ML-DRIVEN PALM PRINT AUTHENTICATION SYSTEM FOR SECURITY APPLICATIONS

Burgu Maheshwari[1*], Pasupuleti Chandana[2], Arsham Arjun[2], Gandloju Ranadheer[2], Puppala Rishikesh[2]

[1]Associate Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (Data Science), Vaagdevi College of Engineering, Bollikunta, Warangal, Telangana.

*Corresponding Email: maheshwariburgu@gmail.com

ABSTRACT

Biometric authentication has gained significant traction, with palm print recognition emerging as a robust and reliable security measure. Research indicates that palm print authentication achieves an accuracy of up to 98.5%, outperforming traditional fingerprint recognition, which averages around 95%. However, conventional manual authentication methods, such as password-based systems and manual biometric verification, suffer from vulnerabilities like identity theft, forgery, and inefficiency in processing large datasets. To address these challenges, we propose an ML-driven palm print authentication system (PPAS) that leverages advanced image processing and machine learning techniques. The system preprocesses palm print images through noise reduction and edge detection to enhance feature extraction. The dataset is then subjected to a Train-Test Split to ensure robust model evaluation. Two machine learning models such as perceptron and extra trees classifier (ETC) are implemented to classify and authenticate palm prints with high accuracy and efficiency. The perceptron model offers a foundational approach to classification, while the ETC model enhances decision-making through an ensemble learning method, improving performance in feature-rich biometric data. Our proposed system demonstrates improved authentication accuracy, reduced false acceptance rates, and enhanced security, making it a viable alternative to conventional biometric verification methods.

Keywords: Palm print authentication, Machine learning, Biometric security, Image preprocessing, Pattern recognition, Extra Trees Classifier.

1.INTRODUCTION

The palmprint refers to the unique pattern of ridges and valleys found on the inner surface of the hand, excluding the wrist and fingers. Palmprints, akin to fingerprints, are a biometric characteristic unique to each individual. An earlier seminal study explored the viability of palmprints as a means of personal identification, establishing them as a form of physical biometrics. Their findings highlighted distinctive features in palmprints, including major lines (life, heart, and headlines), wrinkles, minutiae, and delta points. Each palmprint is unique, and the surface of the palm provides more information space compared to fingerprints, so it contains a greater amount of information. In general, the attributes of the palmprint manifest on multiple levels, each discernible in various types of palmprint images. Typically, these characteristics are visible across different image resolutions, with lower resolution, around 100 pixels per inch (ppi), exhibiting a pronounced texture in which dark lines are of particular significance and visibility. Notably, among these lines, the three widest and longest are termed major lines, constituted by the heart line gathering with the head and lifelines, and the remaining lines are referred to as wrinkles, as illustrated in Fig.1.

Therefore, in the case of low-resolution images, the predominant features are the major lines, wrinkles, and texture. Nevertheless, the edges of the palmprint remain imperceptible in images of low resolution. In contrast, visibility can be achieved in the case of images with high resolution, of approximately 500 dpi, which unveils local texture intricacies, including minute creases, ridges, valleys, and minutiae points.

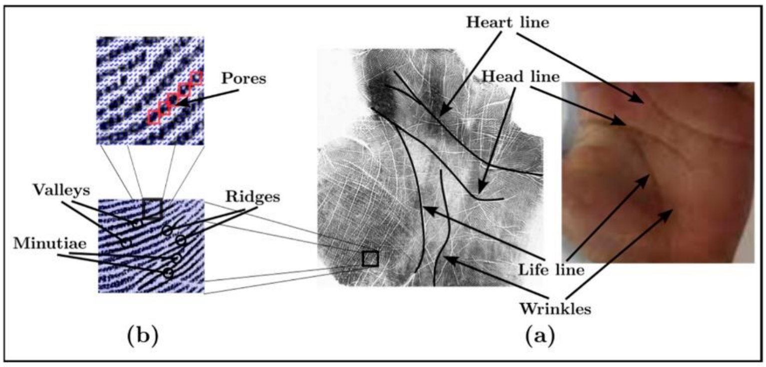

Fig. 1: Palmprint characteristics (**a**) at low resolution and (**b**) at high resolution.

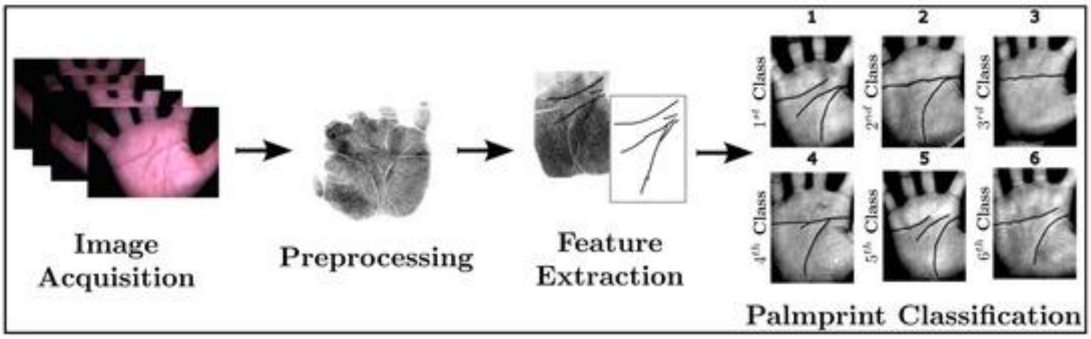

Fig. 2: Architectural framework of PPAS.

Furthermore, images with very high resolution allow an abundance of certain local particular features related to the palmprint to be visualized, including the pores, which can be seen in resolutions exceeding 500 ppi or even reaching 1000 ppi.

2. LITERATURE SURVEY

In the field of biometric recognition, facial identification still has limitations due to persistent challenges, such as pose, lighting, and orientation variations [1]. Conversely, fingerprints have been widely adopted due to their efficiency, although certain populations, such as manual workers and the elderly, may have difficulty with capturing fingerprints. In a networked society, reliable personal authentication remains critical for security [2]. Compared to other biometric modalities, palmprints have proven to be more effective and acceptable. The palmprint biometric system offers higher accuracy than fingerprints and higher acceptance than facial recognition. With characteristics such as uniqueness, reliability, and security, palmprints have been widely adopted by security agencies, providing a cost-effective and non-intrusive option for developing accurate and efficient biometric systems.

Advanced research in palmprint feature extraction [3, 4] has been conducted for contactless systems. Contactless palmprint recognition aims to improve usability and privacy. However, the lack of a knuckle guide can lead to variations in palmprint images due to hand movements. Various methods, such as the utilization of texture operators like local binary pattern (LBP) [5] and Gabor filters [6], were proposed to overcome these challenges.

Palmprint has advantages over other biometric methods, including iris and fingerprint, in terms of identity matching. Palmprints offer the advantage of easy capture with low-resolution devices, mitigating the high costs associated with other modalities. Moreover, law enforcement agencies have extensively employed palmprints for criminal identification, leveraging their unique and stable characteristics [7, 8]. These prints encapsulate diverse features like primary lines, minutiae points, ridges, and overall texture. Each feature class contributes significantly to the individuality and discriminative power of a palmprint. This flexibility permits adaptation to the specific security requirements of individuals and organizations.

Palmprint recognition has enjoyed great research popularity for identity authentication and identification in recent years. It has many unique advantages, e.g., the richness of features, high user-friendliness, suitability for private security, etc. [9]. There are many palmprint identification systems that exhibit encouraging results, but there is a need to improve the performance of the existing systems. State-of-the-art methods can be broadly organized into two main categories: Handcrafted-features-based and deep-learning-based approaches.The texture features are an important low-level feature in palmprint recognition [10] that can describe the contents and details of a specific region in an image, and for that, several handcrafted features-based approaches are based on the analysis of image texture information and provide precise features for the best palmprint recognition rate.

Zhang et al. [11] supplied a multispectral palmprint recognition approach that captured palmprint images under four bands: Red, green, blue, and near-infrared light. A score-level fusion of these bands achieved superior performance compared to any single band. Jing et al. [12] used a two-dimensional (2D) separability judgment to select DCT frequency bands with appropriate linear separability. Then from the given bands, it extracts the linear discriminant features by optimized Fisherface method and classifies by nearest neighbor classifier. Luo et al. [13] proposed a new image descriptor, local line directional patterns (LLDP). This work shows that different implementations of LLDP descriptors perform competitively in palmprint recognition. Kang et al. [14] presented a novel recognition approach for contact-free palm-vein recognition that performs feature extraction and matching on all vein textures distributed over the palm surface, including finger veins and palm veins, to minimize the loss of features information. First, a hierarchical enhancement algorithm is adopted, which combines a DOG filter and histogram equalization to alleviate uneven illumination and highlights vein textures. Second, a Root Scale Invariant Feature Transform (RootSIFT), a more stable local invariant feature extraction method compared to Scale Invariant Feature Transform (SIFT), is used to overcome the projection transformation in contact-free mode.

Recently, many systems and applications have used deep learning for biometric identification. The deep network is trained on a variety of patterns. Once the network has learned all the unique features of the dataset, it can be used to recognize similar patterns. Deep learning approaches have been used primarily to learn features for palmprint recognition. Deep learning can also be very efficient in classification (supervised learning) and clustering (unsupervised learning) tasks. In a classification task, the system classifies the input instances based on their corresponding class labels, while in clustering, the instances are grouped based on their similarity without the need for class labels. Clustering can be used for several well-known problems, such as recommender systems [15] [16] [17]. Several approaches described below are based on deep learning with classification and clustering.

Wang et al. [18] proposed 2D Gabor wavelets for palmprint images. They used a Pulse-Coupled Neural Network (PCNN) to imitate the creatural vision perceptive process and decompose each Gabor subband into a series of binary images. Entropies for these binary images are calculated and regarded as features. An SVM classifier is employed for classification. Minaee and Wang [19] proposed deep scattering convolutional network with a two-layer for palmprint recognition. Then Principal Component Analysis (PCA) is applied to reduce the dimensionality of the data. For classification, a multi-class SVM and the nearest neighbor classifier are used. Svoboda [20] proposed a Convolutional Neural Network (CNN) based on the AlexNet model and trained by optimizing a loss function related to the d-prime index to achieve a better genuine/impostor score distribution separation of touchless palmprint databases.

Meraoumia et al. [21] proposed Principal Component Analysis Network (PCANet) deep learning-based feature extraction using two stages. Then four classifiers (SVM, Radial Basis Function - RBF, Random Forest Transform—RFT, and KNN) are used with the supervised procedure. The testing was performed on multispectral palmprint databases.Cheng et al. [22] proposed Deep Convolutional Features-Based Supervised Hashing (DCFSH). They used CNN-F architecture to extract the palmprint convolutional features, followed by learning binary coding from distilled deep features. DCFSH is evaluated on a multispectral palmprint database. The Hamming distance is employed in the matching steps.

Bensid et al. [10] proposed a simple new deep learning feature extraction algorithm for an efficient multispectral palmprint identification system called Discrete Cosine Transform Network (DCTNet).

Genovese et al. [23] proposed PalmNet, which is a convolutional network that uses Gabor responses and PCA filters through an unsupervised procedure applied on different touchless palmprint databases and uses the 1-NN classifier based on the Euclidean distance for classification step. Fei et al. [24] proposed LRRIPLD which is a new Low-Rank Representation (LRR) model integrated with principal line distance for contactless palmprint recognition. LRRIPLD generates a graph that is more distinct than LRR because main line distances effectively improve clustering results by increasing the weights of the links between similar samples. The approach is tested on three palmprint databases IITD-Touchless, GPDS-Touchless, and CASIA.

Arora et al. [25] introduced PalmHashNet, a novel indexing method that learns compact feature vectors for palmprint identification. They used the Softmax loss function with additive margin to train the model to index the palmprint database and to simultaneously learn the feature vector embeddings. Furthermore, to generate an index table, the learned embeddings are indexed using the k-means clustering and locality sensitive hashing techniques. PalmHashNet is evaluated on four publicly available palmprint databases CASIA, IITD-Touchless, Tongji-Contactless, and PolyU II.Besides, Zhao et al. [26] proposed a joint constrained least-square regression (JCLSR) model with deep convolutional neural networks to solve the under-sampling classification problem by extracting different deep local convolution features using different patches of the same palmprint image. The experiments of the proposed method (JCLSR) are performed on touchless and multispectral palmprint databases.

3. PROPOSED METHODOLOGY

This research is a machine learning-based Palm Print Authentication System (PPAS) designed with a graphical user interface (GUI) built using Tkinter. Its main goal is to authenticate individuals by analyzing the unique features present in their palm prints.

Step-1: User Interface (GUI): The system utilizes Tkinter, a standard Python library, to build the GUI. This framework provides a user-friendly interface that enables users to interact with various system components through buttons, text fields, and dialogs. One of the key features of the interface is the ability to upload a dataset directory that contains subdirectories—each representing a different class or individual, with corresponding palm print images. Additionally, a text widget is implemented within the GUI to display logs, status messages, and evaluation results, offering real-time feedback to the user.

Step-2: Data Acquisition and Preprocessing: When a dataset is uploaded, the application scans the chosen directory to detect subdirectories. Each subdirectory represents a unique class and contains multiple images related to that specific individual. The images are then processed through several steps. First, they are read using OpenCV, resized to a uniform dimension of 64x64 pixels for consistency, and flattened into one-dimensional arrays. Next, the pixel values are normalized to fall within the range [0, 1], ensuring uniformity across the dataset. These processed images, along with their class labels, are stored as NumPy arrays, which enhances processing speed during future runs and improves overall reusability.

Step-3: Dataset Splitting: Following preprocessing, the dataset is divided into training and testing subsets. Eighty-five percent of the data is allocated for training the machine learning models, while the remaining fifteen percent is reserved for evaluating the model's performance. This division ensures that the model is validated on new, unseen data and allows assessment of its generalization capability.

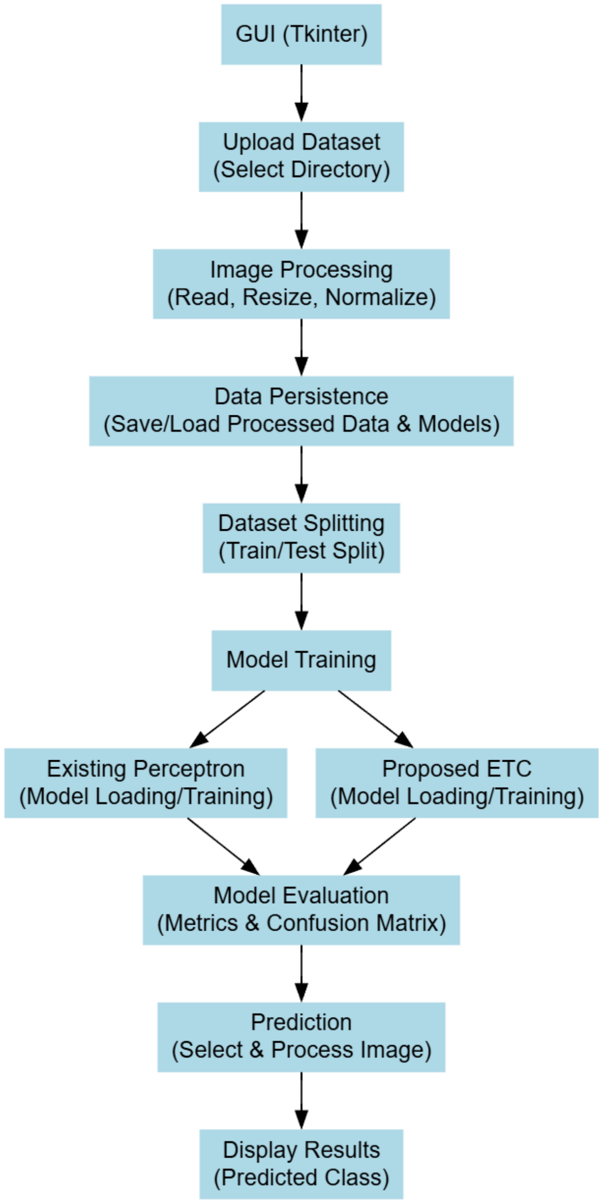

Fig. 3: Block diagram of proposed PPAS using ETC model.

Step-4: Model Training and Evaluation: The system offers two approaches for classification. The first uses a Perceptron model, which may either be loaded from disk if it has been previously trained or trained from scratch using the current training data. This model functions as a simple linear classifier. The second approach employs a proposed Extra Trees Classifier (ETC), an ensemble learning method known for its robustness and accuracy. Like the Perceptron, it is either loaded or newly trained depending on availability. Both models are assessed using multiple performance metrics, including accuracy, precision, recall, and F1-score, to provide a detailed evaluation of classification quality. A confusion matrix is visualized using a heatmap generated with Seaborn and Matplotlib, offering insights into model performance across different classes. Additionally, sensitivity and specificity are calculated to analyze the rates of true positive and true negative predictions.

Step-5: Prediction: For making a prediction, users can select a new test image through the interface. The image undergoes the same preprocessing steps used during training. Once processed, it is passed through the trained classifier—either ETC or Perceptron—to predict the corresponding class, effectively identifying the individual based on their palm print. The predicted class is then overlaid on the image, which is displayed using OpenCV to provide immediate visual feedback.

Step-6: Workflow Summary: The workflow of the system is designed to be both modular and interactive. It proceeds through the following stages: uploading the dataset, processing the images, splitting the dataset, training the model (either Perceptron or ETC), evaluating the model, performing predictions, and finally displaying the results. The modular structure of the system allows for straightforward updates and maintenance. Each component—data processing, training, evaluation, and prediction—can be independently modified or enhanced without disrupting the rest of the system.

3.1 Image Processing

The image preprocessing pipeline ensures that raw images are transformed into a structured, normalized, and labeled dataset suitable for machine learning. This method eliminates inconsistencies, enhances computational efficiency, and prepares the images for accurate classification in security-based applications like biometric authentication. Image processing is performed to prepare images for a machine learning model. The preprocessing pipeline involves image loading, resizing, normalization, and dataset storage, ensuring that the input data is structured efficiently for training. The process begins by checking whether pre-processed image arrays already exist in the model folder. If the saved arrays (X.txt.npy and Y.txt.npy) are available, they are directly loaded to save computational resources. Otherwise, the function performs the entire preprocessing sequence from scratch.

Step-1: Image Loading and Category Assignment

The function iterates through directories containing images, treating each folder as a different category. For every image file found in these directories, the function extracts the category name based on the folder structure. This is a crucial step in supervised learning tasks where images are labeled for classification. The category name is then converted into an index value, which serves as the corresponding label in the Y array. This structured labeling allows the machine learning model to associate input images with their respective categories.

Step-2: Image Resizing and Flattening

Since raw images can have varying sizes, they are resized to a fixed dimension of 64×64 pixels with 3 color channels (RGB). Standardizing image dimensions is essential to ensure uniformity across the dataset, allowing the model to process inputs of the same shape. Once resized, the image data is flattened into a one-dimensional array, transforming the 3D image structure into a 1D numerical format that can be used in machine learning models. This step reduces computational complexity while retaining critical image features for classification.

Step-3: Normalization for Model Efficiency

To enhance model performance, the pixel values of images are normalized by dividing each value by 255. Since pixel intensities range from 0 to 255, this transformation scales values between 0 and 1, making computations more efficient and preventing numerical instability during training. Normalization also improves model convergence, as smaller values help neural networks learn faster and avoid bias toward larger numbers. This step is particularly crucial in deep learning-based image classification tasks, where efficient optimization is required.

Step-4: Dataset Storage for Reusability

Once all images are preprocessed, the structured input (X) and output labels (Y) are converted into NumPy arrays for efficient storage and retrieval. These arrays are saved as .npy files, which ensures that future runs can directly load the processed dataset without repeating the entire preprocessing sequence. This significantly reduces execution time, making the system more scalable and adaptable for larger datasets.

3.2 Proposed ETC

The ETC model is an ensemble learning method that improves classification accuracy by constructing multiple decision trees in a highly randomized manner. The given function Proposed ETC implements the ETC model for palm print authentication, ensuring efficient training and evaluation. The workflow follows a structured approach, starting with model loading, training, and prediction.

Step 1: Model Loading and Initialization

The function first checks if a pre-trained ETC model exists in the specified directory. If found, the model is loaded directly to avoid redundant training, ensuring faster execution. This step is particularly useful in real-world scenarios where model retraining is computationally expensive and unnecessary for every function call. If no saved model is found, a new Extra Trees Classifier instance is created to train from scratch.

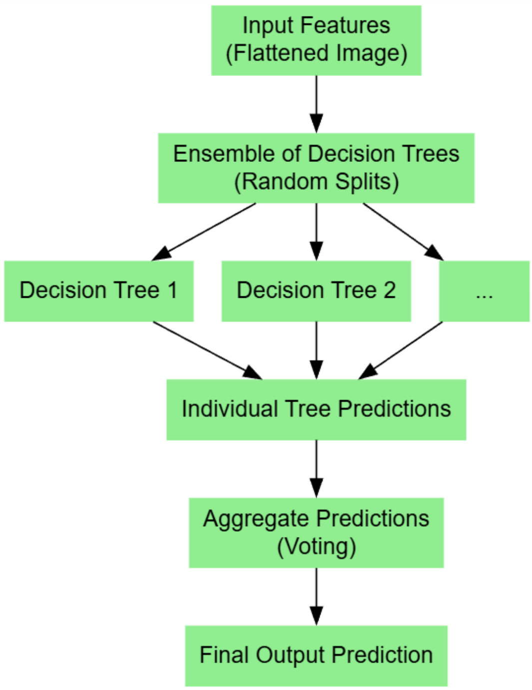

Fig. 4: ETC model architecture.

Step 2: Training the Model with (X_train, y_train)

If the model is newly initialized, it undergoes a training process using (X_train, y_train). During training, the ETC algorithm constructs multiple decision trees by randomly selecting features and samples from the training dataset. Unlike traditional decision trees, ETC introduces additional randomness by selecting split points at random, rather than choosing the most optimal split. This leads to better generalization and reduces the risk of overfitting. Once training is complete, the model is saved as a .pkl file, allowing for future reuse without retraining.

Step 3: Prediction on Test Data (X_test) and Output Generation (y_test)

After training, the model is used to predict class labels for unseen data samples in X_test. The trained ETC classifier evaluates each test sample and assigns a predicted category label, generating the predicted output y_pred. These predictions are then compared to the actual labels (y_test) using the calculateMetrics function, which likely computes accuracy, precision, recall, and confusion matrix values to assess model performance.

Step 4: Ensemble Learning and Performance Enhancement

ETC improves classification accuracy by aggregating predictions from multiple decision trees, making it more robust to noise and data variations. The additional randomness in split selection helps in reducing bias and variance, ensuring that the model performs well on unseen data. The final classification decision is made using a majority voting mechanism, where the most frequently predicted class label among all decision trees is chosen as the final output.

Step 5: Performance Evaluation and Real-World Application

The function calculates performance metrics to determine the model's effectiveness in palm print authentication. The accuracy of predictions, along with other metrics like precision and recall, helps in evaluating how well the ETC model distinguishes between different classes. This robust classification approach ensures that the system can be deployed in security-critical applications such as biometric authentication, banking security, and digital access control.

4.RESULTS AND DISCUSSION

4.1 Dataset Description

The palm print dataset consists of images collected from 20 different individuals, each representing a unique class. Each class corresponds to a specific person and contains multiple palm print images captured under various conditions. Each class (person) contains multiple palm print images captured from different angles and under varying lighting conditions to ensure variability and robustness in the dataset. The dataset aims to provide a diverse and representative collection of palm prints for model training and evaluation in PPASs. This dataset serves as the foundation for training machine learning models to recognize and authenticate individuals based on their unique palm print patterns.

4.2 Result analysis

Fig. 5 displays the confusion matrix of the Perceptron model, which is a table that visualizes the performance of the model by comparing the actual and predicted classes. Each cell in the matrix represents the number of instances where the actual class (rows) was predicted as the predicted class (columns) by the Perceptron model. The confusion matrix provides valuable insights into the model's performance across different classes, highlighting areas of accurate classification and potential misclassifications. This information is crucial for evaluating the model's effectiveness and identifying areas for improvement in the classification task.

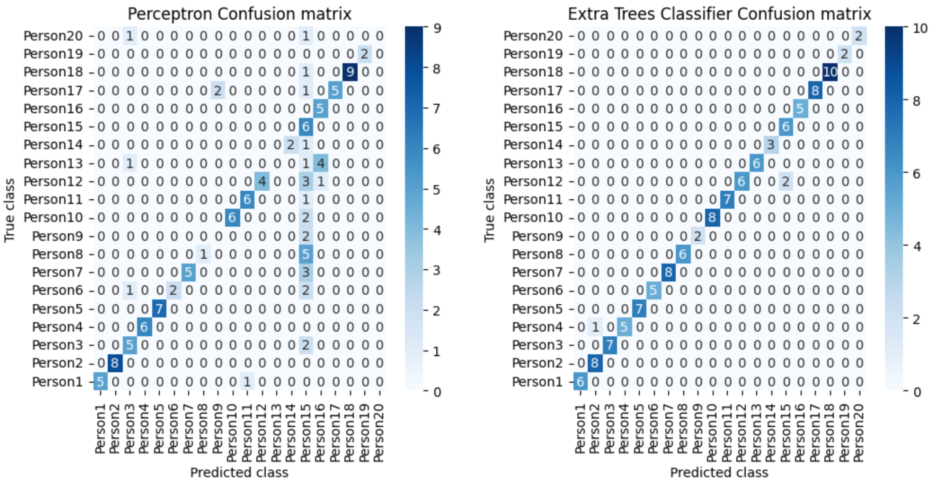

Fig. 5: (a) Confusion matrix of perceptron model. (b) Confusion matrix of ETC model.

Fig. 7 presents the confusion matrix of the Extra Trees Classifier (ETC) model. A confusion matrix is a table that visualizes the performance of a classification algorithm by comparing the actual and predicted classes. Each cell in the matrix represents the number of instances where the actual class (rows) was predicted as the predicted class (columns). The values in the matrix provide insights into the model's performance across different classes.

Table.1: Performance metrics table for the Perceptron and Extra Trees Classifier model.

Model	Accuracy	Precision	Recall	F1-score
Perceptron	70.0%	75.88%	65.19%	65.45%
Extra Trees Classifier	97.5%	98.19%	97.92%	97.82%

The Table-1 presents the performance metrics of two machine learning models: The Perceptron model and the Extra Trees Classifier (ETC) model. These models are used for palm print recognition in a security authentication system. Here's what each metric signifies:

Accuracy: The percentage of correctly classified instances out of the total instances in the dataset. Higher accuracy indicates better overall performance.

Precision: The ratio of correctly predicted positive observations to the total predicted positive observations. It measures the model's ability to avoid false positives. Higher precision indicates fewer false positives.

Recall: The ratio of correctly predicted positive observations to all actual positive observations in the dataset. It measures the model's ability to capture all positive instances. Higher recall indicates fewer false negatives.

F1-score: The harmonic means of precision and recall. It provides a balance between precision and recall and is useful when the classes are imbalanced. Higher F1-score indicates better overall performance.

5. CONCLUSION

The study demonstrates the development and evaluation of a Machine Learning-driven PPAS for security applications. The existing Perceptron model and the proposed Extra Trees Classifier (ETC) model are built and compared to assess their effectiveness in palm print recognition. The Perceptron model, while simple and efficient, is limited in its ability to handle complex, non-linear patterns in the data. It performs adequately for basic classification tasks but falls short in terms of accuracy and robustness when compared to more advanced models. The ETC model, on the other hand, leverages the power of ensemble learning to combine multiple decision trees. This approach significantly improves the model's accuracy and robustness, making it a more suitable choice for palm print authentication. The ETC model's superior performance is evident from the higher accuracy, precision, recall, and F1-scores obtained during the evaluation. The study highlights the importance of selecting appropriate ML models for biometric authentication tasks. The ETC model's ability to handle high-dimensional data and provide accurate predictions makes it a valuable tool for security applications. The practical demonstration of using the ETC model to predict identities from palm print images further underscores its potential for real-world deployment.

REFERENCES

[1] Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* **2020**, *9*, 1188.

[2] Ross, A.; Banerjee, S.; Chen, C.; Chowdhury, A.; Mirjalili, V.; Sharma, R.; Yadav, S. Some Research Problems in Biometrics: The Future Beckons. In Proceedings of the 2019 International Conference on Biometrics (ICB), Crete, Greece, 4–7 June 2019; pp. 1–8.

[3] Alausa, D.W.; Adetiba, E.; Badejo, J.A.; Davidson, I.E.; Obiyemi, O.; Buraimoh, E.; Oshin, O. Contactless Palmprint Recognition System: A Survey. *IEEE Access* **2022**, *10*, 132483–132505.

[4] Zhang, L.; Li, L.; Yang, A.; Shen, Y.; Yang, M. Towards Contactless Palmprint Recognition: A Novel Device, a New Benchmark, and a Collaborative Representation Based Identification Approach. *Pattern Recognit.* **2017**, *69*, 199–212.

[5] Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987.

[6] Kong, W.K.; Zhang, D.; Li, W. Palmprint Feature Extraction Using 2-D Gabor Filters. *Pattern Recognit.* **2003**, *36*, 2339–2347.

[7] Laadjel, M.; Kurugollu, F.; Bouridane, A.; Boussakta, S. Degraded Partial Palmprint Recognition for Forensic Investigations. In Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 1513–1516.

[8] Matkowski, W.M.; Chai, T.; Kong, A.W.K. Palmprint Recognition in Uncontrolled and Uncooperative Environment. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 1601–1615.

[9] Shao H, Zhong D, & Du X (2019). Efficient Deep Palmprint Recognition via Distilled Hashing Coding. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 714-723). IEEE. https://doi.org/10.1109/CVPRW.2019.00098

[10] Bensid K, Samai D, Laallam FZ, Meraoumia A (2018) Deep learning feature extraction for multispectral palmprint identification. Journal of Electronic Imaging 27(3):033018. https://doi.org/10.1117/1.JEI.27.3.033018

[11] Zhang D, Guo Z, Gong Y (2015) Multispectral biometrics: systems and applications. Springer

[12] Jing X Y, & Zhang D (2004). A face and palmprint recognition approach based on discriminant DCT feature extraction. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(6), 2405-2415. https://doi.org/10.1109/TSMCB.2004.837586

[13] Luo YT, Zhao LY, Zhang B, Jia W, Xue F, Lu JT, Zhu YH, Xu BQ (2016) Local line directional pattern for palmprint recognition. Pattern Recognition 50:26–44. https://doi.org/10.1016/j.patcog.2015.08.025

[14] Kang W, Liu Y, Wu Q, Yue X (2014) Contact-free palm-vein recognition based on local invariant features. PloS one 9(5):e97548. https://doi.org/10.1371/journal.pone.0097548

[15] Li S, & Kar P (2015). Context-aware bandits. arXiv preprint arXiv:1510.03164

[16] Mahadik K, Wu Q, Li S, & Sabne A (2020, June). Fast distributed bandits for online recommendation systems. In Proceedings of the 34th ACM international conference on supercomputing (pp. 1-13). https://doi.org/10.1145/3392717.3392748

[17] Li S, Karatzoglou A, & Gentile C (2016, July). Collaborative filtering bandits. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 539-548). https://doi.org/10.1145/2911451.2911548

[18] Wang X, Lei L, Wang M (2012) Palmprint verification based on 2D-Gabor wavelet and pulse-coupled neural network. Knowledge-Based Systems 27:451–455. https://doi.org/10.1016/j.knosys.2011.10.008

[19] Minaee S, & Wang Y (2016). Palmprint recognition using deep scattering convolutional network. arXiv preprint arXiv:1603.09027

[20] Svoboda J, Masci J, & Bronstein M M (2016, December). Palmprint recognition via discriminative index learning. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 4232-4237). IEEE. https://doi.org/10.1109/ICPR.2016.7900298

[21] Meraoumia A, Kadri F, Bendjenna H, Chitroub S, & Bouridane A (2017). Improving biometric identification performance using PCANet deep learning and multispectral palmprint. In Biometric security and privacy (pp. 51-69). Springer, Cham. https://doi.org/10.1007/978-3-319-47301-7_3

[22] Cheng J, Sun Q, Zhang J, & Zhang Q (2017, October). Supervised hashing with deep convolutional features for palmprint recognition. In Chinese Conference on Biometric Recognition (pp. 259-268). Springer, Cham. https://doi.org/10.1007/978-3-319-69923-3_28

[23] Genovese A, Piuri V, Plataniotis KN, Scotti F (2019) PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition. IEEE Transactions on Information Forensics and Security 14(12):3160–3174.

[24] Fei L, Xu Y, Zhang B, Fang X, Wen J (2016) Low-rank representation integrated with principal line distance for contactless palmprint recognition. Neurocomputing 218:264–275. https://doi.org/10.1016/j.neucom.2016.08.048

[25] Arora G, Kalra S, Bhatia A, Tiwari K (2021) PalmHashNet: Palmprint Hashing Network for Indexing Large Databases to Boost Identification. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3123291

[26] Zhao S, Zhang B (2020) Joint constrained least-square regression with deep convolutional feature for palmprint recognition. IEEE Transactions on Systems, Man, and Cybernetics: Systems. https://doi.org/10.1109/TSMC.2020.3003021