International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

VoteX: Secure Voting System Using Biometric Authentication with CNN-Based Fingerprint Liveness Detection

Mrs. Shraddha Raut[1], Ishika Mehere[2], Tanisha Sharma[3], Shravani Dahule[4], Shreyash Borkute[5], Shivani Dharamkar[6], Vrushali Ambulkar[7]

1 Assistant Professor, Department of Computer Science and Engineering (AI), G H Raisoni Skill Tech University Nagpur, Maharashtra, India

2,3,4,5,6,7 Research Scholar, Department of Computer Science and Engineering, G H Raisoni University, Amravati, Maharashtra, India

shraddha.raut@ghrstu.edu.in, ishikamehere@gmail.com, tanishasharma0130@gmail.com, shravanidahule@gmail.com, shreyashborkute4@gmail.com, shivanidharamkar2@gmail.com, vrushaliambulkar74@gmail.com

---

Abstract - Electronic voting systems require robust voter authentication and spoof prevention to ensure election integrity. This paper presents VoteX, a biometric electronic voting system that integrates a deep-learning fingerprint liveness detection module directly into the voting workflow. The proposed architecture pairs an ESP32-based terminal and optical fingerprint sensor with a Flask backend, SQLite database, and an EfficientNetB0 convolutional neural network for binary presentation attack detection. Fingerprint images are preprocessed into normalized regions of interest before liveness classification so that fake or artificial fingerprints can be rejected prior to identity verification and ballot casting. Held-out evaluation on 220 GreenBit test samples using decision threshold $\tau = 0.31$ achieved 85.45% accuracy, 85.00% spoof-class recall, 80.95% F1-score, 0.9256 ROC-AUC, 15.00% APCER, 14.29% BPCER, and 14.64% ACER. Furthermore, preliminary hardware deployment feasibility on 16 R307S sensor scans using threshold $\tau = 0.60$ (calibrated on those same 16 scans as a preliminary check) achieved 93.75% accuracy, 0 of 6 spoof presentations accepted (three PVA-glue replicas, two scans each), 10.00% BPCER, and 5.00% ACER. By embedding presentation attack detection into the biometric verification pipeline, VoteX screens for spoofed fingerprints as an added security layer, while duplicate voting is prevented by the has_voted database check without altering existing registration procedures.

Keywords: Biometric Authentication, Electronic Voting, Presentation Attack Detection, EfficientNetB0, ESP32, R307S.

1. INTRODUCTION

Electronic voting systems simplify vote recording, management, and counting compared with conventional paper-based procedures. However, their security depends on ensuring that only authorized voters participate and that each voter performs only the permitted voting operation. Authentication is therefore essential. While traditional methods use identification numbers, passwords, or cards, biometric authentication uses physiological characteristics such as fingerprints for individual recognition. However, fingerprint systems are vulnerable to presentation attacks involving artificial or manipulated fingerprints. Fingerprint Presentation Attack Detection (PAD), also known as liveness detection, distinguishes genuine presentations from attacks. ISO/IEC 30107-3 provides guidelines for evaluating and reporting PAD performance [3]. Deep learning, particularly Convolutional Neural Networks (CNNs), enables automatic feature extraction for fingerprint liveness detection. The proposed VoteX system integrates an EfficientNetB0-based binary classifier, leveraging compound scaling of network depth, width, and resolution for computational efficiency [4]. The model was trained using GreenBit fingerprint data for LIVE/SPOOF classification. Rather than replacing existing fingerprint registration and matching, the CNN acts as a security layer before authentication and voting. The workflow combines fingerprint acquisition, liveness detection, fingerprint matching, voter verification, voting-status checking, and vote recording.

2. LITERATURE REVIEW

Biometric authentication forms the security backbone of modern identity verification, yet physical presentation attacks remain a critical point of failure at the capture interface [1]. Jain et al. [1] discussed how automated matchers are vulnerable to direct spoofing, while Jain et al. [6] noted that template protection schemes cannot protect a system if the acquisition sensor is compromised. Maltoni et al. [2] detailed how optical capture platens can be deceived by artificial presentation attack instruments (PAIs) made of silicone, gelatin, and latex. To standardize evaluations, ISO/IEC 30107-1 [12] provides the framework and ISO/IEC 30107-3 [3] the testing and reporting metrics; APCER and BPCER are the error rates defined in ISO/IEC 30107-3, while ACER is their average, as used in the LivDet competitions.

To counter these threats, benchmark competitions like LivDet 2017 [5] reported the results of submitted spoof-detection algorithms, including CNN-based methods, on optical scanners. LivDet 2019 [7] extended this evaluation to liveness detection integrated with fingerprint matching systems. Tan and Le [4] showed that compound scaling enables EfficientNetB0 to achieve high feature discriminability with approximately 5.3 million parameters (about 4.0 million excluding the classification head). A known difficulty for PAD models is generalization to sensors and materials unseen in training, which motivates the sensor-level check on the R307S described in Section 3.D and reported in Section 4.C. To handle edge constraints, hardware specifications from Espressif Systems [8] indicate that the ESP32 microcontroller is limited to 520 KB of internal SRAM, precluding local

execution of large convolutional graphs. Consequently, our architecture uses an edge-to-server decoupled topology where the R307S sensor communicates with the ESP32 over UART [9] and the ESP32 sends the image to the Flask backend [10] over Wi-Fi via HTTP, and the backend runs TensorFlow and Keras [11] for real-time liveness verification.

Table 1: Authors and Findings

Ref.	Author(s) and Year	Focus Area / Study	Key Findings
[1]	Jain et al., 2004	Biometric Recognition	Discussed vulnerability of automated matchers to direct sensor spoofing.
[2]	Maltoni et al., 2009	Fingerprint Recognition	Analyzed FTIR optical capture vulnerabilities against artificial silicone and gelatin artifacts.
[3]	ISO/IEC, 2023	Presentation Attack Standards	Defined reporting metrics (APCER and BPCER) for PAD testing.
[4]	Tan & Le, 2019	EfficientNet Scaling	Showed compound scaling for high feature discriminability with fewer parameters.
[5]	Mura et al., 2018	LivDet 2017 Competition	Reported the results of the LivDet 2017 competition, in which CNN-based and other algorithms were evaluated on optical sensors including GreenBit data.
[6]	Jain et al., 2008	Template Security	Noted that template protection schemes cannot protect a system if the acquisition sensor is compromised.
[7]	Orrù et al., 2019	LivDet 2019 in Action	Reported the results of LivDet 2019, which evaluated liveness detection integrated with fingerprint matching systems.
[8]	Espressif Systems, 2023	ESP32 Datasheet	Documented hardware memory constraints (520 KB SRAM) for microcontrollers.
[9]	Adafruit Industries, 2022	R307S Sensor Library	Outlined low-level UART packet formats for optical fingerprint frame streaming.
[10]	Pallets, 2023	Flask Web Framework	Provided lightweight backend routing between hardware and AI models.
[11]	TensorFlow, 2023	Keras API Reference	Enabled deep learning forward inference pipelines for server-side execution.
[12]	ISO/IEC, 2016	PAD Framework	Formulated conceptual taxonomy and definitions for presentation attack detection.

3. METHODOLOGY

A. System Overview

VoteX consists of a fingerprint acquisition unit, an ESP32 voting terminal, a Flask backend, a CNN liveness detection module, a matching mechanism (performed in the Flask backend), and an SQLite database. The system provides voter registration, fingerprint matching, voting-status management, and vote recording. The GreenBit CNN is integrated directly into the authentication path without replacing existing registration workflows.

The proposed sequential pipeline begins with fingerprint capture using the R307S sensor and ROI preprocessing. The preprocessed image is then passed to the CNN-based liveness detection model to determine whether the fingerprint is live or a spoof presentation. If the fingerprint is classified as a spoof, the authentication process is stopped immediately. If it is classified as live, the existing fingerprint matching and voting workflow continues with voter verification, checking the has_voted status, and recording the vote.

B. Hardware & Software Architecture

The hardware comprises an ESP32 microcontroller and an R307S optical fingerprint sensor [8], [9]. The R307S sensor communicates with the ESP32 over UART, and the ESP32 sends captured image frames to the Flask backend over Wi-Fi via HTTP. The software stack uses Python/Flask for processing [10], TensorFlow/Keras for CNN inference [11], and SQLite for persistent storage. The trained model is loaded during backend initialization to minimize inference latency.

C. Dataset and Cross-Sensor Considerations

The fingerprint liveness detection model was trained and evaluated using the GreenBit subset of the LivDet Fingerprint 2017 data, obtained from the LivDet Joint Multimodal Biometric Dataset – Clarkson/Cagliari under a signed dataset release agreement (the LivDet 2017 competition is described in [5]). The images were re-partitioned by the authors into training, validation and test sets, so the 220-image test set is not the official LivDet 2017 test set. The data were split using the numeric identifier in each filename as a conservative grouping unit, so that no identifier appears in more than one split; subject-level separation could not be verified. The classifier evaluates two target classes: LIVE and SPOOF. All frames were preprocessed into region-of-interest patches to ensure spatial consistency.

Table 2: GreenBit Dataset Partitioning

Dataset Split	Live	Spoof	Total
Training	701	389	1,090
Validation	161	90	251
Testing	140	80	220
Total	1,002	559	1,561

D. R307S Sensor-Level Evaluation

To evaluate the behavior of the deployed liveness detection pipeline on the target acquisition hardware, additional

fingerprint captures were collected using the R307S optical fingerprint sensor [9]. The sensor-level evaluation consisted of 16 scan instances, including 10 genuine fingerprint presentations from 10 different fingers and 6 spoof presentations generated using three spoof fingerprint replicas, all fabricated from PVA glue, with two scans per replica. These samples were not used for model training or GreenBit validation/test evaluation. However, the same 16 scans were also used to calibrate the deployment threshold (Section 3.G); results should therefore be interpreted as a preliminary feasibility study rather than an independent test. The purpose of this experiment was to assess the behavior of the GreenBit-trained model under the image characteristics produced by the actual R307S acquisition device.

E. ROI Preprocessing Pipeline

Original GreenBit fingerprint images were captured as 360 × 400 grayscale BMP files, whereas R307S captures are 256 × 288 grayscale images. Foreground ROI extraction was applied to isolate the ridge pattern region; the ROI was then resized to 224 × 224 pixels using PIL LANCZOS resampling (PIL.Image.Resampling.LANCZOS) as implemented in the Flask backend and normalized to pixel values in [0, 1]. The model's first layer (a Rescaling layer) multiplies the [0, 1] input by 255, restoring the 0–255 range that EfficientNetB0 expects, resulting in an input tensor shape of 224 × 224 × 1.

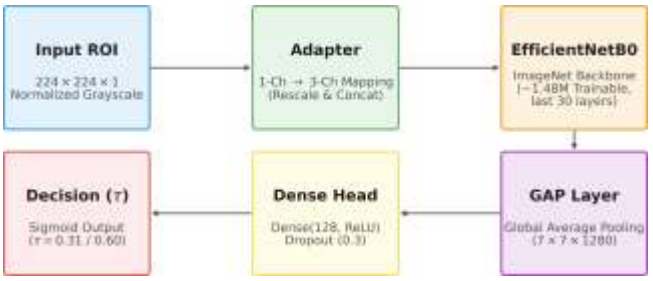


Fig. 1. EfficientNetB0 classification pipeline

F. CNN Architecture Details

The proposed liveness detector pairs an ImageNet-pretrained EfficientNetB0 backbone with a custom binary classification head. Grayscale inputs are duplicated across three channels, and the network sequentially applies Global Average Pooling, dropout (0.3), a 128-neuron dense layer, dropout (0.3), and a sigmoid output neuron. Training used the Adam optimizer and binary cross-entropy loss with a batch size of 32. In the first phase, the classification head was trained at a learning rate of 1e-4 while the backbone remained frozen. In the second phase, the last 30 layers were unfrozen and fine-tuned at a learning rate of 1e-5 while BatchNorm layers were kept frozen. Data augmentation included horizontal flipping, random rotation up to 10 degrees, zoom up to 10%, and brightness adjustments.

G. Model Training and Decision Thresholds

The model was trained using binary cross-entropy loss, with checkpointing monitoring validation loss to identify the optimal weights at the best validation checkpoint. The final sigmoid score corresponds to the estimated spoof probability. During the GreenBit validation procedure, a threshold of 0.31 was selected by minimizing the validation ACER. For deployment with the R307S sensor, an operating threshold of 0.60 was adopted based on the sensor-level evaluation. Under this deployment decision logic, an estimated spoof probability greater than or equal to 0.60 classifies the sample as a spoof presentation, whereas any probability strictly below 0.60 classifies it as a live capture. This deployment threshold was selected empirically to reduce false rejections of genuine R307S captures caused by cross-sensor domain shift; because it was calibrated on a small sample, it requires validation on larger, independent data.

Table 3: GreenBit Validation Performance At The Validation-Optimized Operating Point

Metric	Validation Result
Threshold	0.31
APCER	10.00%
BPCER	10.56%
ACER	10.28%

H. VoteX Workflow Integration

The integration couples physical terminal events directly with backend verification routines. The terminal captures the fingerprint image, and the ESP32 forwards it to the Flask server over Wi-Fi via HTTP. The server derives the normalized ROI, evaluates the CNN spoof score against the deployment threshold of 0.60, and halts execution upon spoof detection. Verified live presentations unlock the minutiae-based matching logic (executed in the Flask backend), inspect voter registration and the has_voted flag in SQLite, and present candidate choices upon successful validation. Once the voter submits their selection, the backend commits the ballot record, updates the voter status to preclude duplicate voting attempts, and logs the transaction.

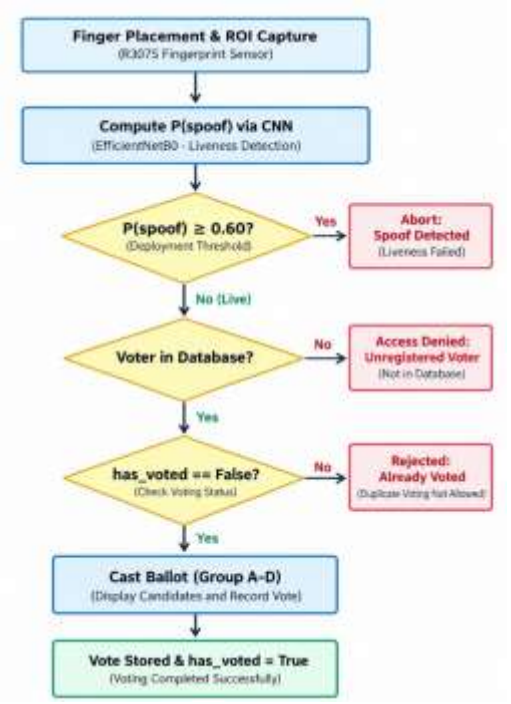


Fig. 2. Operational flowchart of liveness and voting logic

4. RESULTS

During fine-tuning, the best validation checkpoint reached 88.84% accuracy, 86.90% precision (spoof class), 81.11% recall (spoof class), and an ROC-AUC of 0.9535 at the default threshold of 0.5.

A. Outputs

The following interface screenshots capture the complete user journey, encompassing initial landing page interactions, administrative management controls, real-time authentication feedback, and graphical summaries of election results. Together, these views demonstrate the end-to-end operational workflow, highlighting intuitive voter guidance during verification alongside administrative oversight and real-time ballot tabulation.

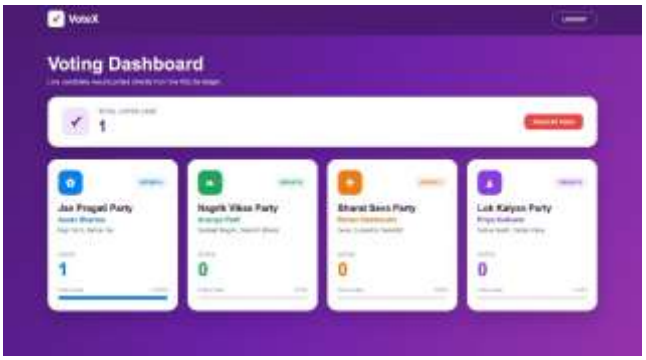


Fig. 3. Admin dashboard interface



Fig. 4. Home page interface



Fig. 5. Fingerprint verification interface

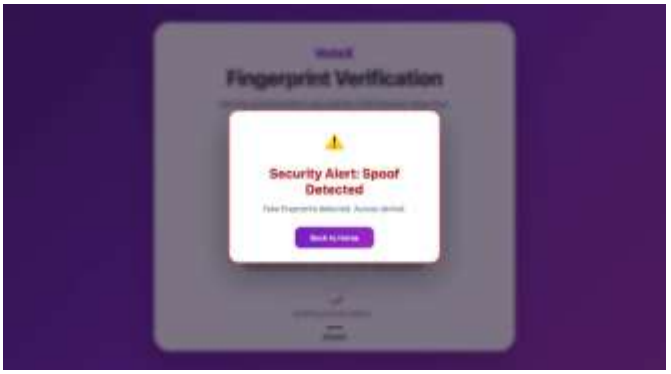


Fig. 6. Spoof attack alert dialog

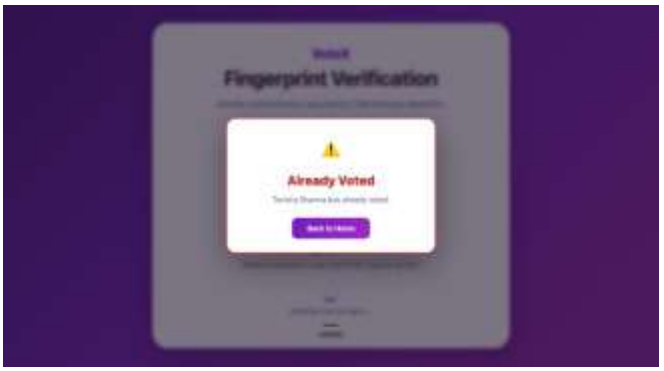


Fig. 7. Duplicate voting attempt alert dialog

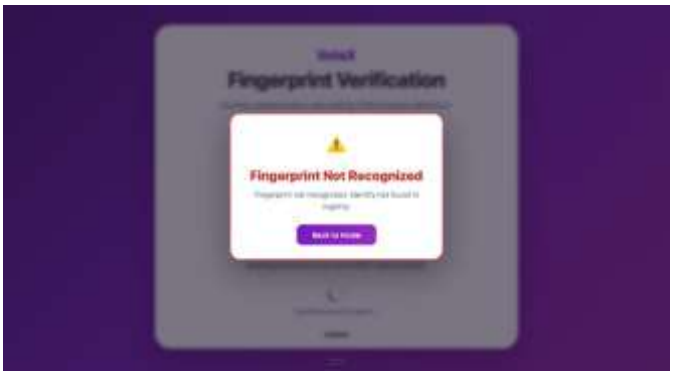


Fig. 8. Unregistered voter alert dialog

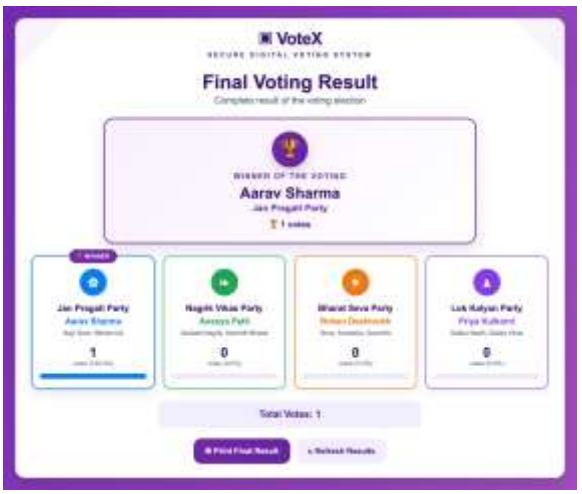


Fig. 9. Election results visualization interface

B. Test Set Results and Confusion Matrix

Final evaluation was executed on the 220 held-out GreenBit test samples using the validation-derived threshold of $\tau = 0.31$. Table 4 and Fig. 10 illustrate the confusion matrix for this GreenBit test set evaluation. Precision, recall, and F1-score are computed with SPOOF as the positive class, while Specificity measures the proportion of live samples correctly accepted.

Table 4: Test Set Confusion Matrix

	Predicted LIVE	Predicted SPOOF
Actual LIVE	120	20
Actual SPOOF	12	68

Table 5: Test Evaluation Metrics

Metric	Test Result
Accuracy	85.45%
Precision (Spoof)	77.27%
Recall (Spoof)	85.00%
F1-score (Spoof)	80.95%
Specificity (Live Accepted)	85.71%
ROC-AUC	0.9256

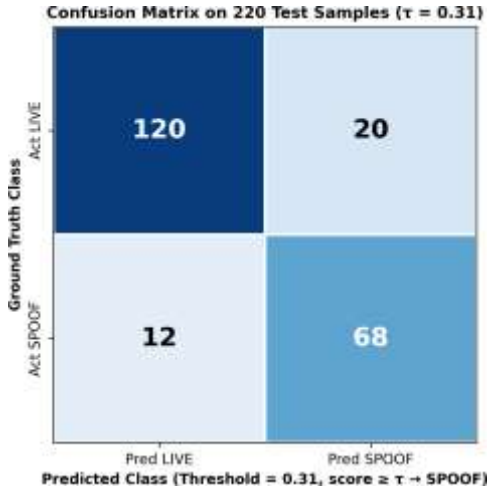


Fig. 10. Confusion matrix on 220 GreenBit test samples

C. R307S Sensor-Level Evaluation

To evaluate preliminary hardware feasibility on the target R307S acquisition hardware, 16 additional fingerprint scans were evaluated using the deployment threshold of 0.60, which was calibrated on those same 16 scans. The sensor-level dataset consisted of 10 genuine fingerprint presentations from 10 different fingers and 6 spoof presentations generated using three spoof fingerprint replicas, all fabricated from PVA glue, with two scans per replica. Nine of the ten genuine presentations were classified as LIVE, while all six spoof presentations were correctly classified as SPOOF. The resulting accuracy was 93.75%, with 0 of 6 spoof presentations accepted (three PVA-glue replicas, two scans each), BPCER of 10.00%, and ACER of 5.00%. Because these 16 scans were used to calibrate the threshold, these results represent a preliminary feasibility check rather than an independent test, and all spoofs were made from a single material (PVA glue), so this result may not generalize to other materials. Statistically, a result of 0 accepted spoofs out of 6 has a two-sided 95% Clopper-Pearson upper bound of approximately 46% on the true APCER, and because the six scans come from only three replicas, they are not independent observations.

Evaluating the impact of increasing the threshold to $\tau = 0.60$ on the GreenBit dataset highlights the trade-off between usability and security. On the GreenBit validation set, increasing the threshold from $\tau = 0.31$ to $\tau = 0.60$ changes APCER from 10.00% to 26.67%, BPCER from 10.56% to 4.35%, and ACER from 10.28% to 15.51%. On the GreenBit test set, $\tau = 0.60$ yields an APCER of 25.00%, BPCER of 6.43%, and ACER of 15.71%, compared to 15.00% APCER, 14.29% BPCER, and 14.64% ACER at $\tau = 0.31$. This higher threshold favors genuine-user usability over spoof rejection on GreenBit data, underscoring that the R307S deployment threshold requires validation on larger, independent datasets. Because roughly one in four GreenBit test spoofs is accepted at $\tau = 0.60$, the 0-of-6 result on the R307S scans should not be read as a near-zero attack success rate; six scans from three replicas cannot rule out a substantially higher acceptance rate.

The empirical findings suggest that integrating deep-learning liveness detection can help safeguard biometric electronic voting systems against presentation attacks, while duplicate voting is prevented by the has_voted database check. Model evaluation on the GreenBit dataset provides a baseline, while calibrated operating thresholds on the R307S optical sensor resulted in no spoof acceptance in the limited, single-material R307S evaluation collection.

5. DISCUSSION

The empirical findings from both held-out test evaluation and physical hardware integration highlight key insights into deploying deep-learning presentation attack detection on embedded voting terminals.

A notable observation is the operational divergence between held-out test evaluation and physical sensor deployment. On the held-out GreenBit dataset, the EfficientNetB0 classifier achieved 85.45% accuracy and an ACER of 14.64% at the validation-optimized cutoff ($\tau = 0.31$). However, transitioning the model to the physical R307S optical capture platen exposed a potential cross-sensor domain shift. Plausible hypotheses for this shift include optical light refraction, 4-bit pixel packing over UART serial channels, and platen noise inherent to frustrated total internal reflection (FTIR) capture, though these factors were not systematically tested.

Calibrating the operational threshold from $\tau = 0.31$ to $\tau = 0.60$ for hardware deployment provided an empirical trade-off between attack rejection and legitimate user access. At $\tau = 0.60$, 0 of 6 PVA glue spoof attempts were accepted on the physical evaluation collection prior to voter authentication. Although the 10.00% BPCER reflects a single false rejection among genuine presentations—hypothesized to be due to dry ridge impressions, though not verified—an overall ACER of

5.00% suggests that the calibrated cutoff achieved a favorable trade-off between attack rejection and genuine-user access on this limited sample. Furthermore, the decoupled edge-to-server topology allowed compute-intensive forward passes to run on the central backend, avoiding the memory and compute limits of the ESP32 microcontroller.

Despite these strengths, several operational constraints remain. The hardware evaluation was conducted on a curated collection of 16 physical scans used for both threshold calibration and feasibility testing; comprehensive electoral deployment necessitates testing across broader demographic cohorts, varied environmental humidity, and a wider range of spoof fabrication materials (e.g., silicone, gelatin, latex), since only PVA glue replicas were tested. Additionally, the architecture relies on local network availability, requiring future fault-tolerant transaction buffering to mitigate potential transmission dropouts. Further limitations include the centralized SQLite database, the need to specify a mechanism for separating voter identity from the ballot record while transaction logging takes place, and the privacy protection of stored biometric templates.

6. CONCLUSION

VoteX demonstrates the practical feasibility of embedding deep-learning-based Presentation Attack Detection into a physical biometric electronic voting workflow. By combining an ESP32 edge terminal, an R307S optical sensor, a Python Flask backend, and a fine-tuned EfficientNetB0 classifier, the system introduces an active liveness gate that screens for artificial replicas before template matching and SQLite database queries can occur.

The system was evaluated across two distinct operational settings. On the 220-sample GreenBit test set provided by the LivDet organizers, the classifier achieved 85.45% accuracy, 80.95% F1-score, 0.9256 ROC-AUC, and 14.64% ACER at an operating threshold of 0.31. When deployed on physical hardware at a calibrated operating cutoff of 0.60 (calibrated on those same 16 scans as a preliminary feasibility check), the pipeline achieved 93.75% accuracy and an ACER of 5.00% across 16 sensor-level fingerprint scans, with 0 of 6 PVA glue spoof presentations accepted. These preliminary results suggest the feasibility of a decoupled client-server architecture for supporting presentation attack detection on commercial hardware without altering existing voter registry mechanisms, though larger independent validation remains required. Duplicate voting is prevented separately via database status checks.

Future research will focus on porting quantized liveness models directly onto edge microcontrollers using TensorFlow Lite Micro to eliminate network dependencies, integrating multi-modal biometric fusion to mitigate ridge-abrasion false rejections, and adopting cryptographic zero-knowledge proofs to decouple voter authentication from ballot tabulation for privacy preservation.

ACKNOWLEDGEMENT

The authors express gratitude to their institution, faculty members, and project guides for guidance throughout the VoteX project. Acknowledgments are also extended to the organizers of the LivDet competitions, the Clarkson University and University of Cagliari teams for providing the LivDet Joint Multimodal Biometric Dataset, and the biometric research community for providing accessible benchmarking datasets.

REFERENCES

[1] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 4–20, 2004.
[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, 2nd ed. London, U.K.: Springer, 2009.
[3] International Organization for Standardization, ISO/IEC 30107-3: Information Technology—Biometric Presentation Attack Detection—Part 3: Testing and Reporting, 2nd ed., ISO/IEC, 2023.
[4] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 6105–6114, 2019.
[5] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis, “LivDet 2017 Fingerprint Liveness Detection Competition 2017,” in Proc. 11th IAPR Int. Conf. on Biometrics (ICB), 2018.
[6] A. K. Jain, K. Nandakumar, and A. Nagar, “Biometric template security,” EURASIP Journal on Advances in Signal Processing, vol. 2008, p. 579416, 2008.
[7] G. Orrù, R. Casula, P. Tuveri, C. Bazzoni, G. Dessalvi, M. Micheletto, L. Ghiani, and G. L. Marcialis, “LivDet in Action—Fingerprint Liveness Detection Competition 2019,” in Proc. Int. Conf. on Biometrics (ICB), 2019.
[8] Espressif Systems, ESP32 Series Datasheet, Espressif Systems, 2023.
[9] Adafruit Industries, Adafruit Fingerprint Sensor Library Documentation, 2022.
[10] Pallets, Flask Documentation (v3.x), Python Web Framework Documentation, 2023.
[11] TensorFlow, Keras Documentation and API Reference, TensorFlow Documentation, 2023.
[12] International Organization for Standardization, ISO/IEC 30107-1: Biometric Presentation Attack Detection—Part 1: Framework, ISO/IEC, 2016.