using Face Recognition. This project aims to build an Aadhaar and Web-based Voting System using Face Recognition. Person should swipe an RFID card on to the RFID reader, the RFID reader sends a character associated with the RFID card. After sending the character to the Arduino UNO, the Arduino UNO uploads the character to MATLab. After receiving the character from the Arduino UNO the MATLab asks for face authentication. If the character obtained from the MATLab and the Arduino UNO is the same, then he is a valid person and will be allowed to vote. MathLab uses face authentication, or face recognition and a set of face image datasets stored on a web server and a machine EVM for voting.

Patril et al. [4] applied New Method for Face Recognition Using Convolutional Neural Network. The author compare accuracy using Convolutional Neural Network(CNN) to three methods: Basic Component Analysis (PCA), Basic Binary Patterns Histogram (LBPH), and K – Nearest Neighbor (KNN). The experiment, using an ORL database of faces, contains ten different images of each of 40 distinct subjects (400 different images). For some subjects, the images were taken at different times and under varying lighting conditions. All the pictures are black and white with 112×92 pixels. The results of the CNN method (accuracy = 98.3%) were better compared with three other methods, such as PCA (accuracy = 85.6%), LBPH (accuracy = 89.8%), and KNN (accuracy = 81.3%).

Fatma and Shadman [5] studied on recognition of Handwritten Digit Using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers. The author describes about observing the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies.

From the literature review, have many researcher done much research on this topic or a related topic, but the system they created did not take fraudulent images into account of fraudulent images. For example, if a person accesses the system using the image of the face owner from the monitor screen (smartphone, tablet, TV Screen, or laptop), that system will be detected and predicted as the face owner. Thus, a new model which rectifies all the above issues is developedt.

2. METHODOLOGY

As mentioned in the previous section, previous research on, smart door systems using face detection and recognition without adopting algorithms such as CNN (Convolutional neural network) results in low accuracy, and the approach is not able to detect fake images which are used to access the system. From this problem, we propose to create and build a smart door system using face recognition that has high accuracy and can detect and predict fake images. The proposed system is composed of two modules: hardware and software.

2.1. Hardware

The developed system consists of multiple components hardwired as per the circuit design demonstrated in Fig. 1.

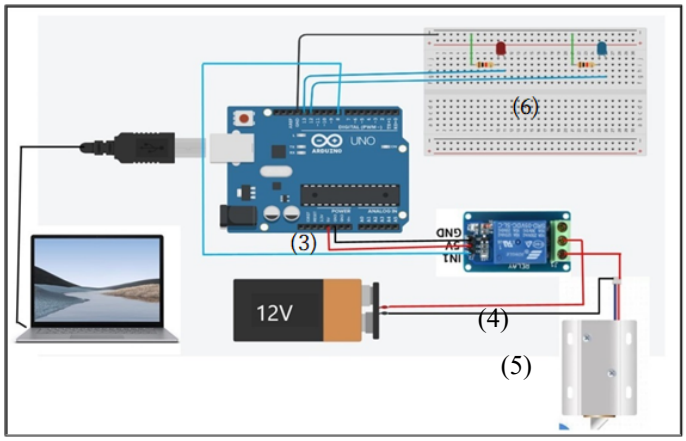

Fig. 1. Hardware Design

(1) Laptop computer: used for running the main software, webcam and supporting power supply +5v to the Arduino Board.

(2) Battery +12V (DC): Use for support power supply to solenoid lock.

(3) Arduino Board Uno: A microcontroller used for managing and controlling hardware except computers (1)

(4) Relay Module: Used to provide + 12V power to the solenoid if ordered from an Arduino board.

(5) Solenoid lock: as a lock used for locking and unlocking

(6) LED (Red and Blue): Used for the message to the owner (If LED Blue turns on, the door is unlocked, else LED red turns on, and the door is a lock).

2.1.1. Relay module

The Single Channel Relay Module is a convenient board that can be used to control high voltage and a high current load

such as motors, solenoid valves, lamps, and AC loads. It is designed to interface with a microcontroller, such as Arduino, etc. The relay terminal (COM, NO, and NC) is being broughtout with a screw terminal. It also comes with a LED to indicate the status of the relay(see Fig .2).

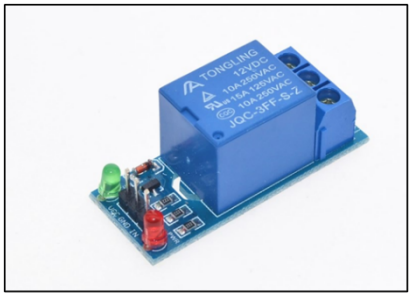

Fig. 2. Ralay 1channel 12v

2.1.2. Solenoid lock

A solenoid is a device comprised of a coil of wire, the housing, and a moveable plunger (armature). When an electrical current is introduced, a magnetic field forms around the coil which draws the plunger in. More simply, a solenoid converts electrical energy into mechanical work (see Fig. 3).

- The coil is made of many turns of tightly wound copper wire. When an electrical current flows through this wire, a strong magnetic field/flux is created.
- The housing, usually made of iron or steel, surrounds the coil concentrating the magnetic field generated by the coil.
- The plunger is attracted to the stop through the concentration of the magnetic field providing the mechanical force to do work.

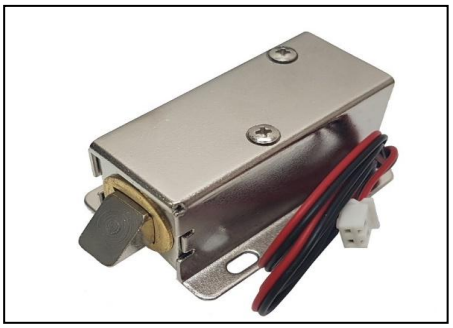

Fig. 3. 12V Solenoid lock

2.1.3. Arduino uno

The Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario, you can replace the chip for a few dollars and start over again.

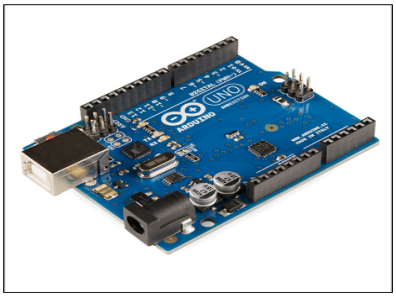

Fig. 4. Arduino uno

2.2. Software

To write a program for this project, we used the following flowchart:

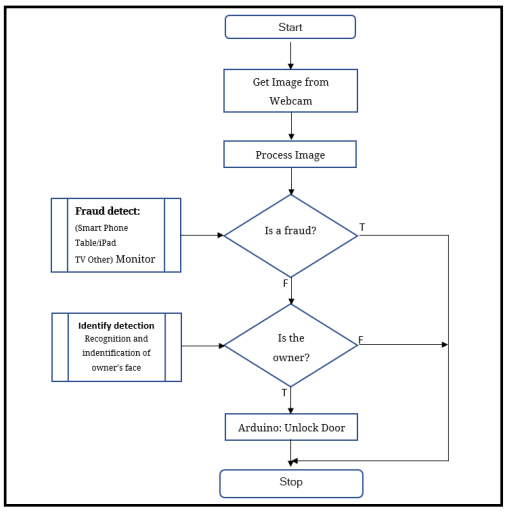

Fig. 5. Flowchart

For the sake of simplicity, we divide the flowchart process into two parts:

2.2.1. Detecting and predicting fraud image

Step 1: Start
Step 2: Get Image from Webcam Sensor: as process for receiving the image from the webcam sensor to the system for the next step.
Step 3: Process image: The Process of the image refers to the gathering of the image data from the webcam sensor. We will only need a part of the face from the whole captured image. For

this, we will have to detect the area of the face in the image. To detect faces, we use a class library in OpenCV.

Step 4: Is it a fraud? Is the step finding process fraud image, an attempt to hack into the system. Fraud images are face images show on smartphone screens, tablet screens, TV monitors, and laptop screens, which is scanned into the system (see Figs. 6 and 7)

Fig. 6. Example fraud image

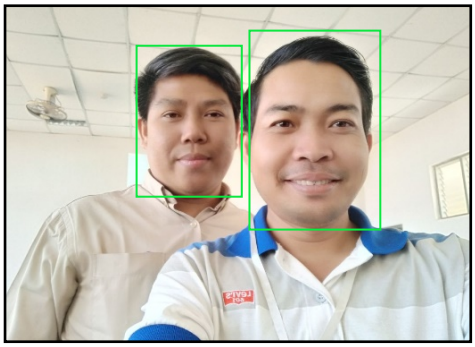

Fig. 7. Example real image

In this step, it determines the pixel location of the face image and the pixel location of the fraud image.

- If the pixel location of the face is found in the pixel position of the object (smartphones, tablets, monitors, TVs, and laptops). Then the system confirms that it is a fake image and the system will go to step 7. Then repeat the first step.
- If the pixel location of the face is not found in the pixel position of the object (smartphones, tablets, monitors, TVs, and laptops). Then the system confirms that it is a real image and the system will go to step 5.

2.2.2. Identifying owner by face recognition

Step 5: Is it the owner?: Identify the owner by facial recognition. We do comparison faces from the webcam sensor with faces in the dataset. For the compare face we use two model from CNN (Convolutional Neural Networks). A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. It has highest accuracy in comparison with the most widely used machine learning algorithms like Basic Component Analysis (PCA), Basic Binary Patterns Histogram (LBPH), and K – Nearest Neighbor (KNN) [4].

2.2.2.1. Model 1

To recognize the face, we created a model from CNN with an eight-layered neural network convolutional with one input layer followed by six hidden layers and one output layer, as designed and illustrated in Fig. 8.

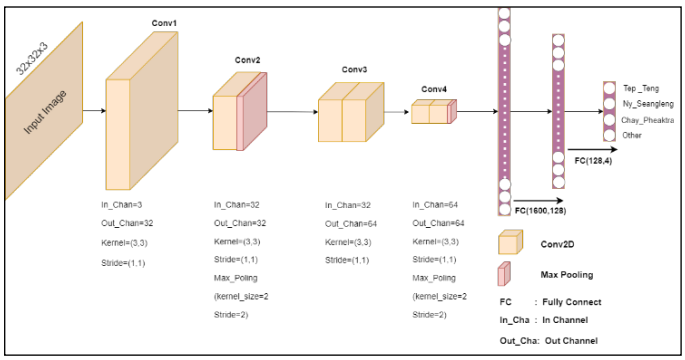

Fig. 8. Architecture of the proposed CNN (model 1)

We can calculate of feature image each layers:

In_Chann = n × n × nc
Filter = f × f ×nc
Padding = p
Stride = s

Feature image = [(n+2p – f)/s+1] × [(n+2p – f)/s+1] × nc **(Eq. 1)**

Addition, if have max-pooling = nh x nw x nc

Feature image = [{(nh – f) / s + 1} × {(nw – f) / s + 1}× nc] **(Eq. 2)**

For this model after completed:

Conv-1: Feature image size = 30×30×32
Conv-2: Feature image size = 14×14×32
Conv-3: Feature image size = 12×12×64
Conv-4: Feature image size = 5×5×64 = 1600.From convolutional we have created a fully connect consisting of an input layer, two hidden layers, and an out layer.

2.2.2.2. Model 2 (renet18 model)

Model 2 we choose to transfer learning from the renet18 model (see Fig. 9).

- Finetuning the convnet: Instead of random initialization, we initialize the network with a pretrained network, like the one that is trained on imagenet 1000 dataset. Rest of the training looks as usual.

- ConvNet as fixed feature extractor: Here, we will freeze the weights for all of the network except that of the final fully connected layer. This last fully connected layer is replaced with a new one with random weights and only this layer is trained. In the application we replace last fully connected with 4 (4 class).

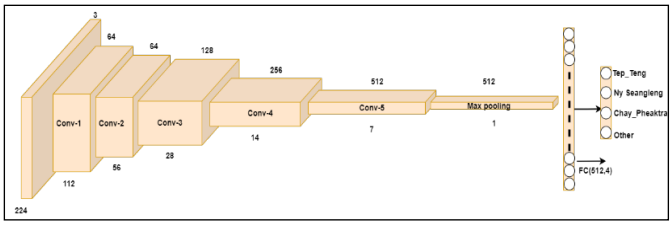

Fig. 9. Propose architecture Model2 (From renet18)

The network architecture includes five convolutional stages (see Table 1 for further details)

Tabel 1 Propose architecture from renet18 model

LayerName	Output Size	Renet18
Conv-1	112x112x64	7 × 7, 64, stride 2
		3 × 3 max pool, stride 2
Conv-2	56 × 56 × 64	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
Conv-3	28 × 28 × 128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
Conv-4	14 × 14 × 256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
Conv-5	7× 7× 512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
Average-pooling	1× 1× 512	7 × 7 average pool
Fully Connect	4	512 × 4 fully connections
Softmax	4	

Step 6: Arduino: Unlock Door an Arduino micro-controller to unlock the door for 30 seconds, if step # 5 detects and identifies the owner.
Step 7: Stop: Re-run from step 1

2.3. Experiment

2.3.1. Dataset

Dataset we use to implement and identify owner or not owner. Dataset of Faces (see Fig. 10), contains 3 (Class: Tep_Teng, Chay_Pheaktra, and Ny_Seangleng) different images of each of 100 distinct subjects, and 1000 images are of class Other (1300 different images in total). For some subjects, the images were taken at different times and under varying lighting.In this database the different moods of the images

(faces) such as open and closed eyes and laughing or without laughing with other details like having a beard or being beardless.

Fig. 10. Example dataset face images

In the experiment, we use:

- 1300 image faces for the training set (100 images of class Tep_Teng, 100 images of class Chay_Pheaktra, 100 images of class Ny_Seangleng, and 1000 images of class Other).
- 120 images face the validated set (each class has 30 face images).

2.3.2. Model Implementation

The Lock and Unlock Door with Face Detection Using OpenCV, Python, and Arduino is implemented with a deep learning approach by using Convolution Neural Network (CNN). For this research, we propose two main works:

2.3.2.1. #1 work: Fraud image detection

Fraud image detection: as part the process finding fake images trying to access the system, we select a model and class library from YOLOv3 to detect and predict fraud images to try to access the system.

2.3.2.2. #2 work: Is the owner?

For this job, the process of finding the owner by recognizing the face, we suggest classs Chhay_Pheaktra Tep_Teng and Ny_Seangleng as the owners and class Other not the owner. In the process of finding and recognizing faces, we created model1 and model2. With this dataset (dataset from 2.3.1), we do experiment with 2 models (Model1 and Model2):

- All the pictures are color RGB with size 32×32×3 pixels for model1 and size 224×224×3 for model2 with standard deviation (std=[0.2023, 0.1994, 0.2010]) and mean(mean=[0.4914, 0.4822, 0.4465])
- SGD**:** Stochastic Gradient Descent (SGD)
- Learning Rate (lr = 0.001)
- Epoch (Epoch =10)
- Batch_Size (batch_size=8)
- Momentum (momentum=0.9)

3. RESULTS AND DISCUSSION

We evaluate our model outcomes by measuring their accuracy by comparing the number of the correct predated face image to the number of all testing face images, which is shortly written in an equation below:

$$Accuracy = \frac{number\ of\ the\ correct\ predicted\ faces}{number\ of\ all\ testing\ faces} \quad \text{(Eq. 3)}$$

With the dataset of 1300 images of faces, we trained and tested the model 10 times(epoch=10) repeatedly with the result as shown in the Table 2 below.

Table 2 Accuracy model 1 and model 2

Epochs	Average accuracy rate	
	Model 1	Model 2
1,2	65.83%	97.91%
3,4	95.00%	95.00%
5,6	93.91%	98.75%
7,8	94.58%	99.17%
9,10	95.83%	100.0%

In the addition experiment, we selected 30 images of the face from each class (4 classes =120 images of the face) for determining to validate. To validate the result, we use:

Confusion Matrix: A confusion matrix is a table used to describe how well a classification system performs.

- **Precision**: is the ratio of correctly predicted positive observations to the total predicted positive observations.
- **Recall**: is the ratio of correctly predicted positive observations to all observations in the actual class.
- **F1-Score**: is the weighted average of Precision and Recall.

The result is shown Figs. 11, 12, 13, and 14.

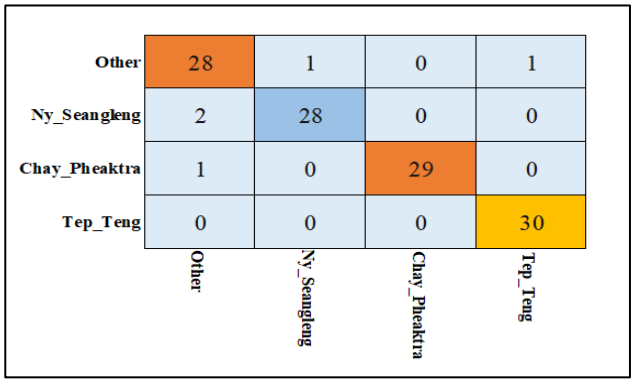

Fig. 11. Confusion matrix Model 1

In this system, we designed and developed a smart door system that "lock and unlock Door with face detection using OpenCV, Python, and Arduino board". In this, we are using the microcontroller Arduino Uno, which has many features that let the user modify it to use in different smart applications. The experiment was performed to implement the fraud image using the OpenCV library, the YOLOv3 Library, and a Python program and face recognition using OpenCV library, Pytorch library, and Python program.

Class	Precision	Recall	F1_Score
Other	0.903	0.933	0.918
Ny_Seangleng	0.966	0.933	0.949
Chay_Pheaktra	1.000	0.967	0.983
Tep_Teng	0.968	1.000	0.984

Fig. 12. Precision, Recall and F1_Score of Model 1

Class	Precision	Recall	F1_Score
Other	0.750	1.000	0.857
Ny_Seangleng	1.000	0.900	0.947
Chay_Pheaktra	1.000	0.967	0.983
Tep_Teng	1.000	0.800	0.889

Fig. 13. Confusion matrix Model 2

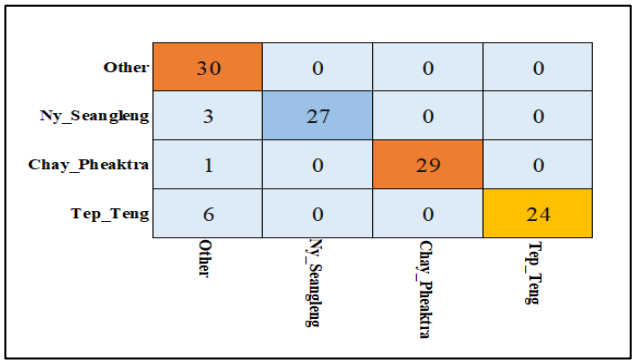

Fig. 14. Precision, Recall and F1_Score of Model 2

The individuals who were authorized to access the door stand in front of the camera, and it will capture the image of that person's face and other objects (smartphone, TV monitor, Tablet, and Laptop), If the system detects and recognizes a smartphone, TV monitor, Tablet or Laptop a system will return the fraudulent image; otherwise, it will return the real image. After the system detects a real image, the system captures the face image and compares it to the face image in the database. If the images are recognized, then the door will automatically unlock.

4. CONCLUSIONS

In this project “lock and unlock Door with face detection using OpenCV, Python and Arduino board". We designed the system, which provides security locks for doors, security, and energy efficiency for the user. In this system, we have implemented a combination of computer (webcam and run main software), Arduino Uno, relay, solenoid door lock, and LED light. And we used the OpenCV library for face detection and the YOLOv3 library for detecting objects (smartphones, tablets, computer laptops, TV screens) and predicting an image as a fraud image or a real image. In face recognition, we use the algorithm Convolutional Neural Networks (CNN) in Pytorch. Furthermore, the designed system is inexpensive due to the minimal usage of components throughout the project and uses the Arduino Uno as a microcontroller, making it is easy to use, write code, and connect to other devices.

In the future, this system can be changed into a double verification mechanism such as a retina scanner, fingerprint scanner, PIN Code, etc. This system will first recognize the face, and if the face is found in the dataset, then it will ask for a second verification mechanism. Further additions to the fraud image method, researchers can research more on fraud recognizing face images for the printed face image as a hard copy (banner, paper, etc.). It will help increase the effectiveness of security.

ACKNOWLEDGEMENTS

This research is fully funded by the University of Heng Samrin Tbong Khmum and the Institute of Technology of Cambodia.

REFERENCES

[1] Raghu Prasath, Aditya Kumar, Akanksha Yadav, Bhuvishri Acharya(IRJET, May 2020), Face Recognition Door Lock System. *Research in Applied Science & Engineering Technology.*07, 2395 – 0072.

[2] Saquib Qureshi, Yashasvi Sharma, Sail Wadwekar, Yashraj Sharma, Yuvraj Singh Panwar, Raksha Sankhala, Pawan Kumar Gupta (IJRASET, June 2020). Face Recognition (Image Processing) based Door Lock using OpenCV, Python, and Arduino. *Research in Applied Science & Engineering Technology.* 8, 2321 – 9653.

[3] Nandan Gowda S H, Jayam Haresh Tharun, Ashik B N, Deepak Lamani (IJRASET, Aug 2020). A smart voting system using Face Recognition. *Research Journal of Engineering and Technology.*7, 2395 – 0072.

[4] Patrik K., Miroslav B., Tomas M., Roman R. (aeee, 2017). A New Method for Face Recognition Using Convolutional Neural Network. Digital Image Processing and Computer Graphics. 15, DOI: 10.15598/aeee. v15i4.2389.

[5] Fatma Siddique and Shadman Sakib (ICAEE-2019). Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers. *International Conference on Advances in Electrical Engineering.* 5, 978-1-7281-4934-9.

[6] Manohar Swamynathan and Jojo Moolayi, 2017. Mastering Machine Learning with Python in Six Steps. 1st ed. apress, pp.320-327. India.

[7] Jack Purdum and Brad Levy , 2012. Beginning C for Arduino. 1st ed. Springer, New York.

[8] Scott Fitzgerald & Michael Shiloh ,2012 . Arduino Project Book. 1st ed . Torino, Italy.

[9] Adeel Javed & Lake Zurich, 2016. Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications. 1st ed. Illinois, USA.

[10] *Torch.Optim*. Retrieved on 30 April 2022 from https://pytorch.org/docs/stable/optim.html

[11] Torchvision. Retrieved on 04 May 2022 from https://pytorch.org/vision/stable/index.html

[12] Artificial Neural Networks for Machine Learning Every aspect you need to know about. Retrieved on 01 February 2022 from https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning