International Journal of
Science and Engineering Investigations vol. 11, issue 121, February 2022
Received on February 17, 2022 ISSN: 2251-8843

Development of a Real-Time Intruder Detection System Using Facial Recognition

Asianuba Ifeoma Benardine[1], Ezeofor Chukwunazo Joseph[2], Owolabi Olakunle Samuel[3]
[1, 2]Department of Electrical/Electronic Engineering, University of Port Harcourt, Rivers State, Nigeria
[3]Centre for Information and Telecommunication Engineering, University of Port Harcourt, Rivers State, Nigeria
([1]ifeoma.asianuba@uniport.edu.ng, [2]chukwunazo.ezeofor@uniport.edu.ng, [3]olakunlesamuel01@gmail.com)

Abstract- The research is aimed at developing a real time intruder detection system using facial recognition, deployable on automated teller machines in Nigerian Banking sector. This research was implemented using python programming language and raspberry pi device. Haar cascade classifier was used to implement face detection and recognition. A total of six thousand images were used to train the classifier. Five thousand positive images and one thousand negative images. The performance of the result was evaluated using five major metrics: Sensitivity, specificity, accuracy, precision and processing time. The results obtained have sensitivity 90.6%, specificity 94%, precision 94.1% and accuracy of 92.2%. A peak time of 16 seconds was taken to detect a face and a maximum of 45 seconds was recorded to recognize a detected face. Achieving a maximum of 45 seconds with 92.2% accuracy real time, indicates this work can be deployed on any real-life system.

Keywords- *Image Recognition, Face Detection, Raspberry Pi, Haar Classifier*

I. INTRODUCTION

The field of biometrics has earned the focus of many researchers and organizations. This is because it affects our daily activities, it finds application in identification and access control, and these include surveillance, detection of deception, security systems, military explorations and medical imaging. Biometric recognition is a science that studies the ability to differentiate persons based on their computable biological (anatomical or physiological) or behavioral characteristics. The palm, voice, iris, fingerprints, and face recognition are examples of biometrics modalities as in [1].

Among the many biometric pointers, the human face is one of the most significant elements used to identify individuals. Every face has several unique landmarks that make up facial features, for instance the shape of the eyes, nose, cheekbones, and jaw. In Nigerian banking system every banking institution has image capturing devices, however research shows that most capturing devices are not used to process identification and recognition rather for record purposes and audit trail. However, in automated teller machines some banks have implemented face identification before transactions can be carried out as mandated in the Central Bank of Nigeria circular referenced BPS/PSP/GEN/CIR/05/001 on installing anti skimming devices on all ATM nationwide, but face recognition is not used as authorizing factor to transactions. This research aims at breaching this gap using face recognition for transactions bases, record purposes and evidential audit trail.

Technological advancement is usually faced with some challenges, among which are security and access control. In the last two decades, researches have explored different opportunities in tackling these challenges. This opened a wide range of research field as human biometrics became a viable option. Due to the uniqueness of some human parts, it has provided authentic solutions to access control and security. However, many parts such as fingerprints and palm prints are limited because they require physical contact while voice and iris are limited because they require extreme focus and specificity. Hence facial recognition systems become more of a best fit. Although not without its challenges, such as aging, face change or external objects on the face.

In financial investigation in Nigeria banking systems, the use of other biometrics is limited compared to facial recognition. The use of cameras on ATM machines and CCTV cameras within banking halls indicates the importance and presence of facial recognition systems. However, research shows that most of these devices just captures the images but does not process for identification and recognition as stated in ATM security of central bank of Nigeria Standards and guidelines on Automated Teller Machine (ATM) operations. Hence this research intends to improve financial investigation in the Nigerian banking systems by exploring recognition systems for transaction authentication within the banking hall and over the Automated Teller Machines (ATM).

II. REVIEW OF RELATED WORKS

In [2], Face Recognition Using Content Based Image Retrieval for Intelligent Security was discussed. The research explored content-based image retrieval method on 10,000 facial datasets. The technique was based on extracting frequency-based features and other salient features from images. The retrieval process was executed by matching query image and images in the database. Java programming language was used to implement the process and an accuracy of 75% was achieved. The limitation to this research is the low accuracy and slow processing time.

In [3], Deep Face Recognition for Biometric Authentication was developed. Pre-trained convolutional neural network (CNN) based on facial recognition system was used. The former detects faces, extract features and as well recognize detected faces. A data set of 9000 images was used, 70% images were used for fine tuning a pre trained CNN model while 30% was used for performance evaluation. The CNN model used were squeezenet model with 1,240,000 training parameters and Resnet50 model. Squeeznet model gave an accuracy of 98.76% with a low computational cost while the Resnet50 model gave an accuracy of 99.41% with computational cost 20 times that of squeezenet. The limitations to this research are training the neural network requires high computing power.

Face Recognition using Histogram of Gradient (HOG) and Different Classification Techniques was explained in [4]. Viola Jones algorithm for face detection was proposed. The HOG was implemented for feature extraction and applied layers of recognition techniques to obtain accuracy in recognition. KVKRG data base of 500 images was used for performance evaluation and testing. SVM gave a Recognition Rate (RR) of 88%, Quadratic SVM, Cubic SVM and Core Gaussian SVM gave 90% RR, Linear SVM gave 91% RR, KNN produced 92% RR, and then Linear Discrimant produced a RR of 100%. However, research accuracy is limited to the KVKRG database as this was the testing, training and evaluation data set while processing time was not measured.

A Review on Different Face Recognition Techniques [5], explored extensively varieties of techniques for recognition only. The research reviewed PCA, LDA, ICA, SVM, Gabor wavelet and ANN with their hybrids. They concluded that individual technique yields lower accuracy than hybrids as the limitation of one method could be augmented by the strength of another. It was therefore recommended that a combination of ANN, SVM and SOM techniques will produce higher accuracy and better efficiency.

IoT Based Real time Face Recognition Door Lock System using Neural Network was implemented in [6]. They deployed Histogram of Oriented Gradients (HOG) for face detection, Principal Component Analysis (PCA) and back propagation neural network for face recognition on a raspberry pi device. The research achieved an acceptance ratio of more than 90% and execution time in few seconds.

In [7], Performance Analysis of Human Face Recognition Technique was discussed. They compared basically three recognition techniques, the fisherface, eigen faces and Linear Binary Pattern Histogram (LBPH). An OpenCV library was used for detection and performance evaluation. The research presented that LBPH has an accuracy of 80%, Fisherface 78%, and Eigen face 70%. Hence the LBPH technique is preferable to either fisherface or Eigen face technique.

In [8], BAT for facial recognition using sensors in Automated Teller Machines (ATM) was implemented. Facial detection as a secondary password while using an ATM was proposed. The research explored the use of local binary pattern Histogram (LBPH) and the fisher-face method for facial detection and recognition. Alarm system using Arduino GSM module was included for users whose face were not identified. However, the model was only implemented as a software on a PC, face detection and keypad were modeled using the web camera of the PC and on-screen keyboard. The research however recommended that other biometrics features can be explored to enhance ATM security.

In [9], Face detection and recognition using python programming language was implemented. They explored the python simple CV and OPENCV libraries in implementing Histogram of Gradient (HOG) method. The research also considered when images captured were not frontal images by using Deep convolution neural network and support vector machine.

Occlusion Face Detection Technology Based on Facial Physiology was detailed in [10]. They considered facial detection and recognition with large area occlusion i.e. from images of large areas where the human face appears very small or blurred. In the research; Adaboost cascade classifier was employed based on the Haar feature algorithm, the human eye and the mouth detected using OPENCV cascade classifiers. The accuracy of large-area occlusion of various occludes was compared. It was verified that the technique is reasonable and robust. However, the system was limited to detecting faces and not recognition.

Real time face detection and recognition system [11], used Haar cascade classifier and PCA in developing recognition system and implemented the system using C#.net programming language. The research was aimed at achieving an automatic face detection using a regular computer's web camera in real time. Although details of their training and testing data set were not discussed, their major achievement was being able to introduce an unidentified face into the database.

In [12], Real-Time Face Detection and Recognition in Complex Background was implemented. They worked on providing effective and robust algorithm for face detection and recognition in complex backgrounds real time. The research was implemented with methods like; Ada Boost, cascade classifier, Local Binary Pattern (LBP), Haar-like feature, facial image pre-processing and Principal Component Analysis (PCA). Large databases with faces and non-faced images were used to train and validate the detection and recognition algorithms. This research achieved an overall true-positive rate of 98.8% for face detection and 99.2% accuracy for facial recognition.

III. DESIGN METHODOLOGY

The system design methods adopted for this research are prototyping methodology for the hardware implementation and agile methodology for the software development.

A. System Hardware

The system hardware is made up of USB key board, USB mouse, Raspberry pi camera, SD card, monitor, Raspberry pi 4 board and Speakers. The blcok diagram of the system hardware is shown in figure 1.

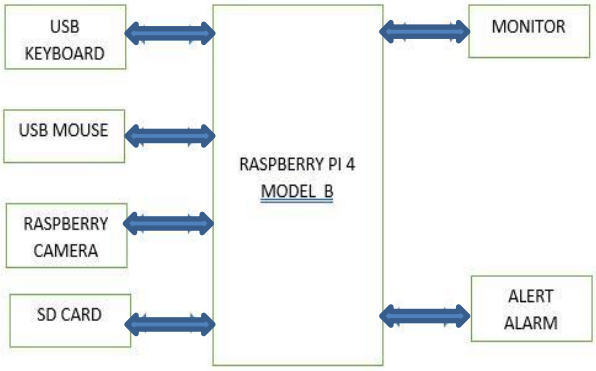

Figure 1. System hardware block diagram

B. *Image Data set*

A total of six thousand 6000) images obtained from Face scrub online data set was used to train Haar cascade classifier. Sample downloaded images in the dataset are shown in Fig. 2.

Figure 2. Sample data of testing images gotten from face scrub data set [13]

C. *Haar Cascade Algorithm Design*

The Haar cascade algorithm was deployed basically for human facial detection. Four stages were used during model development which are: selecting the Haar like features, creating integral images, adaboost training and cascade classifiers. In selecting the haar features all images were pre-processed. The preprocessing of the images involved converting them to gray scale images so we can obtain a two by two (2 by 2) dimensional matrix of each image. All obtained images from the data sets were filtered for noise by applying a median filter. A median filter is a nonlinear filter that calculates the median values of the pixels of the unprocessed image and produces the mean value of the input image. This filter was used for noise removal over all training data set. After applying the median filter each image was converted to its gray scale form. The gray scale representation of an image produces a two-dimensional matrix this is easier for image processing. The following algorithms were used to develop the Haar cascade classifier:

Start:
Step 1: consider adjacent rectangular regions in detection window
Step 2: sum up the pixel intensity in each region
Step3: calculate the difference between the sums
Step 4: compare with existing threshold value
Step 5: decide if yes or no
Stop.

The flow chart shown in figure 3:

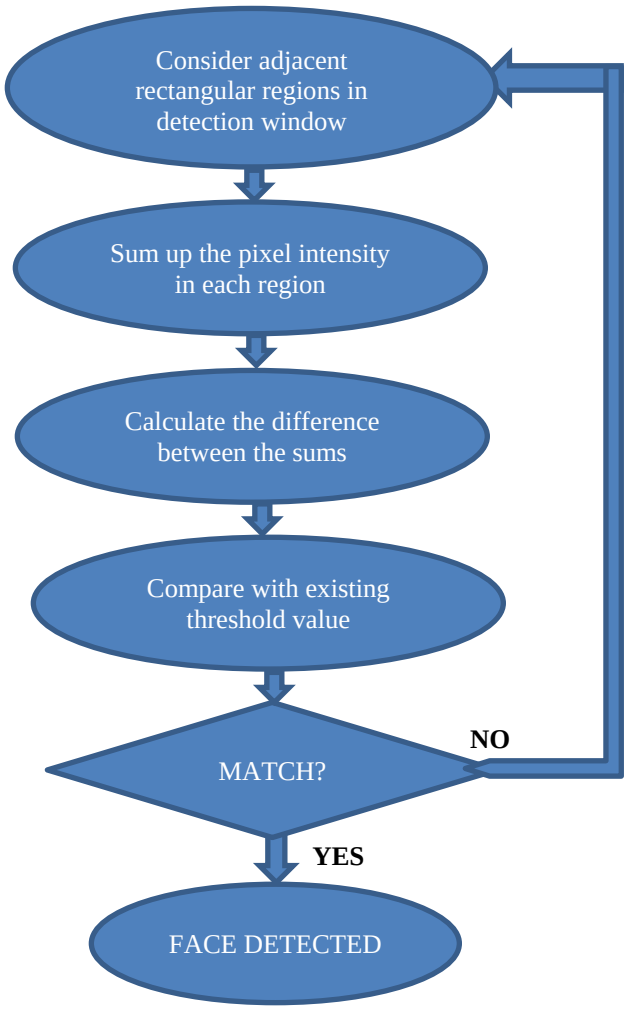

Figure 3. Haar cascade flow chart

The Haar cascade classifier was developed by first extracting features from positive images and negative images. These features were used to train the classifier. Both extracting and training was implemented using python programming language.

D. *Haar Cascade Equations*

Every single gray scale image has its mathematical representation as a two-dimensional matrix. Here, features were obtained by overlaying a rectangular representation on

each pixel. For each image, sixteen thousand (16000) Haar features were obtained. However, not all these features are entirely useful. The useful features are those with a high delta difference between the white and dark pixels. The delta (D) of each region of the Haar feature was calculated using equation 1. Delta D = white region – black region

$$D = 1/n \sum\_dark \llbracket 1(x) \rrbracket - 1/n \sum\_white \llbracket 1(x) \rrbracket \quad (1)$$

When the D is has a high value the haar features at that point is activated and used. Usually, D has a high value around the eyes nose lips and edges of the face.

E. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a type of neural network commonly used for pattern recognition. It was deployed for facial recognition. The CNN architecture for the facial recognition is made up of the following layers:

- Input layer – is a NumPy array of (img_width, img_height, 1)
- Conv2D layer – 32 filters, filter size of 3
- Activation layer – ReLU
- Max_Pooling2D layer – applied the (2, 2) pooling window
- Drop_Out layer, at 25% – this prevented overfitting by dropping some of the values from the previous layer randomly using dilution technique
- Conv2D layer – 64 filters, filter size of 3
- Flatten layer – transforms the next layer's data
- Activation layer using the ReLU function
- DropOut layer, at 25%

F. Face Recognition Procedures

The research work implemented face detection and recognition using python programming language on a raspberry pi computing system. For every image, a face detection algorithm was implemented. If face is detected, the image features would be extracted and compared with features of approved faces in the database. If recognition is positive, system approves transaction, else, denies transaction. For each of this process, the process time was calculated and displayed. The face recognition flow chart is shown in figure 4.

- *Calculating time for execution*

To calculate the time it takes for the operation to be executed, a function record is called which comprises of the clock cycle of the executed program and a function that calls the frequency of the clock cycle. This records the number of clock cycles per seconds. To get the time taken to execute the command, the clock cycle at the beginning of the program is subtracted from the clock cycle at the end of the cycle and divides by the frequency.

$$Time=(e2-e1)/cv2.getTickFrequency() \quad (2)$$

Where e2 is clock cycle at the end of program and e1 is the clock cycle at the beginning of program.

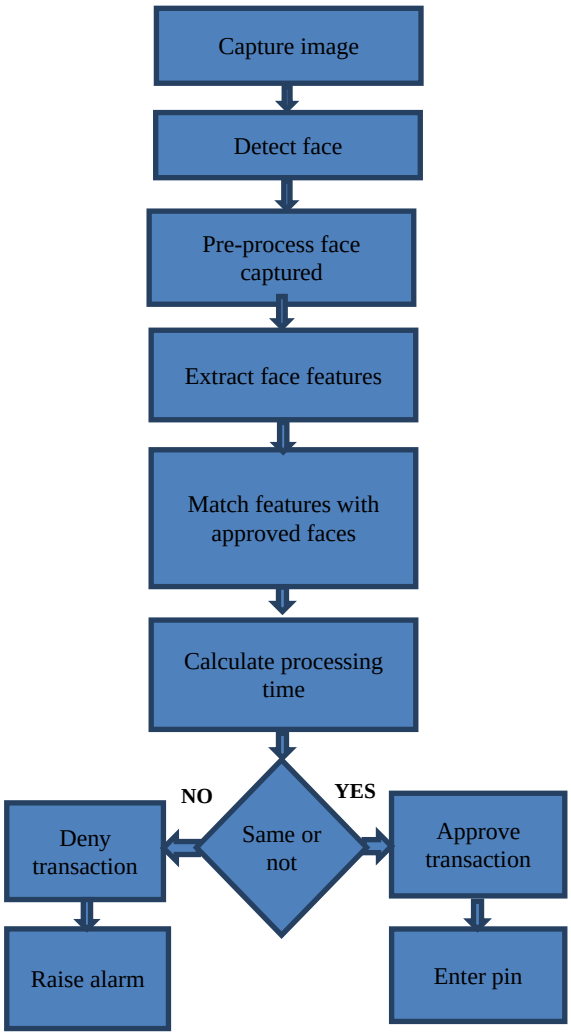

Figure 4. Model processing Steps Flow chart

- *Performance Evaluation Equations*

The classifier's performance was evaluated based on four major metrics: accuracy, precision, specificity and sensitivity. Sensitivity indicates how well a classifier recognizes a positive case (a face), specificity indicates how well it ascertains negative cases (images that are not a human face). Accuracy is the measure if both specificity and sensitivity while precision is how the classifier's decision is compared to its training information.

These metrics are represented mathematically in equation 10 to equation 13:

$$Sensitivity = TP / (TP + TN) \quad (10)$$

$$Specificity = TN / (TN + TP) \quad (11)$$

$$Accuracy = TP + TN / (TP + TN + FP + FN) \quad (12)$$

$$Precision = TP / (TP + FP) \quad (13)$$

Where; True Positive (TP) represents correctly classified positive cases. True Negative (TN) represents correctly negative cases. False Positive (FP) represents incorrectly classified negative cases. False Negative (FN) represents incorrectly classified positive cases.

G. Model laoded in Raspberry Pi

The input-output modules are interfaced with the Raspberry Pi 4. The input part, the keyboard and mouse are used to input commands to the Raspberry pi and Webcam to capture human faces (images), which are stored in Secure Digital Card for further use. The system setup is shown in figure 5.

Figure 5. System setup and tested

The system was configured using the following steps:

Step1: set up device hardware, by physically connecting the peripherals
Step2: downloading and installing the Raspbian OS on the memory card. For this research NOOBs was used for this steps.
Step3: Power up the raspberry device, select the OS and the device is up and ready.

After the raspberry pi hardware had been set up successfully the following steps were used in installing and configuring python on the raspberry pi device.

Step1: The operating system was updated using the code: 'sudo apt -get update'
Step2: Python pre-requisites were installed using; 'sudo apt-get install -y build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev tar wget vim'
Step3: python 3.8 was downloaded
Step4: python was installed using:
'sudo tar zxf Python-3.8.0.tgz
cd Python-3.8.0
sudo ./configure --enable-optimizations
sudo make -j 4
sudo make altinstall'

OpenCV was also installed using the following steps:
Step1: Update the packages
Step2: Install OS libraries
Step3: Install python Libraries
Step4: Downloaded Opencv and Opencv contrib
Step5: complied and installed opencv with contrib modules
Step6: Reset swap files.

IV. RESULT OBTAINED

A. Results of Face detection

After training the classifier and testing the classifier on the database. The developed face detection algorithm was tested with a raspberry pi camera. The real time implementation was done on 50 different persons. Figure 6 shows samples of face detected from a real time image capturing device. The images were taken at different lighting backgrounds, yet faces were detected.

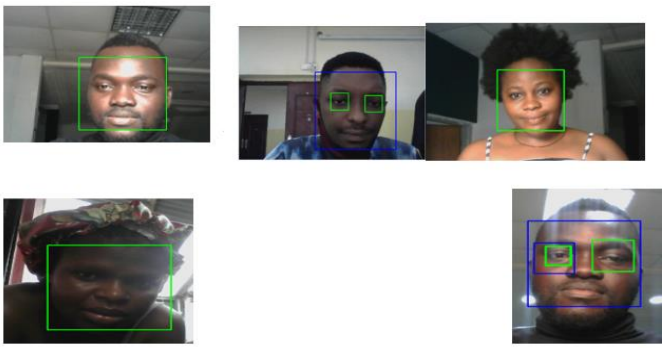

Figure 6. Results of face detection from a real time capturing device

B. Result of Time calculation for face detection

It took approximately 16 seconds to detect a face from a live video using Haar cascade algorithm as shown in figure 7.

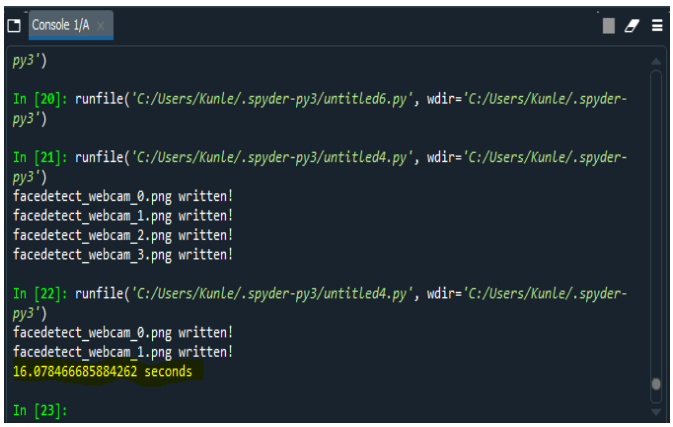

Figure 7. Time for face detection

C. *Recognizer TrainingResults*

Different techniques were applied to achieve the training of the classifier such as feature matching, meanshift and camshaft, Support vector Machine and Haar cascade. Table 1 shows the results in terms of accuracy and speed.

TABLE I. TECHNIQUES FOR TRAINING THE CLASSIFIER

s/n	Method	Accuracy	Speed	No of test items
1	Meanshift and Cam shift	86%	103 sec	50 images
2	Support Vector Machine (SVM)	91%	62sec	50 images
3	Feature Matching	95%	2 min 14 sec	50 images
5	Haarcascade	92.2%	< 45 secs	50 images

D. *Time for Facial Recognition using CNN*

The time taken to recognize human face using CNN algorithm as tested is shown in figure 8. It took 42.13sceonds to recognize face.

Figure 8. Time taken to recognize face

The unrecognized face pattern is shown in figure 9.

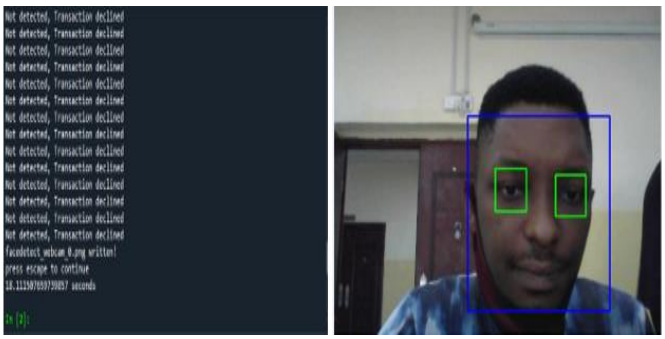

Figure 9. Result of unrecognized face and time

The tested performance metrics of the Haar cascade classifier is shown in table 2.

TABLE II. PERFORMANCE METRICS OF THE HAAR CASCADE

S/N	Performance Metrics	HAAR Cascade
1	True Positive (TP)	48
2	False Negative (FN)	5
3	True Negative (TN)	47
4	False Positive (FP)	3

Sensitivity TP/(TP+FN) = 90.6%
Specificity TN/(TN+FP) = 94%
Precision TP/(TP+FP) = 94.1%
Accuracy (TP+FP)/(TP+TN+FP+FN) = 92.2%

V. CONCLUSION

The aim and objectives of the research; to design and develop a real time intruder detection system using facial recognition has been achieved and objectives met. The face detection system can detect faces from a real time video input in approximately 10 seconds. It further records it and then recognizes the face against existing authorized faces on the database, with an accuracy of 92% and within measured time of 42 seconds which is the major contribution to knowledge as the system was designed for automated teller machines in Nigerian banking sector. Processing time is of great essence as the efficiency of a system in the banking sector is measured in transaction processing time.

The research was implemented using python programming language on a raspberry pi 4 model B device as a prototype. This is because the research was designed to be implemented on any kind of system especially an Automated Teller Machine as second authentication for transacting over the machines and to reduce ATM related frauds. For future work a third-party authentication can be considered in the event an unidentified person is captured.

REFERENCES

[1] D. Cazzato, A. Evangelista, M. Leo, P. Carcagnì, and C. Distante, “A low-cost and calibration-free gaze estimator for soft biometrics: An explorative study,” Pattern Recognition Letters, vol. 82, pp. 196–206, Oct. 2016, doi: 10.1016/j.patrec.2015.10.015.

[2] S. Karnila, S. Irianto, R. Kurniawan, “Face Recognition using Content Based Image Retrieval for Intelligent Security,” International Journal of Advanced Engineering Research and Science, Vol.6 no. 1, pp. 91-98, 2019.

[3] Z. Maheen, S. Fatima, J.K. Muhammad, K. Khurram, “Deep Face Recognition for Biometric Authentication,” Proc. of the 1st International Conference on Electrical, Communication and Computer Engineering (ICECCE), Swat, Pakistan, 24-25 July 2019..

[4] G. S. Pravin, R. M. Ramesh, “Face Recognition Using HOG and Different Classification Techniques,” International Journal for Research in Engineering Application & Management (IJREAM). Pp.29-32, 2019.

[5] S. Kamini, P. Prashant, "Review of Face Recognition Techniques," International Journal of Computer Applications. Vol. 133 no.12, pp. 20-24, 2016

[6] Z. Z.M. Khaing, "IoT based Real Time Face Recognition Door Lock System using Neural Network," International Journal of Trend in Research and Development, Vol. 6. No. 2, pp. 38-41, 2019.

[7] P. Keyur, M. Dev, M. Chinmay, G. Rajesh, K. Neeraj, A. Mamoun, "Facial Sentiment Analysis Using AI Techniques: State-of-the-Art, Taxonomies, and Challenges," Published in IEEE Access, Vol.8, pp. 90495-90518, 2020

[8] M. Jaganiga, S. Vaitheswari, R. Rasitra, S.Lakshmi, "BAT for Facial recognition using sensors in ATM," International Research Journal of Engineering and Technology (IRJET). Vol. 5 no.2, 2018.

[9] D. Tejashree, U. Urvashi, C. Rakshandha, "Face Detection and Recognition using OpenCV and Python," International Research Journal of Engineering and Technology (IRJET), Vol.7 no. 10, pp.1269-1271, 2020.

[10] G. Zhuohao , Z. Weixing , X. Luwei , H. Xiaohui , Z. Zehao, H. Zhou, "Occlusion Face Detection Technology Based on Facial Physiology," Published in: 2018 14th International Conference on Computational Intelligence and Security (CIS), Hangzhou, China.

[11] N.Niranjani, B.Tharmila, C. Sukirtha, K. Kamalraj, S. Thanujan, P.Janarthanan, N.Thiruchchelvan, K. Thiruthanigesan, "The Real Time Face Detection and Recognition System," International Journal of Advanced Research in Computer Science & Technology (IJARCST), Vol.5 no. 4, pp. 52-59, 2017.

[12] X. Zhang, T. Gonnot, J. Saniie, "Real-Time Face Detection and Recognition in Complex Background," Journal of Signal and Information Processing, Vol.8, pp. 99-112, 2017.

[13] Facesrub, A Dataset with Over 100,000 Face Images of 530 People, http://vintage.winklerbros.net/facescrub.html

How to Cite this Article:

Benardine, A. I., Joseph, E. C. & Samuel, O. O. (2022). Development of a Real Time Intruder Detection System Using Facial Recognition. International Journal of Science and Engineering Investigations (IJSEI), 11(121), 7-13. http://www.ijsei.com/papers/ijsei-1112122-02.pdf