sciendo

VOLUME 18, N° 1 2024
Journal of Automation, Mobile Robotics and Intelligent Systems

Unlocking the Future of Secure Automatic Machines: Leveraging FaceReg with HRC & LBPH

Submitted: 18th July 2023; accepted: 20th October 2023

Yamini Vijaywargiya, Mahak Mishra, Nitika Vats Doohan

DOI: 10.14313/JAMRIS/1-2024/7

Abstract:
We propose a Computer Vision and Machine Learning equipped model that secures the ATM from fraudulent activities by leveraging the use of Haar cascade (HRC) and Local Binary Pattern Histogram (LBPH) classifier for face detection and recognition correspondingly, which in turn detect fraud by utilizing features, like PIN and face recognition, help to identify and authenticate the user by checking with the trained dataset and trigger real-time alert mail if the user turns out to be unauthorized also. It does not allow them to log in into the machine, which resolves the ATM security issue. this system is evaluated on the dataset of real-world ATM camera feeds, which shows an accuracy of 90%. It can effectively detect many frauds, including identity theft and unauthorized access which makes it even more reliable.

Keywords: *ATM, Computer vision, PIN, HRC, LBPH recognizer, Face detection, Face recognition, Fraud detection, SMTP module*

1. Introduction

An Automated Teller Machine (ATM) is an electronic telecommunication device invented in early 1970s, which are one of the oldest and most secure machinery used to date, but for nearly 30 years, nothing has been done to improve this system's security, and due to the amelioration & global digitalization, it is even more vulnerable to thefts and frauds, which lead to a massive loss of capital of the users and their banks. This machine enables customers to withdraw cash from their bank accounts without having direct contact with the bank staff and have become a popular mode of transaction for financial clients, including cash withdrawals, deposits, and other transactions. Banks are becoming increasingly concerned about the security of ATMs due to the increase in cases of fraud and money loss at the ATMs.

The rapid amelioration of technology and global digitalization have led to new and more secure ATM models, as new threats also emerge day by day that could undermine their security. Despite the advantages of automation, ATM systems expose financial institutions to fraud.

Figure 1. ATM machine

The current ATM models use a card and a PIN code, which make them inclined to such attacks as a stolen card, static PINs, card fraud, and hacking of PINs. Fraudsters use numerous techniques to extract sensitive information from ATM users, including skimming devices and fake keypads. These incidents not only result in significant financial losses but also cause harm to the reputation of the banking industry.

One way to increase the security of an Automatic Teller Machine is by providing Personal Identification Number (PIN), face detection, and face recognition. Face detection algorithms like Haar cascade (HRC) and for face recognition LBPH (Local Binary Pattern Histogram) can help identify individuals attempting to conduct fraudulent transactions at ATMs.

HRC are highly accurate, fast speed, and can detect faces in real-time video/images, and on the other hand, LBPH uses micro-patterns, which describe the looks and keep execution time short.

By using cameras installed at ATMs to capture the faces of users, the Haar cascade algorithm can quickly identify the user by matching the pin of the registered person with their face, and if the user is unauthorized, it does not allow them to login to the Automatic Teller Machine and send out an alert email to the Authorized user email ID. This technology can alert bank officials of suspicious activity, allowing them to take prompt action and prevent fraud.

2. Literature Survey

In 2012, Hossein Reza Babaei, Ofentse Molalapata and AbdulHay Akbar Pandor et al., [5] developed a system using Biometrics Facial Recognition method to increase the security of the Automatic teller Machines. In this study, they built the system using Rapid Application Development lifecycle, which makes it a high quality system.

In 2015, Mohsin Karovaliyaa, Saifali Karediab, Sharad Ozac, Dr. D.R. Kalbanded et al., [8] Introduced a new concept of randomly generating OTP that frees the user from remembering the PINs during transaction at Automatic Teller machine (ATM), and features like face recognition are used with it, making the system more convenient and usable. This research study utilizes PCA based face recognition technique.

In 2018, T.S. Vishnu Priya, G.Vinitha Sanchez, N.R.Raajan et al., [7] came up with local binary pattern algorithm for face recognition (FR) in order to fulfil the downside of not identifying the identical twins in Biometric FR method. In this study they explain how the local binary patterns were used to identify the face in identical situations because the LBP method can describe appropriately the micro patterns present in the face.

As S. Hazra et al. [11] proposed, an ATM is an electronic device that allows banking transactions without staff interaction. A unique ID card with a PIN is needed to use it. A proposed Smart ATM service uses IoT and Computer Vision-based technology with fingerprint, face, and OTP verifications to enhance security and reduce fraud risk.

In 2020, M.S. Minu, Kshitij Arun, Anmol Tiwari, Priyansh Rampuria et al. [6] proposed an idea about how home security can be improved by leveraging Machine Learning algorithms for face detection and recognition using Haar cascade classifier. In this, they explained complete system flow on how the Modules are working in the project and tell how Image Identification and Recognition is being done. The KNN algorithm is used to compare the features from the image database after feature extraction from the sample image.

In 2020, Dr. S Sasipriya, Dr. P. Mayil Vel Kumar, S. Shenbagadevi et al. [9] propose that the facial recognition system should replace ATM cards with an RFID tag. The captured face image of a person is compared with the database stored image after which the output result is sent to control unit through serial communication. If the person is unauthorized, an alert message is sent to the authorized user. This study utilizes Haar cascade and Local binary pattern Algorithm.

In 2021, Anirudha B Shetty, Bhoomika, Deeksha, Jeevan Rebeiro, Ramyashree, et al. [13] compared two face recognition algorithms: Haar Cascade and Local Binary Pattern for the classification of faces in an image. They concluded that accuracy of Haar Cascade Algorithm is greater, but its execution time is also higher than local binary pattern.

In 2022, J. Ferdinand, C. Wijaya, A.N. Ronal, I.S. Edbert, and D. Suhartono et al. [4] proposed a face recognition system using FaceNet combined with the Haar Cascade Classifier. In this system, customers insert their card, and it will detect and start to identify their face. If it does not match, the card will be blocked. This proposed system achieves accuracy of 90.93%.

Table 1. Input parameters

Input			
Name	Pin	Email id	Real-Time Image Capture

3. Dataset Description

In this study, the model registers people by taking input as Name, Pin, and Email ID, and captures and stores their face images for training purposes in a CSV File.

4. Proposed System

Check Camera

The check camera module is a vital component of any facial recognition system that employs camera technology. Its whole purpose is to ensure that the cameras are functioning properly and that the images it captures are suitable for facial recognition,

	A	B	C
1	**Id**	**Name**	**Email**
2	1234	Prachita Sahu	yaminiself.20@gmail.com
3	5678	Anushka Gupta	yaminiself.20@gmail.com
4	9827	Ayushi Sharma	yaminiself.20@gmail.com
5	9644	Suhani Sharma	yaminiself.20@gmail.com
6	1201	Person	yaminiself.20@gmail.com
7	4289	Kuldeep Yadav	yaminiself.20@gmail.com
8	8296	Garvit Sharma	yaminiself.20@gmail.com
9	5769	Abhijeet Dasan	yaminiself.20@gmail.com
10	5432	Rishi Shah	yaminiself.20@gmail.com
11	1111	Pavitra Shah	yaminiself.20@gmail.com-+
12	2222	Shreya D	yaminiself.20@gmail.com
13	3333	Niyanshi Ag	yaminiself.20@gmail.com
14	8795	Abhimanyu	yaminiself.20@gmail.com
15	4646	Tejas	yaminiself.20@gmail.com
16	5467	Nilesh Parikh	yaminiself.20@gmail.com
17	6745	Madhu Parikh	yaminiself.20@gmail.com
18	3465	Viv	yaminiself.20@gmail.com

Figure 2. CSV file dataset

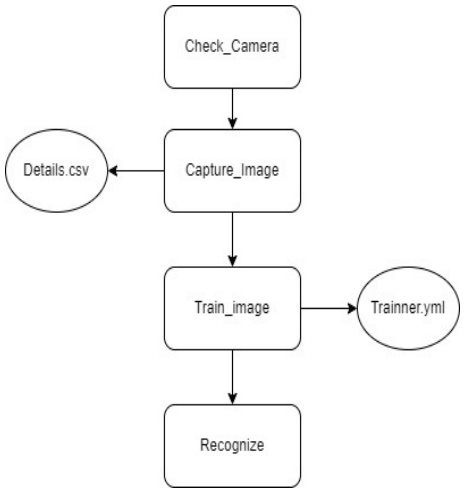

Figure 3. Modules flow

which needs good quality resolution. The higher the camera resolutions, the better the quality of the images, which can improve the accuracy of the whole system. Therefore, it is necessary to ensure that the camera installed at the ATM can capture the image in the desired resolution for optimal performance of the system. Another crucial factor to check during this module is the camera's positioning. Ideally, the camera should capture the entire face of the person standing in front of the ATM to ensure that the facial recognition system can access all the necessary facial features for accurate identification.

Capture Image

This module utilizes the Haar cascade machine learning algorithm from the OpenCV module, as explained in [3]. This component captures images of individuals standing in front of an ATM and processes them to detect the presence and location of their faces using the Haar cascade algorithm. During this process, the user is prompted to provide their name, email ID, and PIN to register as a new customer. These details are stored in a CSV file (Fig. 2). After submitting the details, the camera is activated, displaying the user's face in a rectangular frame (Fig. 4(i)). The camera captures over 100 images of the person and stores the resized images for training purposes. All these images were stored in a folder with the name, ID, and label in the JPG format (Fig. 4(ii)).

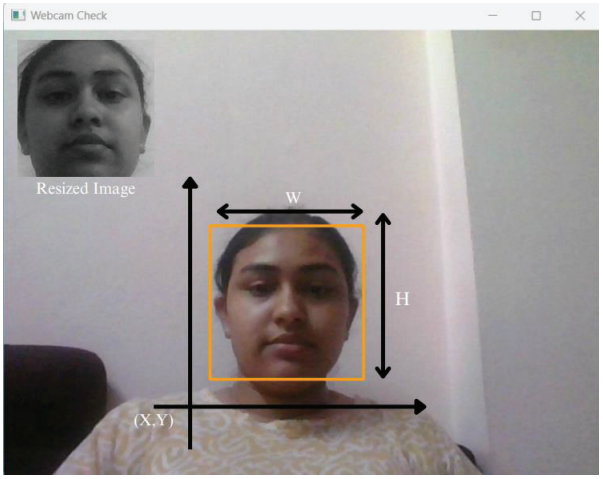

Figure 4(i). Demo of Face Detected which gets resized

Figure 4(ii). Training images

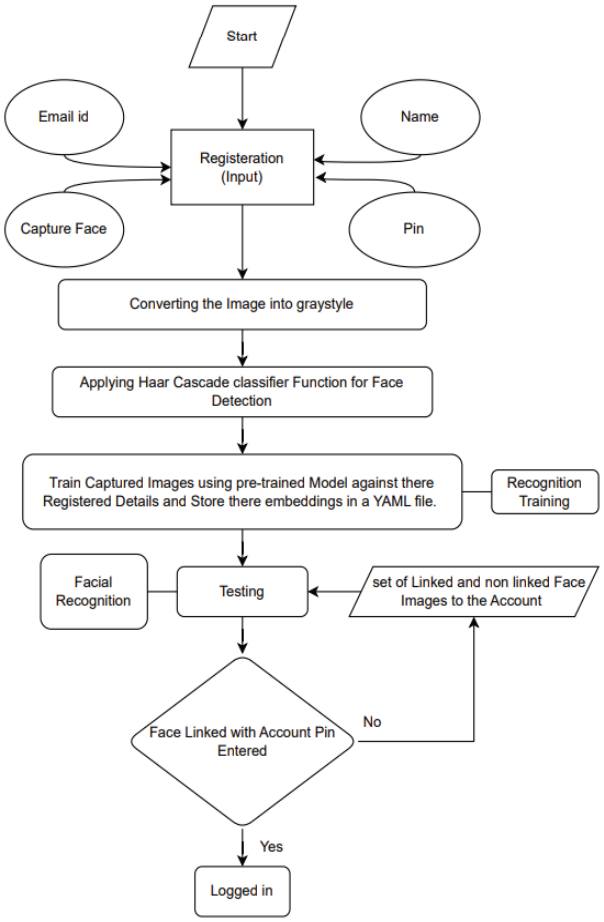

Figure 5. Flowchart for face detection & recognition

Training

To complete the task, we follow a series of steps. Initially, we load the cascade, which will act as a face detector. Subsequently, we extract the faces and their corresponding IDs from the images. then it proceeds to train the face images together with their respective IDs using the LBPH (Local Binary Patterns Histogram) recognizer function (Figs. 4(i), 4(ii), 5). These functions were implemented through the 'cv2.face_LBPHFaceRecognizer.create()' method. Moreover, we employ the 'Thread ()' function from the threading module to create a separate thread specifically for the training process. Finally, we store the obtained embeddings, or facial features, from the training in a YAML file for further step of recognizing (Fig. 5).

Testing & Recognizing

In this module, we analyze a dataset consisting of registered as well as non-registered faces to the account. To detect faces accurately, we utilize a haar cascade classifier. Subsequently, the LBPH recognizer function was used to identify and authenticate the detected faces by using the trained embeddings stored in a YAML file earlier. the recognition process comprises various scenarios. A successful match between a registered account and the detected face is considered a True Positive outcome, signifying a valid verification.

Conversely, if a registered account fails to match the detected face, it falls into the category of False Negative, indicating an inconsistency.

Also, when the model recognizes the face of an unregistered account, that's the case of False Positive, representing an incorrect identification. Lastly, when the model fails to recognize a face that is not linked to any registered account, it falls under the category of True Negative, accurately indicating the absence of a linked account.

Upon completing recognition, the system allows the user to proceed with the transaction if authorized. However, if it detects fraud, the system sends an alert email using the smtp lib module, fetching entries from the CSV file (Fig. 2) of the authorized account holder and stopping the login to the ATM machine (Fig. 10).

With this information, the bank and account holder can take necessary measures to prevent the transaction, thus preventing loss of capital and making the system more secure.

5. Software Design

5.1. Haar Cascade Classifier (HRC)

The Haar Cascade, originally proposed by Paul Viola and Michael Jones et al. in 2001 [1], is a widely used object detection algorithm specifically designed for identifying faces in images and videos.

It employs Haar features (Fig. 6), which consist of white and black pixels representing different regions of the face based on brightness (Fig. 7). To detect faces, the algorithm slides a window of fixed size across the image at various scales.

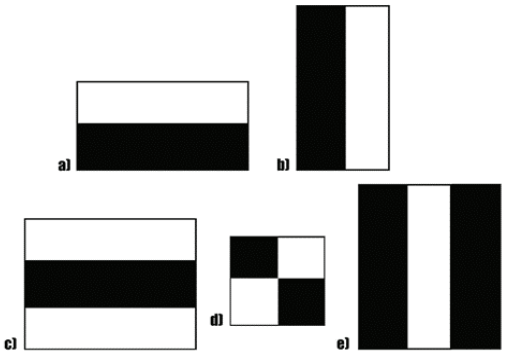

Figure 6. Haar features

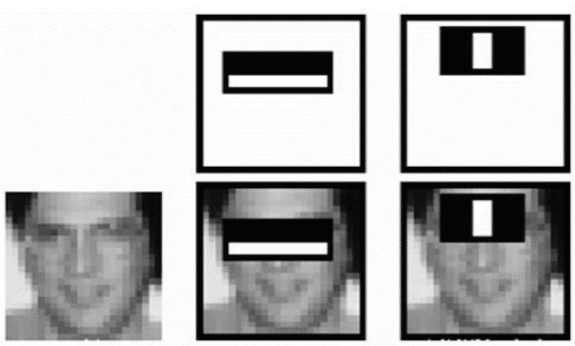

Figure 7. Haar features computing in an image

At each position, it computes five rectangular features by comparing the sum of black and white region pixels. If there are significant variations in pixel intensities or features, the algorithm identifies the region as a face; otherwise, it is a non-face region. Training the Haar Cascade model involves large number of positive images containing faces and negative images without faces. The model is composed of multiple stages, each comprising a set of weak classifiers. These classifiers are trained using Adaptive Boosting, which selects the most effective features for distinguishing between positive and negative objects. Pre-trained Haar Cascade classifier models, such as "haarcascade_frontalface_default.xml" are available in XML format on the OpenCV GitHub repository. By loading these pre-trained classifiers, real-time face detection can be performed without the need for custom training or parameter adjustment.

To apply this Algorithm, we utilized Python and OpenCV [3] function "cv2.CascadeClassifier()," which loads cascades as input, and to detect faces "detectMultiScale()" function was used, which parameters include.

Scale factor parameter is utilized to decrease the image size. A smaller scale factor can result in faster detection, but smaller faces may be missed. However, a more significant scale factor may lead to slower detection but can detect smaller faces. So, a scale factor of 1.3 is used.

The minimum neighbors parameter specifies the number of neighbors a region should have. Increasing this parameter will decrease false positives but may also miss some faces. Therefore, a value of 5 is used.

minimum size parameter (30, 30) specifies the minimum face size that can be detected. Increasing this parameter can boost the detection process speed, but smaller faces may be missed.

The flags parameter is used to enable or disable certain features of the detector, such as scaling the image with the same aspect ratio as the detector or optimizing the detector for speed, so we used "cv2.CASCADE_SCALE_IMAGE."

5.2. Local Binary Pattern Histogram (LBPH)

The Local Binary Pattern (LBP) is a well-established visual representation widely employed in computer vision proposed in [10, 12] and is specifically designed for texture categorization. It is a variation derived from the Texture Spectrum model proposed in 1990 and has gained substantial recognition.

Initially introduced in 1994, the LBP technique serves as a robust feature for texture analysis. It operates by applying the LBP operator to examine individual images as collections of micro-patterns. The frequency of occurrence of these micro-patterns throughout the image is then captured in a histogram of LBP values. To construct the feature vector, the face image is divided into non-overlapping regions (R0, R1,..., Rm).

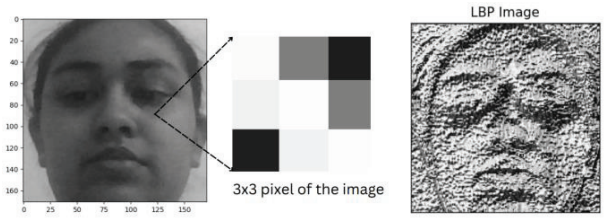

Figure 8. Transformation of gray image to LBP images

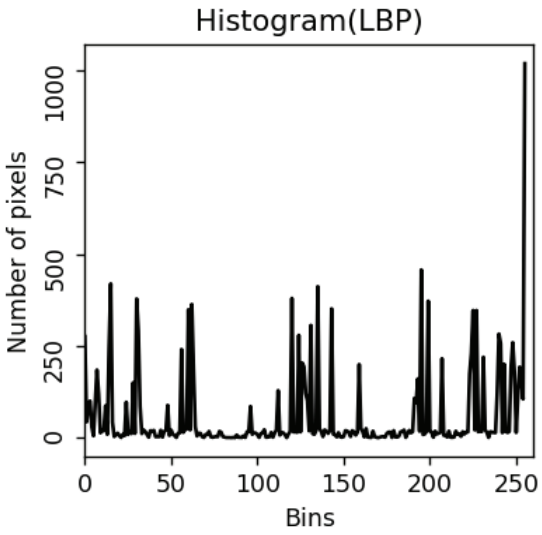

Figure 9. Concatenated histogram of each region

In the original LBP method, pixels are labeled by comparing their central pixel value (threshold) with the values of their 3 × 3 neighborhood (Fig. 8). This comparison assigns distinct numerical values to common features such as edges, lines, and points [2]. During the recognition of a test face, the algorithm calculates the LBP of the test face, divides it into regions, and creates a histogram for each region. These histograms are then concatenated into one histogram (Fig. 9) representing the entire image. Then algo compares the Euclidean distance between the histogram of the test face and the histograms of the trained faces. If the distance falls below a predefined tolerance value, it is considered a match. This approach enables efficient and robust face recognition by using the spatial information captured by the LBP operator and the histogram representation.

In the OpenCV library, the function "cv2.face_LBPHFaceRecognizer.create()" is employed for the LBPH algorithm. This function also facilitates reading the YAML file containing relevant data. The "predict()" method is utilized to predict the label and confidence value of a new face in a test image.

5.3. Threading Module

The Threading Module in Python is used to create and manage threads in a program. It allows multiple threads to run concurrently within a single process, improving the performance and responsiveness of the program. In the context of our project, it is used for image training. It can also be used to speed up the training process by allowing multiple images to be processed simultaneously. This can significantly reduce the time required for training.

Parameters in the module used in the system.

Target: It takes an array of Faces and IDs for training.

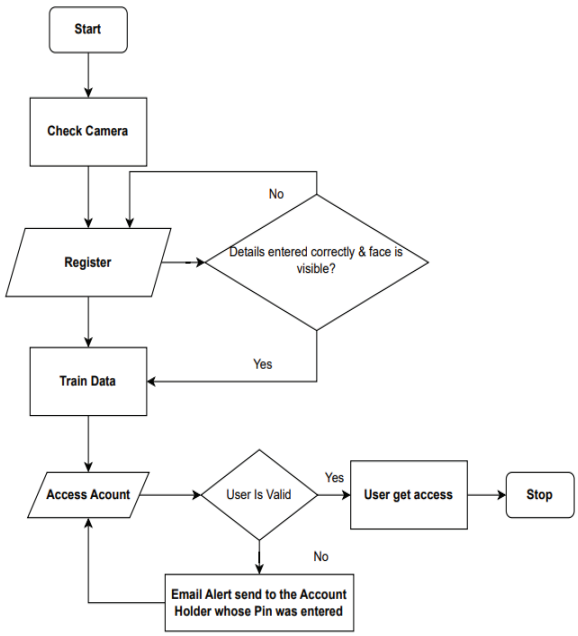

Figure 10. Block diagram of system

5.4. Smtplib

The smtplib module in Python provides a way to send emails using SMTP (Simple Mail Transfer Protocol). It allows you to connect to an SMTP server, authenticate with a username and password, and send emails to one or more recipients by using this "server.sendmail(sender_email, receiver_email, message)" function of the module.

With this, you can send text or HTML messages, add attachments, and set various email headers, such as the subject, sender, and recipient. You can also use it to handle errors and exceptions that may occur during the email-sending process.

6. Result & Discussion

We can implement this system in the field by leveraging cloud servers of banks that store the data of the registered person. By doing this, the ATM machine does not have to store the data of tens of millions of customers, and in fact, it can access this info automatically by generating the API request to those servers, which gives the access to use data of the individual for its recognition system also verify whether the customer is legit or not and this whole process will be completed within five seconds.

In this study, the execution is performed on a real-time dataset by using the Haar cascade for face detection and LBPH for face recognition. As an outcome, we found out that this method depicts a desirable result for the various measures and thus leads to the higher efficiency of our system.

Table 2. Result

Accuracy	90%
Precision	0.933
Recall	0.89
F1 Score	0.91

For the Accuracy calculation,

Case 1 – *True Positive (TP)*: The account is registered, and the Model matches the face of the person correctly.

Case 2 – *False Negative (FN)*: The account is registered, but the face does not match.

Case 3 – *False Positive (FP)*: The account is not linked yet the model still matches the face.

Case 4 – *True Negative (TN)*: The account is not linked, and the model also does not recognize the face of the person.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Were, TP: True Positive, TN: True Negative,
FP: False Positive, FN: False Negative

By using Equation (1), the Accuracy obtained from our system is 90%.

For Precision,

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$

By using Equation (2), the Precision obtained is 0.933.

For Recall,

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (3)$$

By using Equation (3), the Recall was obtained as 0.89.

For F1 Score,

$$\text{F1} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \quad (4)$$

By using Equation (4), the F1 score was obtained as 0.91.

7. Conclusion

In this study, we propose a machine learning model that can accurately detect and provide security towards any wrongful intentions of Automatic teller machine Fraud and the money within it. It can identify and issue real-time alerts/warning messages if a person's face does not match the authorized post's actual face and state, as this could raise suspicion.

Based on these messages, necessary actions can be taken immediately to prevent significant problems in the future.

Thus, with the help of algorithms like Haar Cascade and LBPH (Local Binary Pattern Histogram), a model is developed that can issue warnings and alerts to authorities before any unauthorized transactions occur. This model results in an accuracy of 90 percent with lower false positive rates, which makes it more secure & trustworthy.

Facial recognition is widely recognized as one of the most secure biometric systems, especially good for high-level security purposes like preventing any wrongful intention for the money of any account holder and providing security for ATMs.

AUTHORS

Yamini Vijaywargiya* – Medi-caps University, Indore, Madhya Pradesh, India, e-mail: yaminivijaywargiya2001@gmail.com.

Mahak Mishra – Medi-caps University, Indore, Madhya Pradesh, India, e-mail: missmahak.j@gmail.com.

Nitika Vats Doohan – Medi-caps University, Indore, Madhya Pradesh, India, e-mail: nitika.doohan@gmail.com.

*Corresponding author

References

[1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Comp. Soc. Conf. USA*, December 2001, vol. 1, p. 1, doi: 10.1109/CVPR.2001.990517.

[2] R.J. Rasras, et al., "Developing Digital Signal Clustering Method Using Local Binary Pattern Histogram," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, 2021, pp. 872–878. doi: 10.11591/ijece.v11i1.

[3] G. Bradski and A. Kaehler, "Learning OpenCV: Computer vision with the OpenCV library," *O'Reilly Med. Inc. USA*, 2008.

[4] J. Ferdinand, C. Wijaya, A.N. Ronal, I.S. Edbert, and D. Suhartono, "ATM Security System Modeling Using Face Recognition with FaceNet and Haar Cascade," *2022 6th International Conference on Informatics and Computational Sciences (ICICoS)*, 2022, pp. 111–116, doi: 10.1109/ICICoS56336.2022.9930563.

[5] H.R. Babaei, O. Molalapata, and A.A. Pandor, "Face Recognition Application for Automatic Teller Machines (ATM)," *ICIKM*, vol. 45, 2012, pp. 211–216. doi: 10.9756/BIJSESC.8273.

[6] M.S. Minu, et al, "Face Recognition System Based On Haar Cascade Classifier," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, 2020, pp. 3799–3805.

[7] T.V. Priya, G. Vinitha Sanchez, and N.R. Raajan, "Facial Recognition System Using Local Binary Patterns (LBP)," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15, 2018, pp. 1895–1899.

[8] M. Karovaliya, S. Karedia, S. Oza, and D.R. Kalbande, "Enhanced Security for ATM Machine with OTP and Facial Recognition Features," *Procedia Computer Science*, vol. 45, 2015, pp. 390-396, ISSN: 1877-0509, doi: 10.1016/j.procs.2015.03.166.

[9] S. Sasipriya, D.P. Kumar, and S. Shenbagadevi, "Face Recognition Based New Generation ATM System," *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 4, 2020, pp. 2854–2865.

[10] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence IEEE Comp. Soc.*, vol. 28, 2006, pp. 2037–2041.

[11] S. Hazra, "Smart ATM Service," *2019 Devices for Integrated Circuit (DevIC), Kalyani, India*, 2019, pp. 226–230, doi: 10.1109/DEVIC.2019.8783820.

[12] K. S. do Prado, "Face Recognition: Understanding LBPH Algorithm," *Medium*. Accessed: Feb. 16, 2024. [Online]. Available: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b.

[13] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, "Facial Recognition Using Haar Cascade And LBP Classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, 2021, pp. 330–335, doi: 10.1016/j.gltp.2021.08.044.