Grenze International Journal of Engineering and Technology, June Issue

AI-based Intrusion Detection System for IoT-Enabled Smart Homes

Jayashree[1], Sony Prasad Rao Teetla[2], Vikas Gowda S[3], Samiran Sil[4], Sijal Satyamev[5] and Shivaprakash T[6]
[1-5]Nitte (Deemed to be university), Nitte Meenakshi Institute of Technology, Bangalore, India
Email: jayashree.cse@nmit.ac.in, 1nt23cs413.sony@nmit.ac.in, 1nt23cs417.vikas@nmit.ac.in, 1nt22cs178.samiran@nmit.ac.in, 1nt22cs192.sijal@nmit.ac.in
[6]Vijaya Vittala Institute of Technology, Bangalore, India
Email: spt@ieee.org

***Abstract*— Over the past few years, there has been a surge in demand for intelligent home and office security systems based on rising security needs. This project introduces an AI-powered Smart Intrusion Detection System that combines motion sensing, object detection, and face recognition for real-time monitoring and threat detection. The system employs a PIR motion sensor to monitor activity in the environment and a camera module to record frames on detection. Object detection is carried out through a TensorFlow Lite (TFLite) model — SSD MobileNet V2, that has been trained on the COCO dataset, detecting the presence of humans or other objects in motion with high effectiveness on low-resource devices such as the Raspberry Pi 4. After detection of a person, the DeepFace framework utilizing the VGG-Face model cross-checks the identity of the person by matching their face with pre-stored images of authenticated users. The suggested system operates without an internet connection, can analyze data in real time, and can be connected to MQTT and Telegram alerts for remote alerting. The study demonstrates the creation of intelligent surveillance systems that can autonomously distinguish between authorized users and potential intruders by integrating deep learning algorithms with cost-effective embedded hardware.**

***Index Terms*— AI, DL, SSD, MQTT, TFLite, VGG, COCO, PIR, GPIO.**

I. INTRODUCTION

Smart homes that employ Internet of Things (IoT) devices make it easier to keep an eye on and control home systems from a distance. But being linked like this is quite dangerous for security, and they are likely to be assaulted online and in person. An Intrusion Detection System (IDS) that employs AI, machine learning (ML), and IoT sensors to find and block attacks straight immediately. This project proposes a multi-tiered intrusion detection system employing motion sensors, cameras, and machine learning algorithms to identify anomalous behaviour and promptly inform homeowners.

The Internet of Things (IoT) has grown so quickly that it has turned houses into smart places where you can operate and keep a watch on appliances, sensors, and security systems from anywhere. This new technology makes things easier, more automated, and better for the environment, but it also makes them less safe. Because the smart home has so many linked devices, it might be a target for both cyberattacks such unauthorized access, spoofing, and data manipulation, as well as real-life break-ins that employ weak or unmonitored entry points.

Grenze ID: 01.GIJET.12.2.56_23

Most modern home security systems, like CCTV or simple alarm sensors, are made to react to situations instead of doing things on their own. They also aren't very smart and can't distinguish the difference between regular and abnormal behavior. This means we need a security system that is stronger, smarter, and works on its own. AI-powered IDS is a very appealing option, especially when you think about the problems stated above. AI-based IDS uses machine learning and IoT sensors to find patterns in sensor data, spot abnormal behavior, and block attacks as they happen. With motion sensors, cameras, and processing units that are connected to the cloud, it can always watch your surroundings and spot strange behavior. These kinds of technology not only help you discover items faster and simpler, but they also cut down on false alarms generated by pets, changes in light, or noise from outside.

The major thing the project achieved was design and create a smart home intrusion detection system that uses AI and integrates PIR motion sensors, the Raspberry Pi camera module, and a facial recognition/motion capture model that learns from its mistakes. Our method assures the following, which is not the case with other systems:

- Real-time detection of motion and physical intrusions.
- Automatic photo capture and storage whenever suspicious activity is occurring.
- AI-powered recognition that distinguishes authorized from unauthorized people.
- Modular and scalable IoT architecture: They allow the system to be integrated into any setup of a smart home.
- A low-cost, easily deployable prototype based on Raspberry Pi makes the solution affordable for small households.

The motivation lies in the ever-increasing demand for affordable, intelligent home security solutions that do not always rely on commercial products with heavy price tags or continuous human monitoring. More and more people are getting smart homes, but security issues are also on the rise. An AI-enhanced IDS can make homes that use the Internet of Things significantly safer and more dependable in general.The rest of the paper is set up like this:

- Section II talks about the basic ideas behind IoT security, intrusion detection, and other technologies that are relevant.
- Section III speaks about the proposed system's hardware pieces, architecture, and data flow.
- Section IV speaks about the method, which includes logic for finding motion, a pipeline for taking pictures, and putting the AI model together.
- Section V speaks about how to carry out the strategy and what the testing showed. In Section VI, the job is done and the next stages for making the system smarter and more robust are explained.

II. RELATED WORK

A. Background

TABLE I: TABLE OF LITERATURE SURVEY

SI.no	Author / Year	Method / Model Used	Key Findings
01	Joseph Redmon et al., 2016 [1]	YOLO Object Detection	Real-time object detection with high accuracy.
02	Y. Taigman et al., 2014 [2]	DeepFace (Facebook AI)	Reached human-level performance in face verification with deep neural networks.
03	S. I. Serengil & A. Ozpinar, 2023 [3]	DeepFace Framework (Python-based).	Provided a lightweight, easy-to-use face recognition library for Python supporting multiple models.
04	TensorFlow Developers, 2024 [4]	TensorFlow Lite (TFLite)	Enabled deployment of optimized ML models on mobile and embedded devices with low latency.
05	Eclipse Foundation, 2023 [5]	MQTT IoT Messaging Protocol	Offered lightweight publish–subscribe communication ideal for IoT devices with low power and bandwidth.
06	A. Howard et al., 2017 [6]	MobileNet CNN Architecture	Introduced efficient CNNs for mobile/edge devices, balancing accuracy and computational cost.
07	A. Boukerche & R. W. L. Coutinho, 2018 [7]	Smart Home AAL Framework	Discussed challenges and future directions in smart homes designed for ambient assisted living.
08	H. Abie & I. Balasingham, 2019 [8]	Risk-based Adaptive IoT Security	Proposed adaptive security mechanisms for IoT, improving reliability in medical/eHealth environments.
09	G. Al-Rawi et al., 2020 [9]	Smart Home Cybersecurity Survey	Identified major threats and solutions for securing smart home IoT networks.
10	S. H. Islam et al., 2019 [10]	Raspberry Pi–based IoT Smart Home Security System	Demonstrated a working smart home security system integrating sensors, notifications, and automation.

Smart surveillance systems are taking the role of old-fashioned motion sensors in homes and businesses. Normal CCTV systems only record things; they don't see them and react to them in real time. Computer vision and AI

can now automatically discover persons and recognize faces that have been given permission. You can send messages and keep an eye on things from afar using IoT integration (such Raspberry Pi and MQTT).
Table 1 shows the research papers we used for our study and the conclusions we received from them. Worked further on the project to fill in the holes in the research.

III. System Design

The circuit links a USB camera and a PIR motion sensor to the Raspberry Pi 4. The Pi's 5V and Ground pins are connected to the PIR sensor's VCC and GND pins. The PIR sensor's output pin is connected to GPIO17, which lets it detect motion. The USB-connected camera takes a picture for analysis when it detects movement. The whole system relies on a 5V power source, which makes it easy to connect sensors and AI processing, making it a good real-time intrusion detection system. The Figure 1 shows is the visualization of the Circuit breed board connection. It is a pictorial representation which we found out researching on the hardware connection.

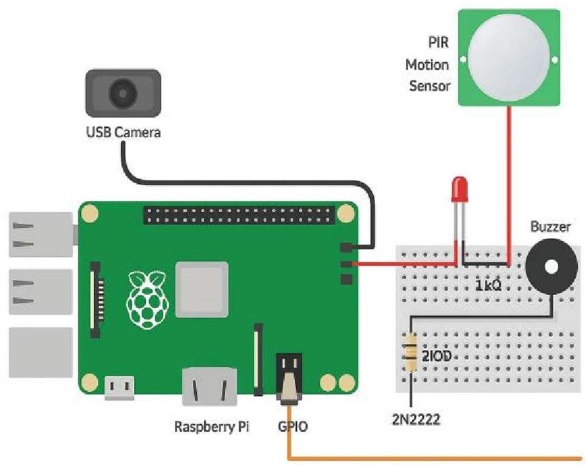


Figure 1: Circuit Breed board Connection

A. Abbreviations and Acronyms AI – Artificial Intelligence

DL – Deep Learning
SSD – Single shot detection
MQTT – Message Queuing Telemetry Transport
TFLite – Tensor Flow Lite
VGG – Visual Geometry Group
COCO – Common Objects in Context Dataset
PIR – Passive Infrared
GPIO – General Purpose Input/Output

B. Units

- Volt (V) - Electrical potential (e.g., PIR sensor operates at 5V)
- Frames Per Second (FPS) - Measures the speed of real-time video processing.
- Watt (W) - Power consumption of Raspberry Pi and sensors.

C. Workflow

- PIR triggers → GPIO event → start camera capture (short clip or N frames).
- Run TFLite detection (person detection).
- If person found → crop region → run face recognition (DeepFace or tflite face embedder).
- If unknown → save snapshot and send alert (MQTT publish / Telegram).
- Optionally trigger local alarm (buzzer/LED).

The Figure. 2 shows is the visualization of the work flow of this complete project. It shows walk through the project how hardware and software are compatible, it gives clear visualization to the non-IT person’s on first glance.

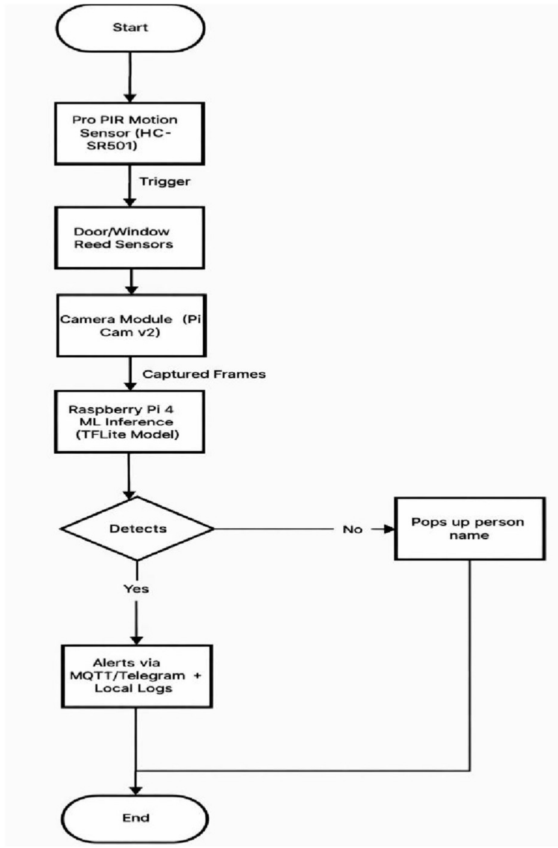


Figure 2: Work Flow

IV. Results and Evaluation

The below are the results that are been executed and this project is focusing on availability to users.

- Successfully integrated TFLite model (ssd_mobilenet_v2_coco.tflite).
- Real-time webcam detection and classification achieved at ~15–20 FPS.
- DeepFace successfully verifies authorized users vs unknown faces.
- Intruder snapshots are stored automatically in captures
- Motion simulation (PIR) triggers camera capture correctly.
- System runs fully offline on laptop; ready for Raspberry Pi deployment.

A. Hardware Part (Circuit Connection)

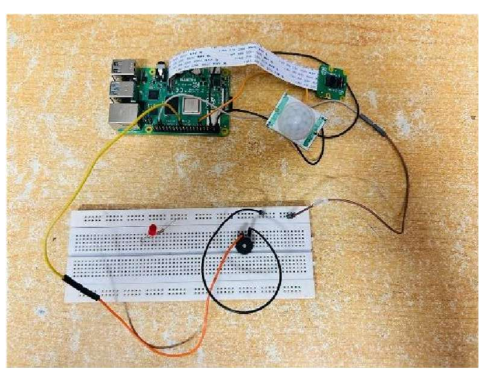
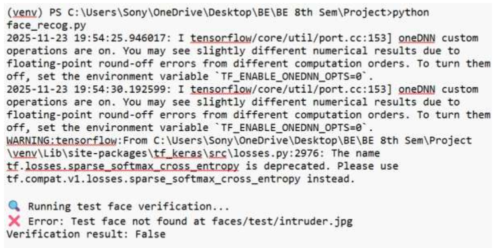


Figure 3: Loading Raspberry OS

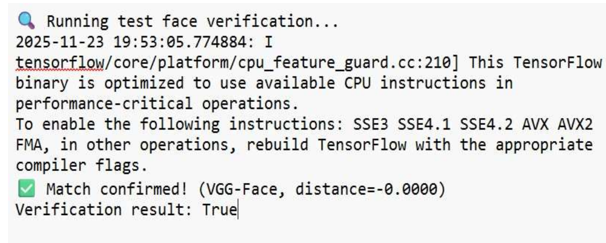
The Figure. 3 shows successfully Raspberry launching is done and it is ready to work. This is the Model where hardware connections are visualized (hardware components included in this are Raspberry Pi 4, Breed Board, LED, Resistor, Buzzer, Raspberry Camera Module 2, PIR sensor and Jumpers to Connect). B. Software Part :



```
(venv) PS C:\Users\Sony\OneDrive\Desktop\BE\BE 8th Sem\Project>python
face_recog.py
2025-11-23 19:54:25.946017: I tensorflow/core/util/port.cc:153] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-11-23 19:54:30.192599: I tensorflow/core/util/port.cc:153] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\Sony\OneDrive\Desktop\BE\BE 8th Sem\Project
\venv\Lib\site-packages\tf_keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

🔍 Running test face verification...
❌ Error: Test face not found at faces/test/intruder.jpg
Verification result: False
```

Figure 4. Mismatch of Manual Comparison of owner and intruder images

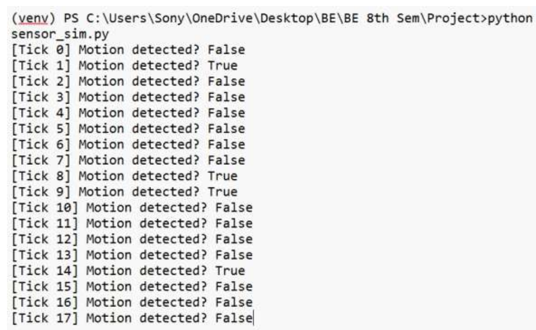


```
🔍 Running test face verification...
2025-11-23 19:53:05.774884: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow
binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2
FMA, in other operations, rebuild TensorFlow with the appropriate
compiler flags.
✅ Match confirmed! (VGG-Face, distance=-0.0000)
Verification result: True
```

Figure 5: Match of Manual Comparison of owner and intruder images

```
System armed. Monitoring...
[20:09:27] ALERT: Motion - PIR triggered!
2025-11-23 20:09:33.539499: I
tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is
optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other
operations, rebuild TensorFlow with the appropriate compiler flags.
✅ Match confirmed! (VGG-Face, distance=-0.0000)
[20:09:44] ALERT: Authorized - Known face detected
[20:09:58] ALERT: Motion - PIR triggered!
✅ Match confirmed! (VGG-Face, distance=-0.0000)
[20:10:05] ALERT: Authorized - Known face detected
[20:10:17] ALERT: Motion - PIR triggered!
[20:10:22] ALERT: Noise - No person detected
[20:10:24] ALERT: Motion - PIR triggered!
[20:10:30] ALERT: Noise - No person detected
[20:10:36] ALERT: Motion - PIR triggered!
✅ Match confirmed! (VGG-Face, distance=-0.0000)
[20:10:43] ALERT: Authorized - Known face detected
[20:10:57] ALERT: Motion - PIR triggered!
✅ Match confirmed! (VGG-Face, distance=-0.0000)
[20:11:03] ALERT: Authorized - Known face detected
```

Figure 6: Real-Time, Automatic Testing via WebCam



```
(venv) PS C:\Users\Sony\OneDrive\Desktop\BE\BE 8th Sem\Project>python
sensor_sim.py
[Tick 0] Motion detected? False
[Tick 1] Motion detected? True
[Tick 2] Motion detected? False
[Tick 3] Motion detected? False
[Tick 4] Motion detected? False
[Tick 5] Motion detected? False
[Tick 6] Motion detected? False
[Tick 7] Motion detected? False
[Tick 8] Motion detected? True
[Tick 9] Motion detected? True
[Tick 10] Motion detected? False
[Tick 11] Motion detected? False
[Tick 12] Motion detected? False
[Tick 13] Motion detected? False
[Tick 14] Motion detected? True
[Tick 15] Motion detected? False
[Tick 16] Motion detected? False
[Tick 17] Motion detected? False
```

Figure 7: Real-Time, Automatic Testing via PIR sensor

The Figure. 4 shows validation between two images when manually given whitelist, that is the list of authorized owners (users) and the images given to test. And returns not found when unknown person enters. The Figure. 5 shows validation between two images when manually given whitelist, that is the list of authorized owners (users) and the images given to test. And returns found when known person enters. The Figure. 6 shows that PIR motion triggers and Automatically it turns on camera and starts to take real time images using WebCam. It triggers and detects continuously until an developer/users Off's it manually. The Figure. 7 shows that PIR motion triggers and Automatically it detects if Human is Found else triggers no Human found. It triggers and detects continuously until an developer/users Off's it manually.

V. CONCLUSION

The planned solution uses deep learning techniques to fully automate intrusion detection from start to finish. The system can recognize authorized persons and detect intruders in real time by combining motion sensing, object identification, and facial recognition in a smart way.

Its lightweight form makes it likely to work well on cheap edge devices, making it a good choice for smart home and business security systems. Next, the system will be connected to a Raspberry Pi platform and use MQTT-based communication and Telegram notifications to let people know about things happening in real time and let them watch from afar [11-13].

FUTURE WORK

In the following step, this project will be improved by connecting the whole system to a Raspberry Pi 4 for real-time use. Using the MQTT protocol for IoT-based communication and Telegram bot notifications for quick user warnings, the intrusion alarms will be automated. More changes include adding more data to make face recognition more accurate, making the model work better on low-power devices, and enabling cloud storage for photographs and event logs that have been acquired. The final goal is to develop a fully autonomous, intelligent surveillance system suitable for smart home and industrial security applications [14-15].

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[2] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708, 2014.

[3] S. I. Serengil and A. Ozpinar, “DeepFace: A lightweight face recognition framework for Python,” GitHub Repository, 2023. [Online].

[4] TensorFlow Lite Documentation, “TensorFlow Lite models and deployment guide,” Google Developers, 2024. [Online].

[5] MQTT Protocol Documentation, “MQTT: Lightweight messaging protocol for IoT,” Eclipse Foundation, 2023. [Online].

[6] A. Howard, M. Sandler, B. Chen, and M. Zhu, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

[7] A. Boukerche and R. W. L. Coutinho, “Smart homes for ambient assisted living: Challenges and future directions,” IEEE Communications Magazine, vol. 56, no. 6, pp. 24–30, 2018.

[8] H. Abie and I. Balasingham, “Risk-based adaptive security for smart IoT in eHealth,” Proceedings of the IEEE, vol. 107, no. 1, pp. 75–90, 2019.

[9] G. Al-Rawi, A. Dehghantanha, K.-K. R. Choo, and S. M. Furnell, “A survey of smart home security challenges and solutions,” IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2310–2336, 2020.

[10] S. H. Islam, M. A. Razzaque, M. M. Hassan, and I. R. Sarwar, “Design and implementation of an IoT-based smart home security system using Raspberry Pi,” IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–6, 2019.

[11] Jayashree, Shiva Prakash T., Venugopal K.R., MUNPE:Multi-view uncorrelated neighborhood preserving embedding for unsupervised feature extraction, Knowledge-Based Systems, Volume 287, 2024, 111421, ISSN 0950-7051,

[12] Karthik, K. Preethi Salian, R. H D, S. Salian, Jayashree and K. Annapoorneshwari Shetty, ”Wearable Sensor-Based Gait Analysis: Advanced Feature Extraction Techniques,” 2024 Second International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, Karnataka, India, 2024, pp. 1-6, doi: 10.1109/ICAIT61638.2024.10690387.

[13] Jayashree, T. Shiva Prakash, K.R. Venugopal, Unsupervised feature extraction based on uncorrelated approach, Information Sciences, Volume 666, 2024, 120447, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2024.120447.

[14] Jayashree, Shivaprakash, T.”Optimal Value for Number of Clusters in a Dataset for Clustering Algorithm, Artificial Intelligence and Sustainable Computing. Algorithms for Intelligent Systems”. Springer, Singapore.

[15] Savio, A. Sanchith, et al. ”Stock Market Analysis using Supervised using Machine Learning Technique”. Grenze International Journal of Engineering and Technology (GIJET) 9.2 2023.