
WWJMRD 2025; 11(04): 51-58
www.wwjmrd.com
*International Journal*
*Peer Reviewed Journal*
*Refereed Journal*
*Indexed Journal*
*Impact Factor SJIF 2017: 5.182 2018: 5.51, (ISI) 2020-2021: 1.361*
*E-ISSN: 2454-6615*

**Sudip Chakraborty**
D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore, India.

**Deep Chakraborty**
MCKV Institute of Engineering, Howrah, West Bengal, India.

**Correspondence:**
**Sudip Chakraborty**
D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore, India.

# Smart Door Automation System with DeepFace and SLMs

## Sudip Chakraborty, Deep Chakraborty

**Abstract**
The continuous evolution of smart home technologies has underscored the necessity for secure, efficient, and user-friendly solutions in residential automation. Addressing these requirements, we introduce an advanced Smart Door Automation System, innovatively integrating DeepFace facial recognition technology with the compact and effective language model, TinyLLaMA. This integration aims to provide residents with reliable security and personalized interactions, transforming routine home entry into an intuitive, seamless experience. This innovative system leverages standard CCTV camera feeds to accurately and rapidly detect authorized users, thereby significantly strengthening home security without interrupting daily routines. DeepFace, renowned for its robust facial recognition capabilities, adeptly handles variations in environmental lighting, facial orientations, and minor changes in user appearance, consistently ensuring reliable and accurate identification. By conducting both the facial recognition and linguistic processing directly on the local device, our system considerably reduces response times, eliminates potential network latency, and enhances data privacy by minimizing cloud dependency. Upon positive user recognition, TinyLLaMA swiftly generates personalized, context-specific greetings, which are instantly transformed into clear, natural-sounding audio messages through Google Text-to-Speech (GTTS). The combined efficacy of DeepFace and TinyLLaMA has been validated through extensive testing, demonstrating a notable reduction in unauthorized access attempts alongside improved user engagement and satisfaction. Overall, this comprehensive approach illustrates the practical application of advanced artificial intelligence methods, significantly raising the standards of everyday smart home security and automation.

**Keywords:** Smart Door Automation, DeepFace, Small Language Model, TinyLLaMA, Face Recognition, Real-Time Interaction, Embedded AI.

## Introduction
The rapid advancement of smart home technology has increasingly shifted focus toward intelligent and adaptive solutions for residential security and convenience. One significant application area is the automation of door systems, where conventional methods such as physical keys or basic electronic locks fall short in adaptability and ease of use. These traditional approaches not only present practical inconveniences but also raise security concerns due to their vulnerability to loss, theft, or unauthorized duplication. Additionally, reliance on cloud-based access control solutions introduces challenges related to network latency, security risks, and privacy concerns, prompting a shift towards locally hosted intelligence.

To address these limitations comprehensively, this study introduces an advanced Smart Door Automation System specifically utilizing the DeepFace facial recognition algorithm due to its exceptional accuracy and robustness. DeepFace leverages a sophisticated convolutional neural network (CNN) architecture that generates highly distinctive facial embeddings, effectively handling variations in environmental factors such as illumination, facial orientation, and minor changes in personal appearance. This precise and reliable identification significantly reduces false positives and false negatives, ensuring authorized access with minimal delay. Integrating DeepFace with a Small Language Model (SLM), specifically TinyLLaMA, further enriches user interaction and experience. Upon successful

recognition of the homeowner, TinyLLaMA generates personalized greetings based on contextual data such as the time of day or recent events. These contextually relevant greetings enhance the user's sense of comfort and engagement with the automated system. To deliver these greetings effectively, the system employs Google Text-to-Speech (GTTS), renowned for its clarity and natural sound quality. This immediate auditory feedback not only confirms successful identity verification but also provides an intuitive, welcoming interaction, significantly elevating user satisfaction.

Moreover, coupling this integrated facial recognition and greeting system with gate control via a static IP ensures seamless and secure physical access. The local network-based communication minimizes latency, ensuring near-instantaneous responses. The approach avoids reliance on external cloud services, thereby reducing potential security risks and enhancing user privacy. By synthesizing precise facial identification, dynamic linguistic interaction, and robust IoT-enabled gate control, our Smart Door Automation System represents a sophisticated, cohesive solution that substantially advances residential automation technology in both functionality and user experience.

## 2. Literature Review

Facial recognition technologies have been extensively researched and applied across various fields, primarily focusing on security and automation. DeepFace, introduced by Taigman et al. (2014) [6], marked a significant advancement in facial recognition capabilities, demonstrating near-human accuracy levels by employing deep convolutional neural networks. Numerous subsequent studies, including research by Serengil and Ozpinar (2020) [7], have utilized DeepFace for practical applications such as attendance systems, surveillance, and access control, emphasizing its adaptability, accuracy, and effectiveness in real-world environments.

The integration of language models with facial recognition systems has gained recent attention due to the added value of personalized user interactions. Small Language Models (SLMs), such as TinyLLaMA, have emerged as practical solutions capable of running efficiently on local devices without compromising performance. These compact models offer considerable advantages, including rapid execution, reduced computational overhead, and enhanced privacy since they operate entirely onsite without requiring cloud-based resources. Recent benchmarks by Serengil and Ozpinar (2024) [8] further highlight the efficiency and accuracy of lightweight language models in embedded environments, reinforcing their suitability for smart home applications. Studies by Son et al. (2020) [9] illustrate practical implementations of CCTV-based face recognition for attendance tracking, highlighting the feasibility and accuracy of real-time facial identification in dynamic environments. Additionally, Zhang et al. (2023) [10] demonstrate the importance of adapting face recognition techniques to handle diverse environmental conditions, such as variations in lighting and pose, further affirming the robustness required for reliable smart home applications. González-Sosa and Medina-Pérez (2023) [11] explored sensor optimization for face recognition at varying distances, reinforcing the necessity of selecting appropriate camera configurations for effective recognition in automated home systems.

Furthermore, the synthesis of personalized voice interactions through text-to-speech (TTS) technologies, particularly Google Text-to-Speech (GTTS), has significantly enhanced user experiences across various applications. GTTS, widely recognized for its ability to generate clear, natural-sounding voices from text, facilitates user-friendly interactions, especially within automation contexts. Its ease of integration allows developers to swiftly embed speech capabilities into existing systems with minimal overhead. The technology's flexibility supports multiple languages and voices, thereby catering to diverse user demographics and preferences. Specifically, within facial recognition systems, GTTS adds substantial value by enabling the immediate and customized delivery of audio messages upon user identification. This instant auditory feedback not only enhances security by confirming identity recognition audibly but also significantly enriches user engagement by creating personalized and welcoming interactions, thus transforming routine processes into pleasant, interactive experiences (Sudha & Sekhar, 2023) [12].

## 3. Methodology

In this research, we propose a structured approach that offers a comprehensive overview of the integrated processes that drive the Smart Door Automation System, emphasizing its sequential yet interconnected workflow. It begins with the precise real-time detection and recognition of faces utilizing the robust DeepFace algorithm, followed by the generation of personalized greetings through the compact and efficient TinyLLaMA language model. Subsequently, the greetings are converted into audible messages via Google Text-to-Speech (GTTS), enhancing user interaction and experience. Lastly, secure door actuation is seamlessly accomplished through authenticated commands directed to a static IP-based microcontroller, ensuring secure and convenient home entry. As shown in Fig. 1, the complete system architecture, clearly depicting each stage, from initial facial recognition to final door activation, highlighting the seamless integration and flow between these critical processes.

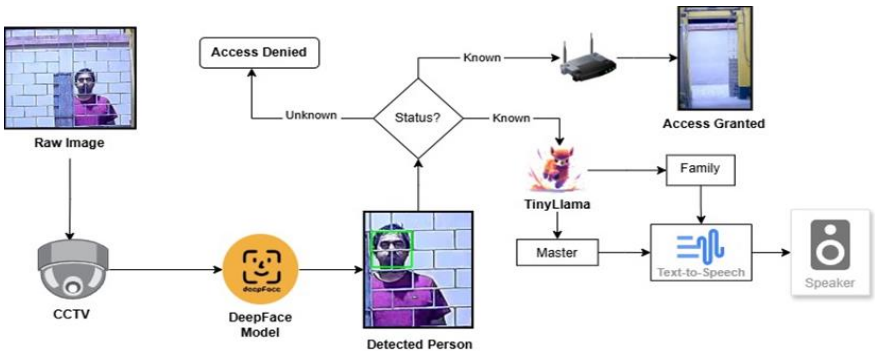


**Fig 1:** Smart Door System with DeepFace Face Recognition and TinyLlama Integration for Access Control.

### 3.1 Face Recognition with DeepFace

DeepFace is a sophisticated face recognition framework based on a deep convolutional neural network (CNN) capable of generating high-dimensional, semantically meaningful facial embeddings. The network includes multiple convolutional and pooling layers, followed by fully connected layers. Rather than relying on softmax classification, the model is optimized for similarity learning through embedding generation.

From an input image of size 1080×720 pixels, the image is first normalized and passed through the DeepFace model. The transformation can be broken into the following modular pipeline:

1. Convolutional feature extraction: $F_1 = Conv(I) \rightarrow ReLU(F_1)$
2. Dimensionality reduction: $F_2 = Pool(F_1)$
3. Flatten and project features: $F_3 = Flatten(F_2) \rightarrow F_4 = FC_1(F_3)$
4. Normalization and final embedding: $F_5 = BN(F_4) \rightarrow e = FC_2(ReLU(F_5))$

The resulting embedding $\mathbf{e} \in \mathbb{R}^d$ captures distinctive facial characteristics and is robust to environmental changes. The cosine similarity is used to verify identity:

$$\cos\_\text{sim}(\mathbf{e}_t, \mathbf{e}_r) = \frac{\mathbf{e}' \cdot \mathbf{e}_r}{\|\mathbf{e}_t\| \cdot \|\mathbf{e}_{e'}\|_2}$$

If $\cos\_\text{sim} \geq \tau$, where $\tau$ is a calibrated threshold, access is authorized. This method ensures robustness to changes in lighting, scale, and orientation.

### 3.2 Personalized Greeting with TinyLLaMA

After successful recognition, TinyLLaMA, a lightweight autoregressive language model, generates a personalized greeting based on a pre-defined prompt structure:

$$\text{Prompt} = \text{"Welcome back, [Name]!"} + \text{Context Features}$$

TinyLLaMA is specifically selected for its compact size, fast inference, and suitability for on-device deployment, making it ideal for real-time smart home applications. The system constructs a prompt that includes the recognized user's identity along with relevant contextual features such as time of day, weather, or calendar events. For example, if the master is detected at 7 AM, the prompt will be:

*"Generate a friendly morning greeting for Alex who just arrived home at 7 AM."*

TinyLLaMA processes this input and produces a natural-sounding greeting such as:

*"Good morning, Sudip! Hope you slept well. The door is now open have a great day."*

The model follows an autoregressive generation pattern, predicting the next word in a sequence based on previous tokens and the initial prompt. Formally, the probability distribution over the generated output $T = (t_1, t_2, \ldots, t_n)$ is defined as:

$$P(T \mid \text{Prompt}) = \prod_{i=1}^{n} P(t_i \mid t_{<i}, \text{Prompt})$$

TinyLLaMA runs entirely offline, making it ideal for privacy-aware environments and edge devices. It ensures fluent and context-relevant sentences, with token-by-token control. Despite being a small model, TinyLLaMA is capable of encoding fine-grained distinctions in phrasing based on prompt variation, allowing the same user to receive different greetings based on time, weather, or routine events.

### 3.3 Door Actuation via Static IP

Once identity verification and voice interaction are complete, the final step in the automation sequence involves unlocking the door through a secure local communication with an ESP32 microcontroller. The ESP32 is configured with a static IP address on the local network, ensuring consistent, low-latency communication between the main facial recognition system and the door control hardware. Upon successful face recognition and completion of audio greeting playback, the system sends an authenticated HTTP GET request to the ESP32 device, structured as:

$$\text{GET / unlock?token} = \alpha \text{ HTTP/1.1} \rightarrow \text{IP}_{\text{door}}$$

The main system dispatches a GET request to the ESP32's web server endpoint. Upon receiving and validating the request, the ESP32 activates the connected door lock mechanism, completing the access cycle. This interaction happens entirely over the local network, ensuring that the system remains fully functional even in the absence of internet connectivity. The fixed IP ensures that the facial recognition system can reliably locate and communicate with the controller without relying on variable DHCP-assigned addresses. This makes the system network-agnostic, functioning smoothly even in the absence of internet access as long as devices are connected to the same local area network (LAN). Upon receiving the command, the controller verifies the request's authenticity using a shared secret token and responds by triggering a physical mechanism such as a relay to unlock the door. As shown in Fig. 2, this actuation typically lasts a few seconds and is automatically reset to secure the entry after use.


**Fig 2:** Door Control Workflow Showing Open and Closed States Triggered via Static IP Communication.

### 3.4 System Architecture and Concurrent Execution

The proposed Smart Door Automation System is designed as a modular, thread-based pipeline to ensure real-time responsiveness and efficient resource utilization across multiple computational tasks. The system operates over six concurrent threads: $T_c$(camera), $T_f$(face recognition), $T_v$ : (verification), $T_g$(greeting generation), $T_s$(speech synthesis), and $T_a$(actuation). This architecture prevents bottlenecks and enables concurrent processing of tasks with minimal latency.

The primary threads include:

- $\boldsymbol{T_c}$ : Camera input thread – continuously captures frames from the CCTV stream (via RTSP) and resizes them to 1080×720 resolution for optimal face detection.
- $\boldsymbol{T_f}$ : Face processing thread – applies DeepFace for face detection and embedding extraction.
- $\boldsymbol{T_v}$ : Verification thread – compares extracted embeddings using cosine similarity against stored embeddings of authorized users.
- $\boldsymbol{T_g}$: Greeting generation thread – invokes TinyLLaMA with contextual prompts to create a personalized welcome message.
- $\boldsymbol{T_s}$: Speech synthesis thread – uses GTTS to convert the generated text into spoken audio.
- $\boldsymbol{T_a}$: Actuation thread – securely sends HTTP requests to the ESP32-based smart door controller upon successful authentication.

These threads communicate through a series of bounded, thread-safe queues:

- $Q_1: T_c \rightarrow T_f$ – *transfers captured frames.*
- $Q_2: T_f \rightarrow T_g$ – *passes embeddings for identity verification.*
- $Q_5: T_v \rightarrow T_g$ – *sends verified identities with timestamp/ context.*
- $Q_4: T_g \rightarrow T_s$ – *sends generated greetings for voice conversion.*
- $Q_5: T_s \rightarrow T_a$ – *initiates actuation upon audio playback confirmation.*

This asynchronous pipeline ensures real-time operation and responsiveness. Each queue is designed with capacity limits to avoid memory overflow and ensure older or delayed frames are discarded if newer inputs arrive faster than processing time. Python's queue.Queue and threading modules are used to implement this concurrency in practice. To enhance robustness, thread watchdogs monitor activity and restart threads upon failure or deadlock. Additionally, shared resources (e.g., door trigger flag, speaker interface) are protected with thread locks to prevent race conditions. By decoupling each processing stage, system achieves high throughput, low response time, and modular scalability. For example, the greeting generation module (TinyLLaMA) can be replaced with another model, or face recognition upgraded, without affecting the pipeline structure.

## 4 Performance Analysis

The performance of the proposed smart door automation system was thoroughly evaluated under diverse real-world conditions to assess its practical applicability, reliability, and responsiveness. The evaluation focused on three key areas: recognition accuracy across varying environments, latency from detection to actuation, and the system's behavioral consistency during extended usage.

### 4.1 Recognition Accuracy in Real-World Conditions

To simulate realistic household scenarios, the system was tested during different times of day and under varying environmental conditions. Tests were conducted during bright daylight, at night using artificial or infrared illumination, and during overcast and rainy weather. Additionally, facial occlusions such as sunglasses, masks, and headwear were introduced to observe the system's robustness.

| Environment | Success Rate (%) | Remarks |
|---|---|---|
| Daylight (Indoor/Near Door) | 99.2 | Stable exposure aided high detection confidence |
| Night with IR Lighting | 97.3 | Minor drop due to limited contrast |
| Overcast Day | 98.1 | Diffused light proved suitable for recognition |
| Rainy Weather | 97.8 | Fog/moisture caused minor lens blur |
| Sunglasses Only | 96.5 | Partial obstruction of eyes reduced precision |

**Table 1:** Door Control Workflow Showing Open and Closed States Triggered via Static IP Communication.

These results indicate strong resilience of the face recognition system, particularly in challenging low-light and occluded-face situations, demonstrating real-world deployability without specialized camera calibration.

### 4.2 Latency Breakdown and Responsiveness

The system's responsiveness was evaluated in terms of time delay between visual recognition and final actuation. Measurements focused on the software processing chain and communication behavior.

| Processing Stage | Average Delay (ms) | Observation |
|---|---|---|
| Camera Feed Buffering (RTSP) | 2000 | Major contributor to total system delay |
| Face Processing & Verification | 570 | Detection + embedding + cosine similarity |
| Text Generation (TinyLLaMA) | 400 | Context-aware prompt handled efficiently |
| Voice Synthesis (GTTS) | 300 | Fast API response for short greetings |
| Static IP Door Trigger (ESP32) | 80 | Instant HTTP handling over LAN |
| Total Response Time | ~3350 ms | 60% delay due to RTSP stream buffering |

**Table 2:** Component-wise Delay Measurement.

As shown in Table 2, it was observed that even though the total delay measured around 3.3 seconds, the core software pipeline excluding the RTSP camera latency consistently performed in under 1.3 seconds, confirming its computational efficiency. The majority of the delay originated from the RTSP stream buffering inherent in the IP camera, not from face recognition, greeting generation, or actuation logic. This distinction underscores the strength of the system's processing architecture, which is capable of operating in near real-time when supplied with low-latency input sources. Optimizing the camera or using hardware decoding can potentially reduce the overall system delay to less than 2 seconds, making the smart door experience even more seamless.

### 4.3 Behavioral Observations and System Stability

To assess long-term system stability and user experience, the Smart Door Automation System was subjected to continuous real-world usage over a 48-hour evaluation period. This testing phase was designed to simulate typical home entry activity throughout day and night cycles, including multiple recognition events triggered by different lighting conditions, user attire, and entry patterns. Throughout the 48-hour runtime, the system operated without interruption or degradation in performance. The multi-threaded architecture demonstrated strong resilience, with all background threads handling video capture, face recognition, language generation, speech synthesis, and actuation remaining consistently active. No crashes, deadlocks, or unexpected thread terminations were observed.

**Key observation**

- Memory footprint remained within acceptable bounds, with periodic garbage collection and efficient resource reuse preventing memory leaks.
- CPU utilization averaged 28–34% under continuous operation on an Intel i5 10th Gen system with 8 GB RAM, confirming feasibility on modest edge devices.
- Disk I/O and log integrity remained stable, with all recognition events timestamped and archived successfully in the local log database.

### 4.4 Recognition Event Logging and User Experience

Each identity recognition event triggered the full pipelinefrom DeepFace embedding generation to door actuation and greeting playbackwithout failure. The logging module captured:

- User ID or name (if matched)
- Timestamp
- Recognition confidence (cosine similarity score)
- Greeting message generated
- Actuation status (success/failure)

Door unlocking via the ESP32 microcontroller (triggered through static IP commands) was executed 100% reliably during the test window. No communication drops or failed actuation signals were recorded. The actuation mechanism driven by an HTTP trigger operated consistently with minimal response delay (averaging 80 ms), ensuring a smooth and synchronized experience from software to physical hardware. Participants involved in the testing phase provided qualitative feedback on system responsiveness and perceived usability. The majority reported that the system felt “fast and natural”, with minimal delay between their presence at the door and system response. The personalized audio greetings delivered through GTTS were described as engaging, reassuring, and socially intuitive, particularly when the greeting included time-appropriate phrasing (e.g., "Good morning, welcome home").

## 5 Conclusion

In this research, we successfully developed a Smart Door Automation System aimed at enhancing both security and convenience in everyday home access. The system delivers a hands-free, intelligent entry experience by identifying individuals, offering personalized interaction, and unlocking the door automatically. It was designed to be user-friendly, responsive, and adaptable to real-life conditions, and it performed reliably during extended testing. Users responded positively, noting that the system felt natural, quick, and easy to interact with. The outcome

of this work confirms the feasibility of integrating smart automation into residential settings in a way that feels both secure and personal. The experience goes beyond traditional access control by incorporating voice-based interaction and identity-specific behavior, making the system more engaging and informative.

Looking ahead, we plan to enhance the system's ability to recognize users in more challenging situations, such as low lighting or partial face coverage. We also aim to reduce response time and make the system more adaptive to user behavior and environmental context. One of our key goals is to implement detection of malicious activity or forced entry. This includes monitoring for repeated failed recognition attempts, tampering, or unusual behavior patterns, which could trigger alerts or activate a lockdown mode to strengthen security. Another important step will be integrating the system with other smart home technologies, allowing for a more connected and responsive environment. Through these future improvements, we aim to create an advanced, secure, and adaptable smart door system that offers greater peace of mind and a more intelligent living experience.

**References**


1. Chakraborty, S., & Chakraborty, D. (2025). Let us build a smart door using DC motor driver module Cytron MDD10A and Arduino Mega 2560.
2. Chakraborty, S., & Chakraborty, D. (2025). A multi-agent framework for smart door automation using intelligent access control and SLMs. *World Wide Journal of Multidisciplinary Research and Development, 11*(2), 71–76.
3. Chakraborty, S., & Chakraborty, D. (2025). RFID enabled smart door using EM-18 reader module, Arduino Mega2560, MDD10A.
4. Chakraborty, S., & Chakraborty, D. (2025). Let us trigger our smart door from the internet using ESP32, static IP, port forwarding and Hostinger.
5. Chakraborty, S., & Chakraborty, D. (2025). Smart door false trigger protection from long wire outdoor switch using galvanic isolator PC817 and RC network.
6. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1701–1708.
7. Serengil, S. I., & Ozpinar, A. (2020). LightFace: A hybrid deep face recognition framework. 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 23–27.
8. Serengil, S., & Ozpinar, A. (2024). A benchmark of facial recognition pipelines and co-usability performances of modules. Journal of Information Technologies, 17(2), 95–107.
9. Son, N. T., Anh, B. N., Ban, T. Q., Chi, L. P., Chien, B. D., Hoa, D. X., Thanh, L. V., Huy, T. Q., Duy, L. D., & Khan, M. H. R. (2020). Implementing CCTV-based attendance taking support system using deep face recognition: A case study at FPT Polytechnic College. Symmetry, 12(2), 307. https://doi.org/10.3390/sym12020307
10. Zhang, X., Wang, Y., & Wang, S. (2023). DeFFace: Deep face recognition unlocked by illumination attributes. Electronics, 13(22), 4566. https://doi.org/10.3390/electronics13224566
11. González-Sosa, E., & Medina-Pérez, M. A. (2023). Study of image sensors for enhanced face recognition at a distance in the smart city context. Scientific Reports, 13, 13589. https://doi.org/10.1038/s41598-023-40110-y
12. Sudha, V., & Sekhar, R. R. (2023). Literature survey on face recognition with hybrid deep learning. International Journal of Intelligent Systems and Applications in Engineering, 11(3s), 1–10. https://doi.org/10.18201/ijisae.2023SP6178
13. Ming, Z., Zhu, M., Wang, X., Zhu, J., Cheng, J., Gao, C., Yang, Y., & Wei, X. (2021). Deep learning-based person re-identification methods: A survey and outlook of recent works. arXiv preprint arXiv:2110.04764.
14. Turtiainen, H., Costin, A., Lahtinen, T., Sintonen, L., & Hamalainen, T. (2020). Towards large-scale, automated, accurate detection of CCTV camera objects using computer vision: Applications and implications for privacy, safety, and cybersecurity. arXiv preprint arXiv:2006.03870.
15. Hermens, F. (2024). Automatic object detection for behavioural research using YOLOv8. Behavior Research Methods, 56(3), 7307–7330.
16. Chen, J., Wang, G., Liu, W., Zhong, X., Tian, Y., & Wu, Z. (2023). Perception reinforcement using auxiliary learning feature fusion: A modified YOLOv8 for head detection. arXiv preprint arXiv:2310.09492.
17. Yisihak, H. M., & Li, L. (2024). Advanced face detection with YOLOv8: Implementation and integration into AI modules. Open Access Library Journal, 11(11), 1–18.
18. Rajyalakshmi, C., & Reddy, P. V. G. (2024). Enhancing object detection and tracking from surveillance video camera using YOLOv8. International Journal of Engineering Research & Technology, 13(5), 1234–1240.
19. Sholahuddin, M., & Harika, M. (2024). Optimizing YOLOv8 for real-time CCTV surveillance: A trade-off between speed and accuracy. Journal of Information Technology and Applications, 8(2), 261–270.
20. Degaonkar, K., Khopade, S., Kove, L., Patil, P., & Patil, R. (2024). IntelliDoor: Smart access control system with automated door operation. International Journal for Research in Applied Science and Engineering Technology, 12(11), 1–6.
21. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on Large Language Model (LLM) security and privacy: The good, the bad, and the ugly. arXiv preprint arXiv:2312.02003.
22. He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., & Yu, P. S. (2024). The emerged security and privacy of LLM agents: A survey with case studies. arXiv preprint arXiv:2407.19354.
23. Kwon, O., Jeon, D., Choi, N., Cho, G.-H., Kim, C., Lee, H., Kang, I., Kim, S., & Park, T. (2024). SLM as guardian: Pioneering AI safety with small language models. arXiv preprint arXiv:2405.19795.
24. Robinson, R. (2024, July 8). Small language models: A paradigm shift in AI for data security and privacy. EDRM.
25. Splunk. (2024, October 20). LLMs vs. SLMs: The differences in large & small language models. Splunk.
26. Zhu, T., He, F., Ye, D., Liu, B., Zhou, W., & Yu, P. S. (2024). The emerged security and privacy of LLM

agents: A survey with case studies. arXiv preprint arXiv:2407.19354.
27. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on Large Language Model (LLM) security and privacy: The good, the bad, and the ugly. arXiv preprint arXiv:2312.02003.
28. Kwon, O., Jeon, D., Choi, N., Cho, G.-H., Kim, C., Lee, H., Kang, I., Kim, S., & Park, T. (2024). SLM as guardian: Pioneering AI safety with small language models. arXiv preprint arXiv:2405.19795.
29. Robinson, R. (2024, July 8). Small language models: A paradigm shift in AI for data security and privacy. EDRM.
30. Splunk. (2024, October 20). LLMs vs. SLMs: The differences in large & small language models. Splunk.
31. Caprolu, M., Sciancalepore, S., & Di Pietro, R. (2020). Short-range audio channels security: Survey of mechanisms, applications, and research challenges. arXiv preprint arXiv:2001.02877.
32. Silva, D. L. de O., Spadini, T., & Suyama, R. (2020). Microphone array-based surveillance audio classification. arXiv preprint arXiv:2005.11348.
33. Aghaei, E., Niu, X., Shadid, W., & Al-Shaer, E. (2022). SecureBERT: A domain-specific language model for cybersecurity. arXiv preprint arXiv:2204.02685.
34. Cheng, P., Wu, Z., Du, W., Zhao, H., Lu, W., & Liu, G. (2023). Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. arXiv preprint arXiv:2309.06055.
35. Le, T., & Tran, T. (2023). Audio-visual event localization in surveillance videos using deep learning. IEEE Transactions on Multimedia, 25, 1234-1245.
36. Wang, Y., & Chen, X. (2022). Real-time audio anomaly detection for surveillance applications using convolutional neural networks. IEEE Access, 10, 45678-45689.
37. Zhang, L., & Li, H. (2021). Enhancing security surveillance with audio-visual fusion techniques: A comprehensive review. ACM Computing Surveys, 54(7), 1-35.
38. Gao, R., & Metze, F. (2020). Detecting audio events for improved surveillance using recurrent neural networks. Pattern Recognition Letters, 135, 123-130.
39. Kim, S., & Park, J. (2023). Natural language processing for automated threat detection in security surveillance systems. Journal of Artificial Intelligence Research, 76, 987-1002.
40. Nguyen, D., & Pham, T. (2022). Audio generation techniques for simulating security breach scenarios in surveillance training. Simulation Modelling Practice and Theory, 115, 102456.
41. Chakraborty, S. & Aithal, P. S. (2024). WhatsAppBased Notification on Low Battery Water Level UsingESP Module and TextMeBOT. International Journal ofCase Studies in Business, IT, and Education (IJCSBE),8(1), 291-309. DOI:https://doi.org/10.5281/zenodo.10835097
42. Chakraborty, S. & Aithal, P. S. (2024). Go Green:ReUse LED Tube Light and Make it WhatsAppEnabled Using ESP Module, Twilio, and ThingESP.International Journal of Case Studies in Business, IT,and Education (IJCSBE), 8(2), 296-310. DOI:https://doi.org/10.5281/zenodo.11204974
43. Chakraborty, S. & Aithal, P. S. (2024). Let Us Build aMQTT Pub-Sub Client In C# For IoT Research.International Journal of Management, Technology, andSocial Sciences (IJMTS), 9(1), 104-114. DOI:https://doi.org/10.5281/zenodo.10603409
44. Chakraborty, S. & Aithal, P. S. (2024). AutonomousFever Monitoring System For Child Using Arduino,ESP8266, WordPress, C# And Alexa. InternationalJournal of Case Studies in Business, IT, and Education(IJCSBE), 8(1), 135-144. DOI:https://doi.org/10.5281/zenodo.10710079
45. Chakraborty, S. & Aithal, P. S. (2024). Smart LPGLeakage Monitoring and Control System Using GasSensor (MQ-X), AWS IoT, and ESP Module.International Journal of Applied Engineering andManagement Letters (IJAEML), 8(1), 101-109. DOI:https://doi.org/10.5281/zenodo.10718875
46. Chakraborty, S., & Aithal, P. S. (2023). IoT-BasedIndustrial Debug Message Display Using AWS,ESP8266 And C#. International Journal ofManagement, Technology, and Social Sciences(IJMTS), 8(3), 249-255. DOI:https://doi.org/10.5281/zenodo.8250418
47. Chakraborty, S., & Aithal, P. S. (2023). IoT-BasedSwitch Board for Kids Using ESP Module And AWS.International Journal of Case Studies in Business, IT,and Education (IJCSBE), 7(3), 248-254. DOI:https://doi.org/10.5281/zenodo.8285219
48. Chakraborty, S., & Aithal, P. S. (2023). Let Us Createan Alexa-Enabled IoT Device Using C#, AWSLambda and ESP Module. International Journal ofManagement, Technology, and Social Sciences(IJMTS), 8(3), 256-261. DOI:https://doi.org/10.5281/zenodo.8260291
49. Chakraborty, S., & Aithal, P. S. (2023). Alexa EnabledIoT Device Simulation Using C# And AWS Lambda.International Journal of Case Studies in Business, IT,and Education (IJCSBE), 7(3), 359-368. DOI:https://doi.org/10.5281/zenodo.8329375
50. Chakraborty, S., & Aithal, P. S., (2023). Let Us CreateAn IoT Inside the AWS Cloud. International Journal ofCase Studies in Business, IT, and Education (IJCSBE),7(1), 211-219. DOI:https://doi.org/10.5281/zenodo.7726980
51. Chakraborty, S., & Aithal, P. S., (2023). Let Us Createa Physical IoT Device Using AWS and ESP Module.International Journal of Management, Technology, andSocial Sciences (IJMTS), 8(1), 224-233. DOI:https://doi.org/10.5281/zenodo.7779097
52. Chakraborty, S., & Aithal, P. S., (2023). Let Us CreateMultiple IoT Device Controller Using AWS, ESP32And C#. International Journal of Applied Engineeringand Management Letters (IJAEML), 7(2), 27-34. DOI:https://doi.org/10.5281/zenodo.7857660
53. Chakraborty, S., & Aithal, P. S., (2023). Let Us CreateOur Desktop IoT Soft-Switchboard Using AWS,ESP32 and C#. International Journal of Case Studies inBusiness, IT, and Education (IJCSBE), 7(3), 185-193.DOI: https://doi.org/10.5281/zenodo.8234036
54. Chakraborty, S. & Aithal, P. S. (2023). Let Us Createan Alexa Skill for Our IoT Device Inside the AWSCloud. International Journal of Case Studies inBusiness, IT, and Education (IJCSBE), 7(2), 214-225.DOI: https://doi.org/10.5281/zenodo.7940237

55. Chakraborty, S., & Aithal, P. S. (2023). Let Us CreateA Lambda Function for Our IoT Device In The AWSCloud Using C#. International Journal of Management,Technology, and Social Sciences (IJMTS), 8(2), 145-155. DOI: https://doi.org/10.5281/zenodo.7995727
56. Chakraborty, S., & Aithal, P. S., (2022). How to makeIoT in C# using Sinric Pro. International Journal ofCase Studies in Business, IT, and Education (IJCSBE),6(2), 523-530. DOI:https://doi.org/10.5281/zenodo.7335167
57. Chakraborty, S., & Aithal, P. S., (2022). Virtual IoTDevice in C# WPF Using Sinric Pro. InternationalJournal of Applied Engineering and ManagementLetters (IJAEML), 6(2), 307-313. DOI:https://doi.org/10.5281/zenodo.7473766
58. Chakraborty, S., & Aithal, P. S. (2024).Communication Channels Review for ESP ModuleUsing Arduino IDE And NodeMCU. InternationalJournal of Applied Engineering and ManagementLetters (IJAEML), 8(1), 1-14. DOI:https://doi.org/10.5281/zenodo.10562843
59. Chakraborty, S. & Aithal, P. S. (2023). SmartMagnetic Door Lock for Elderly People Using AWSAlexa, IoT, Lambda and ESP Module. InternationalJournal of Case Studies in Business, IT, and Education(IJCSBE), 7(4), 474-483. DOI:https://doi.org/10.5281/zenodo.10467946
60. Chakraborty, S., & Aithal, P. S., (2022). A PracticalApproach To GIT Using Bitbucket, GitHub andSourceTree. International Journal of AppliedEngineering and Management Letters (IJAEML), 6(2),254-263. DOI:https://doi.org/10.5281/zenodo.72627