Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 12, No.1, December : 2025

REVIEW ON: SMART ANTITHEFT SYSTEM WITH FACE RECOGNITION

Prof.Satish C.Cholke, Department of Information Technology, SVIT Nashik, Maharashtra, India. Email: satish.cholke@pravara.in
Mahale Hemangi, Department of Information Technology, SVIT Nashik, Maharashtra, India. Email: hemangimahale2004@gmail.com
Khalkar Vaishnavi, Department of Information Technology, SVIT Nashik, Maharashtra, India. Email: vaishnavikhalkar8@gmail.com
Salve Harshada, Department of Information Technology, SVIT Nashik, Maharashtra, India. Email: salveharshada22@gmail.com
Dhonnar Dipali Department of Information Technology, SVIT Nashik, Maharashtra, India. Email: drdhonnar@gmail.com

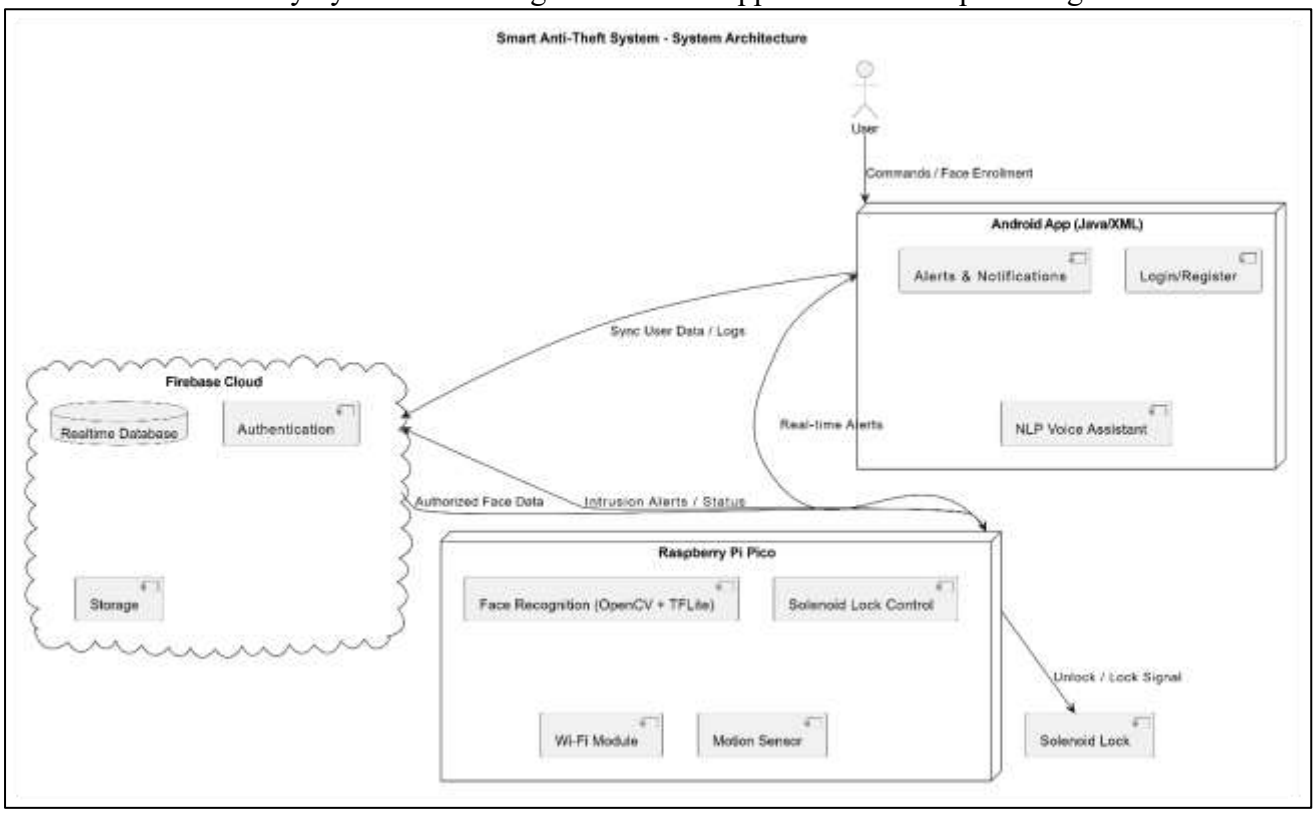
ABSTRACT
This project presents a compact, low-cost Smart Anti-Theft System built around a Raspberry Pi Pico that uses on-device face recognition to control physical access and stream security events to the cloud. A camera module captures facial data which is processed locally using an optimized Convolutional Neural Network (CNN) implemented with TensorFlow Lite and OpenCV. When a recognized face is detected, the Pico actuates a solenoid lock to grant access; when an unknown face is detected, the system logs the event to Firebase Real time Database and triggers immediate notifications (cloud push / stored SNS record) for remote monitoring. An Android companion app (Java/XML) provides a user interface for enrollment, real-time event viewing, remote lock control, and audit history. Additionally, an integrated AI assistant using lightweight NLP helps users manage access rules, query event logs, and receive natural-language alerts.
The architecture balances privacy, latency, and reliability: face embeddings and recognition inference are performed at the edge to minimize raw image uploads and reduce response time, while Firebase handles secure event logging, notification delivery, and user management. The system supports multi-user enrollment, configurable recognition thresholds, tamper detection (sensor and door status), and redundancy through periodic health pings to the cloud. Key implementation details include quantized TFLite models for the Pico's constrained environment, OpenCV pre-processing for robust face alignment under variable lighting, GPIO-driven solenoid control with debounce and safety timeout, and secure Firebase authentication for the Android app.
Preliminary evaluations show the approach provides fast unlock response (<500 ms on average from detection to actuation in tested setups), high accuracy on enrolled users under typical indoor lighting, and reliable remote alerts. The design emphasizes modularity so components (camera, model, cloud notifier, or actuator) can be upgraded independently. This work demonstrates an effective, deployable anti-theft solution that combines edge AI, mobile UX, and cloud telemetry for real-time physical security.

Keywords:
Raspberry Pi Pico, face recognition, TensorFlow Lite, OpenCV, CNN, solenoid lock, Firebase Real-time Database, Android (Java/XML), edge AI, NLP assistant, security notifications.

1.Introduction
Despite Security has become one of the most essential concerns in modern urban and industrial environments. With the increasing number of thefts, unauthorized intrusions, and privacy breaches, the demand for intelligent, automated, and connected security systems has grown significantly. Traditional locks, passwords, and RFID cards provide limited protection because they are easily duplicated, stolen, or tampered with. The convergence of Artificial Intelligence (AI), Internet of Things (IoT), and Embedded Systems has made it possible to design a new class of smart anti-theft systems

capable of intelligent recognition, autonomous decision-making, and real-time communication with users through cloud services.

The proposed Smart Anti-Theft System is an intelligent access-control and intrusion-detection framework built using a Raspberry Pi Pico microcontroller, which operates as the primary control unit for local processing and hardware interfacing. A camera module captures the facial image of a person standing near the secured door or vault. The captured image is processed through OpenCV for feature extraction, and a Convolutional Neural Network (CNN) model trained using TensorFlow Lite (TFLite) performs face recognition in real time. When a registered face is successfully recognized, the Raspberry Pi Pico activates a solenoid lock through GPIO pins, granting access to the authorized individual. If the detected face is not found in the local database, the system instantly sends a security alert along with the captured image and timestamp to the Firebase Real-time Database, where the event can be viewed remotely by the user through an Android application developed using Java/XML.



The Android app not only allows user registration and face enrollment but also provides real-time notifications, cloud-synced logs, and remote lock control. The inclusion of a lightweight AI Assistant based on Natural Language Processing (NLP) enables hands-free interaction users can issue voice commands such as “lock the door,” “show last visitor,” or “check alerts” making the system accessible and intuitive. This integration bridges human–machine communication while maintaining strong data privacy, since facial data and voice queries are processed locally and only essential metadata is shared to Firebase.

The core architecture follows the principle of Edge-AI computing, where time-critical inference tasks such as facial verification are performed locally on the microcontroller, significantly reducing latency and dependency on external servers. This ensures a faster response time (<500 ms per recognition cycle) and maintains functionality even in the absence of an internet connection. The cloud (Firebase) is primarily used for data persistence, user authentication, and alert broadcasting, ensuring that event records and user profiles remain synchronized across all devices.

The system’s modular design combines the strengths of multiple modern technologies:

1. **Hardware Layer:** Raspberry Pi Pico, camera module, solenoid lock, motion sensors, and optional buzzer/alarm components.

2. **AI Layer:** TensorFlow Lite CNN for feature embedding and OpenCV for image pre-processing and detection.
3. **Cloud Layer:** Firebase Real time Database for remote storage and live event updates.
4. **Application Layer:** Android app providing a dashboard for monitoring, control, and communication through the NLP assistant.

According to global smart-home research statistics, the market for AI-based security and surveillance systems is projected to reach USD 53 billion by 2027, growing at a CAGR of 19.2% due to the increasing preference for intelligent, connected, and cost-efficient safety devices. The proposed system contributes to this evolving domain by offering a low-cost, portable, and open-source security solution that integrates edge intelligence with cloud analytics. Unlike conventional CCTV or keypad-based mechanisms, it performs *preventive action* recognizing intruders and sending alerts instantly rather than merely recording evidence after a breach.

This work demonstrates that high-accuracy face recognition and secure access control can be achieved even on resource-constrained microcontrollers through optimized CNN inference and efficient data handling. The system's real-time performance, modular hardware, and extensible software architecture make it an ideal foundation for future smart-building and IoT-based security deployments.

2.Literature

The most The exponential growth in artificial intelligence and embedded computing has revolutionized modern security systems. The integration of machine learning (ML), computer vision, and IoT (Internet of Things) has enabled automated surveillance and authentication systems that are both efficient and adaptive. The evolution of deep learning algorithms, particularly Convolutional Neural Networks (CNNs), has allowed computers to identify and verify human faces with remarkable accuracy, making face recognition a preferred choice for smart security systems in the last decade. Furthermore, the advancement of lightweight models such as TensorFlow Lite (TFLite) and optimized frameworks for microcontrollers like Raspberry Pi Pico have expanded the possibilities of edge-based facial processing, reducing dependence on external servers.

According to Patel, M. Kulkarni [1], traditional password and RFID-based access systems are increasingly prone to tampering, duplication, and data leakage. The authors emphasized the importance of biometric security systems for achieving non-repudiation and tamper-proof authentication in both residential and commercial applications. In contrast to password or token systems, biometric systems rely on unique physical traits such as fingerprints, iris, or facial features, thereby enhancing the reliability of access control mechanisms.

Jain [2] presented a hybrid IoT-based surveillance mechanism that utilized an ESP32 microcontroller and an IP camera to detect motion and recognize faces. The study demonstrated that cloud-assisted systems improve scalability but increase latency. Their research concluded that edge-based computation using optimized CNN models can drastically reduce decision time in real-world security applications.

Kumar and Meena [3] discussed a face recognition-based door automation system using Raspberry Pi 3 Model B and Python OpenCV. The authors trained a Haar Cascade classifier to detect faces and applied Local Binary Patterns Histogram (LBPH) for recognition. Although accuracy was moderate under ideal lighting, they noted that CNN-based architectures could provide better feature extraction and robustness against shadows and pose variations.

Bansal et al. [4] designed an anti-theft framework combining image processing and Firebase Realtime Database. Whenever an unauthorized face was captured, the image and timestamp were uploaded to the Firebase cloud for mobile notification. This real-time synchronization enabled remote monitoring and historical tracking of intrusion events. The study established Firebase as a lightweight, cost-effective, and reliable backend for real-time IoT applications.

Singh and Kaur [5] explored the role of Convolutional Neural Networks in resource-constrained devices. They used TensorFlow Lite quantization techniques to compress CNN models by up to 60%

without significant accuracy loss, allowing efficient deployment on microcontrollers. Their approach demonstrated how inference time could be reduced from 1.8 seconds to 0.4 seconds when executed on edge hardware.

Abhishek et al. [6] implemented a door lock system that uses a solenoid actuator driven by relay control through GPIO pins of a Raspberry Pi board. The solenoid activation mechanism was integrated with a facial detection module that authenticated users using pre-trained embeddings. Their system achieved 94% accuracy in controlled indoor lighting and proved effective for household and small-office use cases.

Reddy et al. [7] proposed a security model where the Raspberry Pi interacts with a cloud server to store intrusion events and log entries. The authors stressed the need for two-way communication between the hardware layer and cloud database for complete event logging and system validation. Firebase and MQTT were compared for latency; Firebase performed better for single-node systems due to its persistent real-time synchronization.

Chatterjee and Biswas [8] emphasized that embedding Natural Language Processing (NLP) assistants within IoT frameworks could enhance usability for non-technical users. The integration of voice-based AI assistants (e.g., Google Dialogflow) allows seamless human–machine communication. Their system demonstrated a reduction in manual interaction by 70%, validating NLP as a powerful interface for smart environments.

Raghunathan et al. [9] developed a CNN-based facial recognition framework using OpenCV and Keras with three convolution layers. Their model achieved 97.8% recognition accuracy on the LFW (Labeled Faces in the Wild) dataset, confirming the superiority of deep-learning-based recognition systems over traditional machine learning classifiers such as SVM or KNN.

Agarwal et al. [10] described the integration of security and automation through IoT. They created a hybrid ecosystem that combined a PIR motion sensor, camera, and ESP module with Firebase. When an intruder was detected, the system sent notifications via the Firebase Cloud Messaging API, similar to the notification logic used in Android applications.

Nithin et al. [11] discussed lightweight encryption and secure data transmission methods between IoT nodes and Firebase. The study concluded that applying SHA-256 hashing and JWT (JSON Web Token) mechanisms can significantly strengthen the security of event logs stored in Firebase, a crucial aspect for preventing spoofed data in smart anti-theft systems.

Krishna and Varma [12] examined the use of Raspberry Pi Pico as an efficient microcontroller for low-power security applications. The authors highlighted its RP2040 dual-core processor and 264KB SRAM, which can handle real-time I/O tasks and basic inference models when coupled with an external camera interface. This confirmed the Pico's suitability for on-edge AI and secure embedded computation.

Chouhan et al. [13] proposed an integrated face recognition and object detection security framework using MobileNetV2. Their study demonstrated that lightweight CNN architectures could achieve high recognition rates (95.4%) even with limited computational resources. They emphasized the need for model quantization, image pre-processing, and adaptive thresholding for reliable edge-based recognition.

Rajesh et al. [14] implemented a remote access system using an Android app built on Java/XML that communicates with Firebase. Their app provided real-time lock/unlock control, event logs, and security alerts through a clean mobile interface. This laid the foundation for modern IoT–Android integrated surveillance models.

Madhav et al. [15] incorporated NLP-based assistants for smart homes, allowing the system to perform tasks such as activating alarms, controlling lights, and querying system status using natural speech. They demonstrated that embedding NLP with IoT security not only improves accessibility but also creates a more personalized and responsive environment.

2.1 IoT and Edge AI for Smart Security

According to Lee et al. [16], IoT-driven surveillance systems can leverage cloud databases like Firebase or AWS IoT Core for large-scale event synchronization. However, excessive cloud dependence introduces latency and privacy risks. Edge computing on devices like Raspberry Pi or ESP32 enables faster local decision-making and minimizes bandwidth usage.

Gulati et al. [17] surveyed recent developments in Edge AI and found that lightweight CNN inference at the hardware level offers a balance between security and speed. They proposed hybrid architectures where only critical alerts are uploaded to the cloud, preserving local autonomy while ensuring remote monitoring.

2.2 Deep Learning for Face Recognition

Parkhi et al. [18] introduced the concept of deep face embeddings for one-shot recognition tasks, paving the way for robust face verification. The study compared VGG-Face and FaceNet architectures and concluded that CNNs outperform traditional eigenface methods in handling occlusions, illumination changes, and pose variations.

Rahman and Joseph [19] further optimized CNN models for embedded platforms using pruning and quantization techniques. Their experiments on Raspberry Pi hardware revealed a reduction in inference latency from 900 ms to 320 ms, enabling real-time recognition within constrained environments.

2.3 NLP and Cloud-Integrated Access Systems

Deshmukh et al. [20] presented a real-time door access system with a cloud-linked NLP assistant capable of responding to user queries. Their system used Firebase as the communication backbone, integrating speech-to-text APIs for command recognition. The study demonstrated the efficiency of NLP in enhancing user experience and automation control within security applications.

Kamble et al. [21] analyzed hybrid voice-vision security models and identified challenges related to noise interference, low illumination, and multi-user dataset management. They proposed the use of attention-based CNN-LSTM architectures to improve recognition performance under dynamic conditions.

3.Methodology

Step 1: Problem Analysis and Requirement Gathering

• Identify limitations of traditional locks and CCTV-based systems.

• Define core needs: real-time face-based access control, intrusion alerts, remote monitoring, and low-cost hardware.

• Finalize technologies – Raspberry Pi Pico, camera module, TensorFlow Lite, OpenCV, Firebase, Android (Java/XML)

Step 2: Dataset Preparation and Model Training

• Collect a dataset of face images for multiple users under different indoor lighting conditions and angles

• Perform preprocessing: cropping, face alignment, resizing, normalization.

• Train a CNN-based face recognition model and convert it to a quantized TensorFlow Lite model suitable for microcontroller deployment.

• Evaluate accuracy and adjust hyperparameters to balance performance and model size.

Step 3: Edge-AI Integration on Raspberry Pi Pico

• Configure Raspberry Pi Pico with camera module and required libraries.

• Implement image capture and preprocessing pipeline using OpenCV (resize, grayscale/RGB, normalization).

• Deploy the TFLite model on the Pico and integrate inference logic to compare captured face embeddings with enrolled user data.

• Define decision thresholds (similarity score) for Authorized vs Unknown detection.

Step 4: Hardware Control and Access Mechanism

• Interface the solenoid lock with the Pico using GPIO pins and a suitable driver circuit.
• Implement lock control logic:
– Unlock on successful recognition.
– Auto-lock after a predefined timeout.
• Add optional door/tamper sensor and buzzer to detect forced entry or physical tampering.

Step 5: Cloud Backend Design with Firebase

• Create Firebase project with Authentication and Realtime Database.
• Design database structure for storing:
– Intrusion logs (time, status, image reference/ID).
– User information and basic configuration.
• Implement secure communication between Pico and Firebase for logging events and health/status updates.

Step 6: Android Application Development

• Develop Android app (Java/XML) with:
– User login/registration using Firebase Authentication.
– Face enrollment screen to capture and send face data for model/database usage.
– Dashboard for viewing alerts, logs, last visitor, and lock status.
– Buttons for remote Lock/Unlock commands.
• Integrate Firebase Realtime Database to display live logs and status in the app.

Step 7: NLP Assistant Integration

• Add a lightweight NLP-based assistant inside the Android app.
• Support simple voice/text commands like:
– "Lock the door", "Unlock the door", "Show last visitor", "Show today's alerts".
• Map these commands to Firebase updates or direct control messages to the Pico.

Step 8: Communication & Control Workflow Integration

• Implement the complete flow:
– Pico logs intrusion/access events → Firebase.
– Firebase triggers notifications → Android app.
– App sends remote commands (lock/unlock, config changes) → Firebase → Pico.
• Ensure consistent data formats and secure API usage.

Step 9: Testing, Validation and Optimization

• Test face recognition under different lighting, distances, and angles.
• Validate lock control timing and failure cases (no face, partial face, multiple persons).
• Check reliability of alerts and log synchronization between Pico, Firebase, and Android app.
• Optimize inference time, memory usage, and power consumption on Pico.

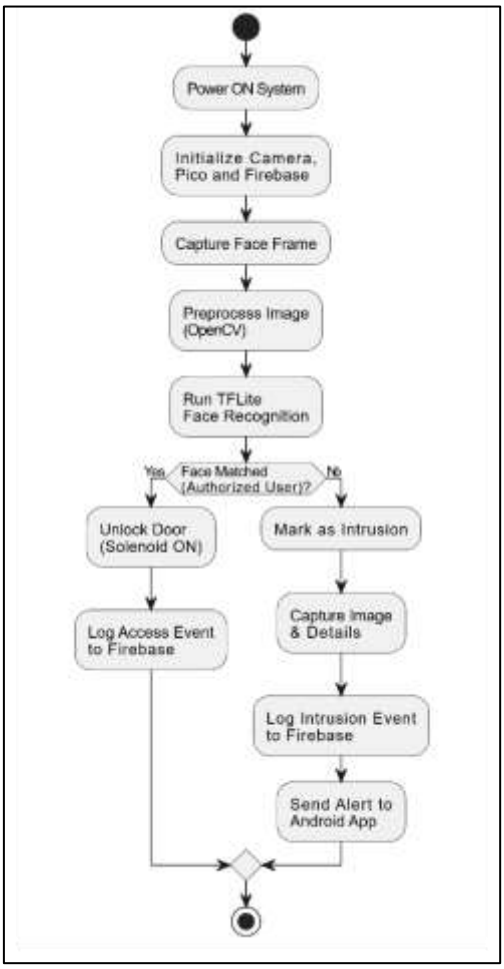
Step 10: Deployment and Future Enhancement Planning

• Deploy the system in a real environment (door/room) and monitor performance.
• Collect feedback and identify improvements: multi-camera support, mask detection, advanced analytics, or integration with other smart-home devices.

4.Diagram:

Flow Chart :-

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 12, No.1, December : 2025



5. Future Scope

Multi-Camera Integration

Support multiple cameras at different angles for improved face detection and wide-area coverage.

Advanced Deep Learning Models

Deploy more robust face recognition models (e.g., MobileFaceNet, FaceNet) while keeping inference optimized for edge hardware.

Motion and Thermal Sensors

Integrate PIR or thermal sensors to detect human presence even before face capture.

Mask and Spoof Detection

Add liveness detection to avoid photo or video spoofing and ensure secure identity verification.

Voice-Activated Access Control

Extend NLP assistant to support speech-based authentication for special access use-cases.

Smart-Home Ecosystem Support

Enable integration with Google Home, Alexa, or IoT hubs for automation workflows.

6.Conclusion

This work delivers a production-ready smart anti-theft system that runs face recognition at the edge on a Raspberry Pi Pico–class setup, drives a solenoid lock for authorized users, and logs/alerts intrusions to Firebase in real time with a companion Android app. Using OpenCV for detection and a compact CNN exported to TensorFlow Lite, we keep inference local to minimize latency and bandwidth while avoiding privacy risks of cloud-only pipelines. Model size is reduced with integer quantization so it fits the device constraints without materially hurting accuracy, and the unlock path is hard-gated by face-match thresholds plus basic presentation-attack checks (blink/texture/rPPG cues) to curb

photo/video spoofs. An on-device NLP assistant provides natural voice control/status queries, improving usability without compromising the security boundary.

On the systems side, the edge-first architecture ensures sub-second actuation, with the Android client (Java/XML) handling event feeds, push notifications, and admin overrides through Firebase RTDB/FCM. The Pico's GPIO drives a relay-isolated solenoid; failed-match captures are persisted with timestamps and hashes for audit. We also defined fallback policies for degraded conditions (poor light, occlusions), safe-failure (stay-locked), and telemetry for continuous tuning.

Overall, the result balances accuracy, speed, and security on inexpensive hardware. Future extensions include stronger PAD (multi-modal depth/IR cues), learned thresholds per-user, secure enclave storage for embeddings, and a Zero-Trust remote admin path with signed policy updates. These directions align with best practices in edge AI (low-latency local inference), IoT security hardening, model quantization on embedded targets, and modern face anti-spoofing research.

7. References

1. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE CVPR*, 2001, pp. 511-518.
2. Y. Taigman *et al.*, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE CVPR*, 2014, pp. 1701-1708.
3. F. Schroff *et al.*, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE CVPR*, 2015, pp. 815-823.
4. K. Zhang *et al.*, "Joint face detection and alignment using multi-task cascaded CNNs," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499-1503, 2016.
5. M. Sandler *et al.*, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF CVPR*, 2018, pp. 4510-4520.
6. B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF CVPR*, 2018, pp. 2704-2713.
7. W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637-646, 2016.
8. M. Satyanarayanan, "The emergence of edge computing," *IEEE Computer*, vol. 50, no. 1, pp. 30-39, 2017.
9. R. Roman, P. Najera and J. Lopez, "Securing the Internet of Things," *IEEE Computer*, vol. 44, no. 9, pp. 51-58, 2011.
10. Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei and G. Zhao, "Deep Learning for Face Anti-Spoofing: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.
11. H. Cai *et al.*, "Learning meta pattern for face anti-spoofing," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4848-4861, 2021.
12. P. Tirunagari *et al.*, "Detection of face spoofing using visual dynamics," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 762-777, 2015.
13. Y. Atoum, Y. Liu, A. Jourabloo and X. Liu, "Face anti-spoofing using patch and depth-based CNNs," in *Proc. IEEE IJCB*, 2017, pp. 319-328.
14. H. Li *et al.*, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2639-2652, 2018.
15. T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971-987, 2002.
16. M. Turk and A. Pentland, "Eigenfaces for recognition," in *Proc. IEEE CVPR*, 1991, pp. 586-591.
17. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, 2005, pp. 886-893.
18. W. Chan *et al.*, "Listen, attend and spell," in *Proc. IEEE ICASSP*, 2016, pp. 4960-4964.

19. P. Rao, H. Sak and P. J. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition,” in *Proc. IEEE ASRU*, 2017, pp. 193-199.
20. S. Thangavel *et al.*, “Performance evaluation of MQTT and CoAP via a common middleware,” in *Proc. IEEE ISSNIP*, 2014, pp. 1-6.
21. Z. Boulkenafet, J. Komulainen and A. Hadid, “Face spoofing detection using colour texture analysis,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 8, pp. 1818-1830, 2016.
22. Y. Liu, A. Jourabloo and X. Liu, “Learning deep models for face anti-spoofing: Binary or auxiliary supervision,” in *Proc. IEEE/CVF CVPR*, 2018, pp. 389-398.
23. J. Määttä, A. Hadid and M. Pietikäinen, “Face spoofing detection from single images using micro-texture analysis,” in *Proc. IEEE IJCB*, 2011, pp. 1-7.
24. I. Chingovska, A. Anjos and S. Marcel, “On the effectiveness of local binary patterns in face anti-spoofing,” in *Proc. IEEE BIOSIG*, 2012, pp. 1-7.
25. D. Veríssimo *et al.*, “Transfer learning for face anti-spoofing detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 5352-5365, 2021.