iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78508

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Temple and Pilgrimage Crowd Management

Dr. T. P. Anadamuthu[1], Shobana S[2], Hemalatha V[3], Harini V[5], Yasmin Banu S[6]
[1]Corresponding Author, M.P. Nachimuthu M. Jaganathan Engineering College
[2, 3, 4, 5]Department of Information Technology M.P. Nachimuthu M. Jaganathan Engineering College

Abstract: Temples and religious places frequently experience heavy crowd congestion, particularly during festivals, special events, and peak visiting hours. Managing such large gatherings manually is often difficult and inefficient, leading to long waiting times, overcrowding, and potential safety risks for devotees. Traditional queue management methods cannot provide real time monitoring or effective crowd control. Therefore, there is a growing need for an intelligent system that can monitor, analyse, and regulate crowd movement within temple premises. This research proposes a Smart Temple Crowd and Queue Management System using Artificial Intelligence (AI) and Internet of Things (IoT) technologies to improve crowd monitoring and ensure better management of temple visitors. The proposed system integrates computer vision techniques and sensor-based monitoring to accurately estimate the number of people inside the temple area. A camera placed within the temple environment captures live video feeds, which are processed using a YOLOv8-based deep learning model for real-time human detection and counting. The model identifies individuals in each video frame and calculates the number of people currently present in the monitored area. This allows the system to determine crowd density dynamically and update the status continuously. In addition to the camera-based monitoring system, IR sensors connected to an ESP32 microcontroller are installed at the temple entrance and exit points. These sensors detect the movement of devotees entering and leaving the temple. The sensor data helps maintain an accurate record of the number of people entering and exiting the premises. By combining the information from both the AI-based vision system and the sensor-based entry– exit monitoring system, the system can estimate the real-time crowd population more reliably. All collected data is transmitted to a Flask-based backend server, where it is processed and stored in a database for further analysis. The backend system categorizes the crowd density into different levels such as Low, Medium, and High based on predefined threshold values. This classification helps temple administrators understand the current crowd situation and make informed decisions regarding visitor management. The system also provides a web-based administrative dashboard where live crowd statistics, alerts, and camera feeds can be monitored in real time.

To further improve crowd control, the system introduces a smart token generation mechanism for devotees visiting the temple. Visitors can obtain tokens that include a specific time slot for temple entry, helping distribute visitor flow evenly throughout the day. When the crowd density reaches a high level, the system automatically restricts immediate token generation and instead allows booking for future time slots. This approach prevents overcrowding and helps maintain a safer and more organized queue management system. Overall, the proposed Smart Temple Crowd and Queue Management System aim to provide an efficient, automated, and intelligent solution for temple crowd management. By combining Artificial Intelligence, IoT sensors, real-time data processing, and a smart token system, the system improves visitor experience, enhances safety, and supports temple authorities in managing large gatherings more effectively.

Keywords: Artificial Intelligence, Crowd Management, YOLOv8, Internet of Things (IoT), ESP32, Computer Vision, Smart Queue Management, Temple Management System .

I. INTRODUCTION

Religious places such as temples attract many devotees every day, especially during festivals, special ceremonies, and weekends. In many famous temples, thousands of visitors gather within a limited period, which often leads to overcrowding and long waiting queues. Managing such large crowds efficiently is a major challenge for temple authorities. Traditional crowd management methods usually rely on manual monitoring by staff members and security personnel. However, manual systems are often inefficient in handling sudden increases in crowd size and may fail to provide accurate real-time information about the number of people present inside the temple premises. As a result, devotees may experience long waiting times, discomfort, and even potential safety risks due to overcrowding. Crowd congestion in temples not only affects the experience of visitors but also creates operational challenges for temple management. During peak hours or festival days, the number of devotees entering the temple can increase rapidly, making it difficult to regulate the flow of people. Without proper monitoring systems, authorities may not be able to identify overcrowded areas or manage entry and exit effectively.

In extreme cases, overcrowding may lead to stampede-like situations, which pose serious safety concerns. Therefore, there is a growing need for an intelligent system that can monitor crowd density in real time and assist authorities in controlling the flow of devotees efficiently. Recent advancements in Artificial Intelligence (AI), Computer Vision, and Internet of Things (IoT) technologies have opened new possibilities for developing smart crowd management solutions. AI-based computer vision systems can analyse live video streams and detecting objects such as humans with high accuracy. Deep learning models such as YOLO (You Only Look Once) have become widely used for real-time object detection tasks due to their speed and reliability. By applying such models to surveillance cameras installed in temple environments, it becomes possible to automatically detect and count the number of people present in a specific area. This information can be used to estimate crowd density and monitor congestion levels continuously. In addition to camera-based monitoring systems, IoT technologies provide an effective way to track physical movement through sensors and connected devices. Sensors such as Infrared (IR) sensors can detect the presence or movement of individuals passing through entry and exit points. When integrated with microcontrollers such as ESP32, these sensors can transmit real time data to a centralized system through wireless communication. Combining sensor-based monitoring with AI powered computer vision improves the overall accuracy of crowd estimation and provides a more reliable solution for managing large gatherings. Another major challenge faced by temples is the management of queues for darshan (temple visit). Devotees often must stand in long lines for several hours, especially during peak visiting periods. Manual token systems or physical queue arrangements are often inefficient and may cause confusion among visitors. A digital queue management system can help distribute visitors across different time slots, reducing congestion and improving the overall experience for devotees. By integrating a smart token generation system with crowd monitoring technologies, temple authorities can regulate the entry of visitors based on real-time crowd conditions. The proposed **Smart Temple** Crowd and Queue Management System aim to address these challenges by integrating AI-based crowd detection, IoT-based entry and exit monitoring, and a digital token management system. In this system, cameras installed within the temple premises capture live video feeds that are processed using a YOLOv8 deep learning model to detect and count people in real time. At the same time, IR sensors connected to an ESP32 microcontroller monitor the number of people entering and leaving the temple. The combined data from these sources is sent to a Flask-based backend server, where the crowd density is analysed and categorized into different levels such as low, medium, and high. The system also includes an administrative web dashboard that allows temple authorities to monitor real-time crowd statistics, live camera feeds, and historical data. This dashboard helps administrators understand the current crowd situation and take appropriate actions when the crowd level becomes high. Additionally, a smart token booking system is integrated into the system, allowing devotees to reserve a time slot for temple entry. When the crowd density exceeds a predefined threshold, the system temporarily stops issuing immediate tokens and instead allows booking for future time slots. This mechanism helps prevent overcrowding and ensures smoother crowd flow throughout the day.

Overall, the integration of Artificial Intelligence, IoT sensors, and web-based monitoring platforms offers a modern and efficient solution for temple crowd management. The proposed system not only improves safety and operational efficiency but also enhances the experience of devotees by reducing waiting times and ensuring a more organized queue system. By adopting such intelligent technologies, temples can move towards smarter infrastructure that supports better management of large gatherings and provides a safer environment for visitors.

II. IMPLEMENTATION

The implementation of the proposed Smart Temple Crowd and Queue Management System focuses on integrating Artificial Intelligence, IoT hardware components, and web-based technologies to develop a functional system capable of monitoring and managing temple crowds in real time. The system is implemented using a combination of hardware devices, software modules, and **a** backend server architecture. Each component works together to collect data, process crowd information, and display results through an administrative dashboard.

A. Hardware Implementation

The hardware implementation consists of IR sensors and an ESP32 microcontroller installed at the temple entrance and exit points. The IR sensors detect the movement of devotees passing through the gates. When a person crosses the entry sensor, the system records it as an entry event, and when the exit sensor is triggered, it records an exit event.

The sensors are connected to the ESP32 microcontroller, which processes the sensor signals and counts the number of people entering and leaving the temple premises. The ESP32 device is capable of wireless communication and sends the collected data to the backend server through a network connection. This hardware setup provides continuous monitoring of visitor flow and helps maintain an accurate count of people inside the temple.

B. AI-Based Crowd Detection Module

The AI module is implemented using Python, OpenCV, and the YOLOv8 deep learning model. A camera installed inside the temple premises captures live video streams, which are processed by the AI detection module. The video frames are analysed using the YOLOv8 model to identify and detect human objects.

The model draws bounding boxes around each detected person in the video frame and counts the total number of individuals present in the scene. This detection process runs continuously to estimate the crowd density in real time. The detected crowd count is then sent to the backend server for further processing.

C. Backend Server Implementation

The backend system is developed using the Flask web framework in Python. The backend server acts as the central processing unit of the entire system. It receives data from both the AI detection module and the ESP32 sensor module. The server processes the incoming information and calculates the current number of people inside the temple.

The backend also determines the crowd level category based on predefined threshold values such as low, medium, or high crowd density. The processed data is stored in a MySQL database, which maintains records of crowd statistics, entry and exit counts, and historical data for analysis.

D. Web-Based Admin Dashboard

A web-based administrative dashboard is implemented to allow temple authorities to monitor the system in real time. The dashboard is developed using HTML, CSS, and JavaScript, and it communicates with the Flask backend server through API endpoints.

The dashboard displays important information such as the current crowd count, entry and exit statistics, crowd level status, and live camera feed. Graphical representations of crowd data are also included to help administrators analyse trends and identify peak visiting hours. This dashboard allows administrators to observe the temple environment continuously and take appropriate actions when crowd levels increase.

E. Smart Token Management Integration

To regulate visitor flow and reduce congestion, the system incorporates a smart token generation mechanism. Devotees can obtain tokens through a mobile or web-based interface, where each token is associated with a specific time slot for temple entry. The token information is managed by the backend server and stored in the database.

When the crowd level reaches a high threshold, the system temporarily restricts the generation of new tokens for immediate entry and instead allows booking for future time slots. This mechanism ensures that the number of visitors inside the temple remains within safe limits and improves the overall queue management process.

F. System Integration

The final implementation integrates all components into a unified system. The camera-based AI detection module continuously analyses the crowd, while the ESP32 sensor system tracks entry and exit movements. Both data sources are sent to the Flask backend server, which processes the information and updates the database.

The processed results are displayed on the administrative dashboard, enabling temple authorities to monitor crowd conditions in real time. By combining AI based detection, IoT hardware, backend data processing, and a smart token system, the implementation provides a complete solution for intelligent temple crowd management

III. PROBLEM STATEMENT

Temples and other religious places attract many devotees every day, particularly during festivals, special rituals, and peak visiting hours. Managing such large crowds is a major challenge for temple authorities. Traditional crowd management methods mainly rely on manual supervision by staff and physical queue systems. These conventional approaches are often inefficient and may fail to provide accurate real-time information about the number of people present inside the temple premises. As a result, devotees frequently experience long waiting times, overcrowded queues, and difficulty in accessing temple services smoothly. One of the major problems in temple crowd management is the lack of real-time crowd monitoring systems. Temple authorities usually do not have an automated way to determine how many people are currently inside the temple area. Without proper monitoring tools, it becomes difficult to control the flow of visitors or prevent overcrowding in certain areas. In crowded situations, the absence of accurate crowd data can lead to confusion and poor decision-making, which may affect the safety and comfort of devotees.

Another significant issue is the inefficient management of entry and exit points. In many temples, there is no proper mechanism to track how many people enter or leave the premises at any given time. This lack of monitoring makes it difficult to maintain an accurate count of visitors and increases the risk of congestion near entrance gates. During peak periods, this problem can lead to overcrowding and unsafe conditions. Queue management is also a major challenge in temples. Devotees often must stand in long lines for extended periods to receive darshan. Traditional queue systems do not effectively distribute visitors across different time periods, resulting in uneven crowd distribution throughout the day. This leads to overcrowding during certain hours while other times remain relatively less crowded. Such inefficient visitor distribution reduces the overall efficiency of temple operations and negatively affects the visitor experience. In addition, the absence of an intelligent system that integrates crowd monitoring, entry–exit tracking, and queue management limits the ability of temple authorities to respond quickly to changing crowd conditions. Without real-time information, administrators cannot implement timely measures to regulate visitor flow or prevent excessive crowd build up. Therefore, there is a need for a smart and automated system that can monitor crowd density in real time, track entry and exit movements accurately, and regulate visitor flow through an efficient queue management mechanism. Such a system should combine modern technologies such as Artificial Intelligence, Computer Vision, and Internet of Things (IoT) to provide reliable crowd monitoring and improve overall temple management. The proposed Smart Temple Crowd and Queue Management System aim to address these challenges by providing an intelligent solution for real-time crowd detection, visitor flow monitoring, and digital token-based queue management.

IV. PROPOSED SYSTEM

To address the challenges associated with traditional temple crowd management methods, this research proposes a Smart Temple Crowd and Queue Management System that integrates Artificial Intelligence (AI), Computer Vision, and Internet of Things (IoT) technologies. The proposed system aims to monitor crowd density in real time, track entry and exit movements accurately, and regulate the flow of devotees through an intelligent token-based queue management mechanism. The system utilizes a camera-based AI detection module to analyse live video feeds from inside the temple premises. The captured video frames are processed using the YOLOv8 object detection model, which can identify and count people in real time. By continuously analysing the camera feed, the system estimates the number of individuals present within the temple area and determines the crowd density level. Based on predefined thresholds, the crowd is categorized into levels such as Low, Medium, and High, allowing temple authorities to quickly understand the current crowd situation. In addition to camera-based crowd monitoring, the proposed system incorporates IR sensors connected to an ESP32 microcontroller at the temple entrance and exit points. These sensors detect the movement of devotees entering or leaving the temple premises. The ESP32 processes the sensor signals and maintains an accurate count of entry and exit events. This information helps in calculating the number of people currently inside the temple and complements the AI-based crowd detection module. All the collected data from the AI detection system and IoT sensors is transmitted to a Flask-based backend server. The backend acts as the central processing unit of the system, where incoming data is analysed and stored in a MySQL database. The backend server processes crowd data, determines the current crowd density level, and manages other functionalities such as token generation and queue regulation. To improve the overall queue management process, the proposed system introduces a smart token generation mechanism. Devotees can obtain tokens for temple entry through a digital interface. Each token contains a unique token number and an assigned time slot for darshan. When the crowd density reaches a high level, the system automatically restricts immediate token generation and instead allows devotees to book tokens for future time slots. This approach helps distribute visitor arrivals more evenly throughout the day and prevents overcrowding during peak hours. A web-based administrative dashboard is also implemented to allow temple authorities to monitor the system in real time. The dashboard displays live crowd statistics, entry and exit counts, crowd level indicators, and live camera feeds from the temple premises. By providing clear and real-time information, the dashboard helps administrators make quick decisions to manage the crowd efficiently.

Overall, the proposed system provides an automated, intelligent, and scalable solution for temple crowd management. By combining AI-based crowd detection, IoT-based entry and exit monitoring, and a smart token-based queue system, the solution improves safety, reduces waiting time, and enhances the overall experience for devotees visiting the temple.

V. SYSTEM ARCHITECTURE

The proposed Smart Temple Crowd and Queue Management System is designed using an integrated architecture that combines Artificial Intelligence, IoT-based sensing, backend server processing, and a web-based monitoring dashboard. The system architecture ensures continuous monitoring of temple crowd conditions, efficient data processing, and effective crowd regulation through a smart token management mechanism. The architecture mainly consists of four major layers: Data Acquisition Layer, Processing Layer, Backend Server Layer, and User Interface Layer.

A. Data Acquisition Layer

The Data Acquisition Layer is responsible for collecting real-time information about the number of devotees entering, exiting, and staying inside the temple premises. This layer includes two primary components: camera-based monitoring and sensor-based entry–exit detection. A camera installed inside the temple area captures continuous video footage of the crowd. The video frames are processed using a YOLOv8 deep learning model to detect and count the number of people present in the monitored area. YOLOv8 is an advanced object detection algorithm capable of identifying humans in real time with high accuracy. By analysing each frame of the video stream, the system determines the current number of individuals inside the temple premises. In addition to camera-based monitoring, Infrared (IR) sensors are placed at the entry and exit points of the temple. These sensors detect when a person passes through the entrance or exit gate. The sensors are connected to an ESP32 microcontroller, which records the number of people entering and leaving the temple. This sensor-based monitoring helps maintain an accurate count of visitors and supports the AI-based detection system.

B. Processing Layer

The Processing Layer is responsible for analysing the collected data and calculating the current crowd density. In this layer, the video frames captured by the camera are processed by the YOLOv8 model using OpenCV and Python. The model detects human objects and provides bounding boxes around each detected individual. By counting these detections, the system estimates the number of people present in the camera's field of view. At the same time, the ESP32 microcontroller processes signals from the IR sensors to determine whether a person has entered or exited the temple. The microcontroller updates the entry and exits counts and sends the data to the backend server through network communication. Combining both camera-based and sensor-based data improves the reliability of the crowd estimation process.

C. Backend Server Layer

The Backend Server Layer acts as the central processing and data management unit of the system. A Flask-based backend server receives data from both the AI detection module and the ESP32 sensor system. The server processes this information and calculates the current number of people inside the temple. The backend server also categorizes the crowd level into predefined categories such as Low, Medium, and High based on threshold values. These levels help administrators understand the current crowd situation and take necessary actions when congestion increases. All collected information, including crowd counts, entry and exit data, and system status, is stored in a MySQL database for future analysis. This stored data can be used to generate crowd history reports and identify peak visiting hours.

D. User Interface Layer

The User Interface Layer provides a platform for temple administrators and users to interact with the system. The system includes a web-based admin dashboard that displays real-time crowd statistics, live camera feeds, and graphical representations of crowd data. The dashboard allows administrators to monitor the temple environment continuously and respond quickly to overcrowding situations. In addition, the system supports a mobile application for devotees, where users can book temple visit tokens based on available time slots. The token system helps regulate the number of visitors entering the temple at a given time, reducing congestion and improving queue management.

E. Overall System Workflow

The overall workflow of the system begins with real-time data collection through cameras and sensors. The collected data is then processed by the AI detection module and the ESP32 microcontroller. The processed information is transmitted to the Flask backend server, where crowd density is calculated and stored in the database. Finally, the results are displayed on the admin dashboard and mobile application, enabling efficient monitoring and crowd management. Through this integrated architecture, the system provides a real-time**,** automated, and intelligent solution for temple crowd monitoring and queue management, improving safety and operational efficiency while enhancing the overall experience for devotees.

VI. METHODOLOGY

The methodology of the proposed Smart Temple Crowd and Queue Management System describes the process used to monitor, analyse, and manage crowd density within temple premises. The system integrates Artificial Intelligence, Computer Vision, Internet of Things (IoT) sensors, and a web-based monitoring platform to provide real-time crowd information and improve queue management. The methodology consists of several stages, including data acquisition, crowd detection, sensor-based monitoring, data processing, and token management.

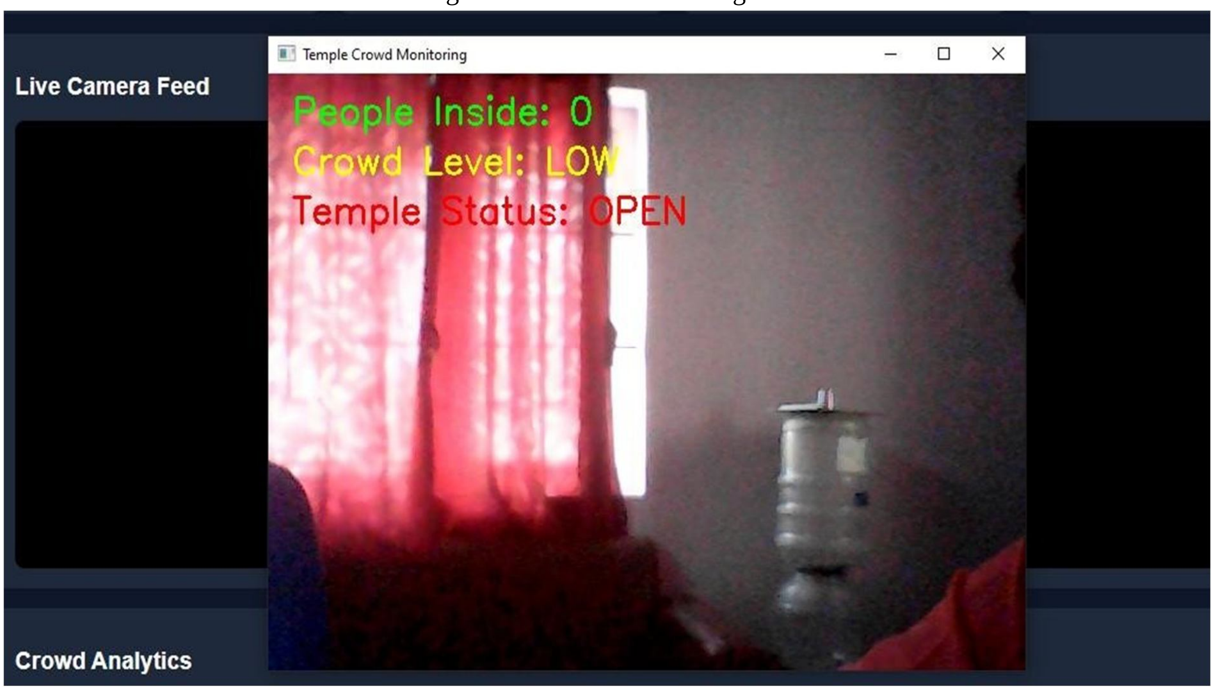
A. Data Acquisition

The first step in the methodology involves collecting real-time data related to the number of people inside the temple premises. This data is obtained through two main sources: camera-based monitoring and sensor-based entry–exit detection. A camera installed in the temple area continuously captures live video footage of devotees moving within the premises. These video frames serve as the input for the AI-based crowd detection module. In addition to the camera system, Infrared (IR) sensors are placed at the entry and exit points of the temple. These sensors detect the movement of individuals passing through the gates. When a person crosses the entry sensor, the system increments the entry count, and when a person crosses the exit sensor, the exit count is increased. The sensors are connected to an ESP32 microcontroller, which collects the sensor signals and transmits the data to the backend system.

B. Crowd Detection Using YOLOv8

The captured video frames are processed using a YOLOv8 (You Only Look Once version 8) deep learning model. YOLOv8 is a real-time object detection algorithm capable of detecting multiple objects in a single frame with high speed and accuracy. In this system, the model is trained to detect human objects within the camera feed. Each frame from the video stream is analysed, and the model identifies individuals by drawing bounding boxes around detected persons. The total number of detected individuals is counted and used to estimate the crowd density in the monitored area. This process allows the system to determine how many people are currently present inside the temple environment.

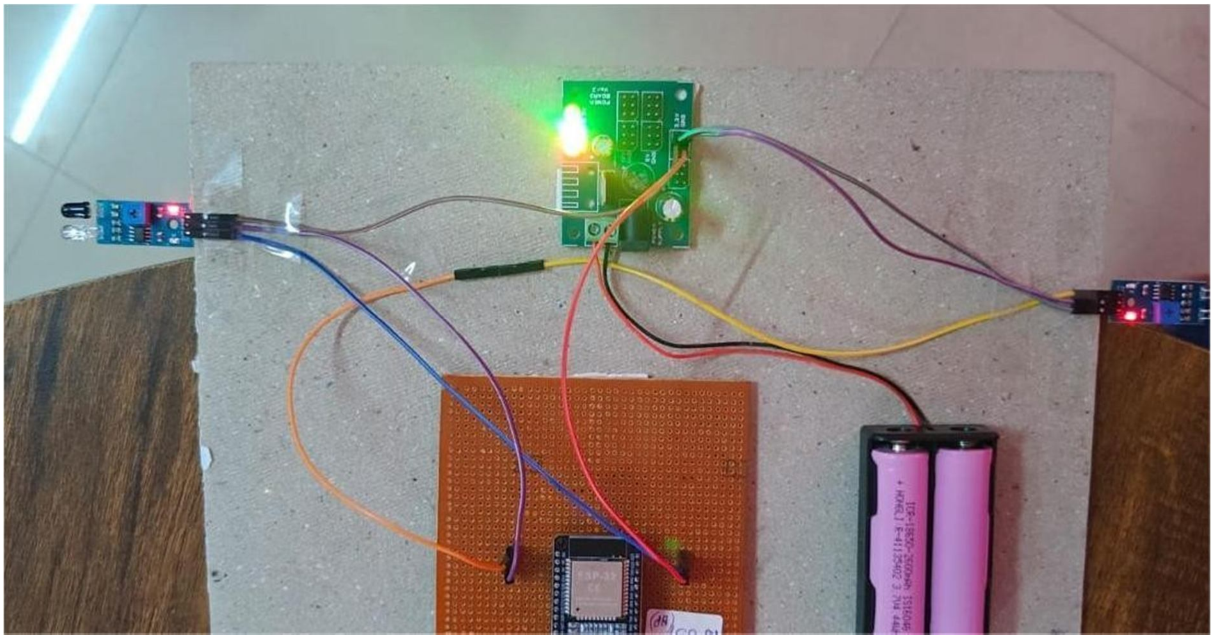
Fig 1. Crowd Detection Using YOLOv8



C. Sensor-Based Entry and Exit Monitoring

To improve the accuracy of crowd estimation, the system integrates IR sensors connected to an ESP32 microcontroller at temple entrances and exits. These sensors detect the movement of devotees entering or leaving the temple premises. When a person crosses the sensor beam, the microcontroller records the event and updates the corresponding entry or exit counter. The entry and exit counts help maintain a running estimate of the total number of people inside the temple. This sensor-based monitoring complements the AI-based crowd detection system and ensures more reliable crowd tracking. Fig 2 this describes the structure of sensors along with ESP32.

Fig 2. diagram of sensors with ESP32



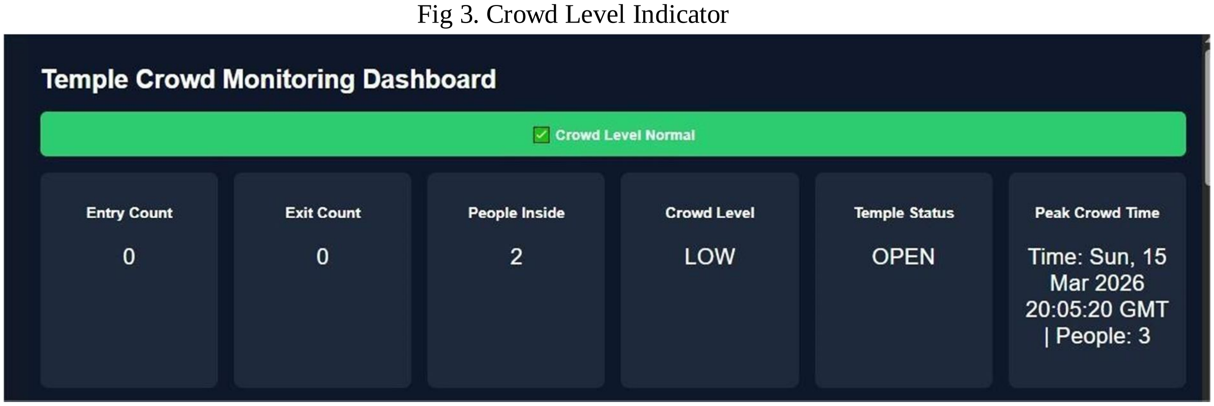
D. Data Processing and Crowd Level Classification

The data obtained from the AI detection module and the sensor system is transmitted to a Flask-based backend server. The server processes the incoming data and calculates the total number of people present inside the temple premises. Based on this value, the system classifies the crowd density into predefined categories such as:

- Low Crowd Level – Normal visitor flow with minimal congestion.
- Medium Crowd Level – Increased visitor flow requiring moderate crowd control.
- High Crowd Level – Heavy congestion where entry regulation is required.

This classification allows temple administrators to understand the current crowd condition and take appropriate actions when necessary.

Fig 3. Crowd Level Indicator



E. Smart Token Management System

To regulate the flow of devotees and prevent overcrowding, the system introduces a smart token generation mechanism. Devotees can obtain tokens that include a specific time slot for temple entry. This method helps distribute visitors throughout the day and reduces long waiting queues. When the crowd density exceeds a predefined threshold, the system temporarily stops issuing immediate tokens and instead allows booking for future time slots. This ensures that the number of visitors entering the temple remains within manageable limits and helps maintain a safe environment.

Fig 4. Token Queue Management

Token Queue Management

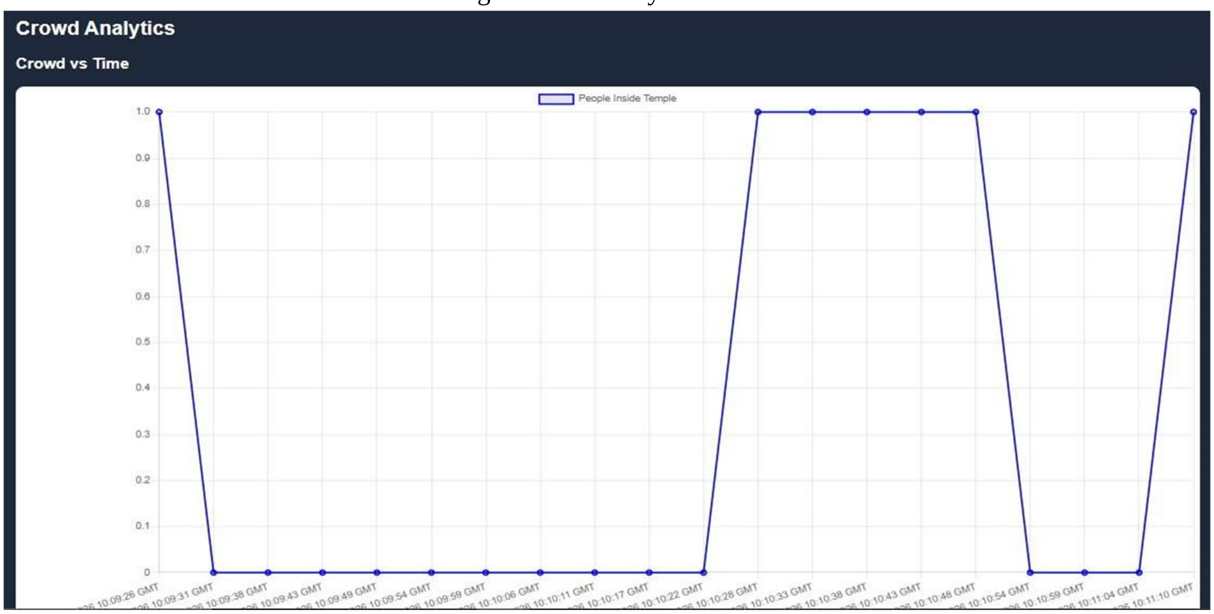
Token	Members	Time Slot	Status	Action
T105	1	6:00 - 6:30 PM	Called	Complete
T104	5	7:30 - 8:00 PM	Completed	Completed
T103	4	7:00 - 7:30 AM	Completed	Completed
T102	3	8:30 - 9:00 AM	Completed	Completed
T101	4	8:00 - 8:30 AM	Completed	Completed

F. Real-Time Monitoring and Visualization

The processed data is displayed through a web-based administrative dashboard. This dashboard provides real-time information such as the current crowd count, entry and exit statistics, crowd level status, and live camera feed.

Administrators can monitor these details continuously and respond quickly if the crowd density becomes too high. Through this methodology, the system combines AI-based crowd detection, IoT sensor monitoring, backend data processing, and a smart queue management system to provide an effective and automated solution for temple crowd management.

Fig 5.Crowd Analysis



VII. RESULTS AND DISCUSSION

The proposed Smart Temple Crowd and Queue Management System was developed and tested to evaluate its effectiveness in monitoring and managing crowd density within temple premises. The system integrates Artificial Intelligence, IoT sensors, and a web-based monitoring platform to provide real time insights into crowd conditions. The implementation demonstrates that the integration of computer vision and sensor-based technologies can significantly improve the efficiency of temple crowd management. During testing, the YOLOv8-based crowd detection module successfully detected and counted individuals from live camera feeds. The model was able to identify people in real time and draw bounding boxes around detected individuals. The total number of detected persons was continuously updated and displayed on the administrative dashboard. This allowed administrators to monitor the crowd situation dynamically and observe changes in crowd density as people moved within the temple premises. The IR sensor and ESP32 - based entry – exit monitoring system also performed effectively during testing. The sensors accurately detected movements at the entrance and exit points and transmitted the data to the backend server. By combining the entry and exit counts, the system was able to maintain an approximate count of the number of devotees currently present inside the temple. This information complemented the camera-based detection system and helped improve the reliability of crowd monitoring.

The backend server developed using the Flask framework successfully received and processed data from both the AI detection module and the sensor module. The system analysed the collected data and categorized the crowd density into different levels such as Low, Medium, and High. These crowd levels were displayed on the administrative dashboard, enabling temple authorities to quickly understand the current situation and take appropriate actions if necessary. The web based administrative dashboard provided a centralized platform for monitoring all system activities. It displayed live camera feeds, current crowd count, entry and exit statistics, and crowd level indicators. The dashboard interface allowed administrators to observe crowd trends and monitor the system in real time. This real-time visibility helps temple management respond quickly to overcrowding situations and implement necessary control measures. The smart token management mechanism also contributed to improving queue efficiency. When the crowd level reached a predefined high threshold, the system restricted immediate token generation and allowed devotees to book tokens for future time slots. This approach helped distribute visitor flow more evenly throughout the day and reduced congestion during peak hours. Overall, the experimental results indicate that the proposed system can effectively assist temple authorities in managing large crowds. By combining AI-based people detection, IoT-based entry–exit tracking, and intelligent queue management, the system improves crowd monitoring accuracy and operational efficiency. The results demonstrate that such smart technologies can play an important role in enhancing safety, reducing waiting times, and improving the overall experience for devotees visiting temples.

VIII. CONCLUSION

Managing large crowds in temples is a challenging task, especially during festivals, weekends, and special religious events. Traditional crowd management methods mainly rely on manual supervision and conventional queue systems, which often result in overcrowding, long waiting times, and inefficient visitor flow. These limitations highlight the need for an intelligent and automated system capable of monitoring and controlling crowd density effectively. This research presented a Smart Temple Crowd and Queue Management System that integrates Artificial Intelligence, Computer Vision, and Internet of Things (IoT) technologies to improve temple crowd monitoring and queue management. The proposed system utilizes a YOLOv8-based computer vision model to detect and count people from live camera feeds, enabling real-time estimation of crowd density within temple premises. In addition, IR sensors connected to an ESP32 microcontroller monitor entry and exit movements at temple gates, allowing the system to maintain an accurate count of visitors entering and leaving the temple. The collected data from the AI detection module and sensor system is processed through a Flask-based backend server and stored in a MySQL database. Based on predefined thresholds, the system categorizes the crowd density into levels such as Low, Medium, and High, which helps administrators quickly understand the current crowd situation. A web-based administrative dashboard provides realtime visualization of crowd statistics, entry–exit counts, and live camera feeds, enabling temple authorities to monitor and manage the crowd effectively. Furthermore, the system incorporates a smart token-based queue management mechanism that regulates the flow of devotees entering the temple. When the crowd level becomes high, the system restricts immediate token generation and instead allows booking for future time slots. This approach helps distribute visitors more evenly throughout the day and prevents excessive crowd accumulation during peak hours.

Overall, the proposed system demonstrates how modern technologies can be effectively applied to improve temple crowd management. By combining AI based detection, IoT-based monitoring, and intelligent queue regulation, the system enhances safety, reduces waiting time, and provides a more organized experience for devotees. The research highlights the potential of smart systems in managing large gatherings and contributes toward the development of efficient crowd management solutions for religious places.

IX. FUTURE WORK

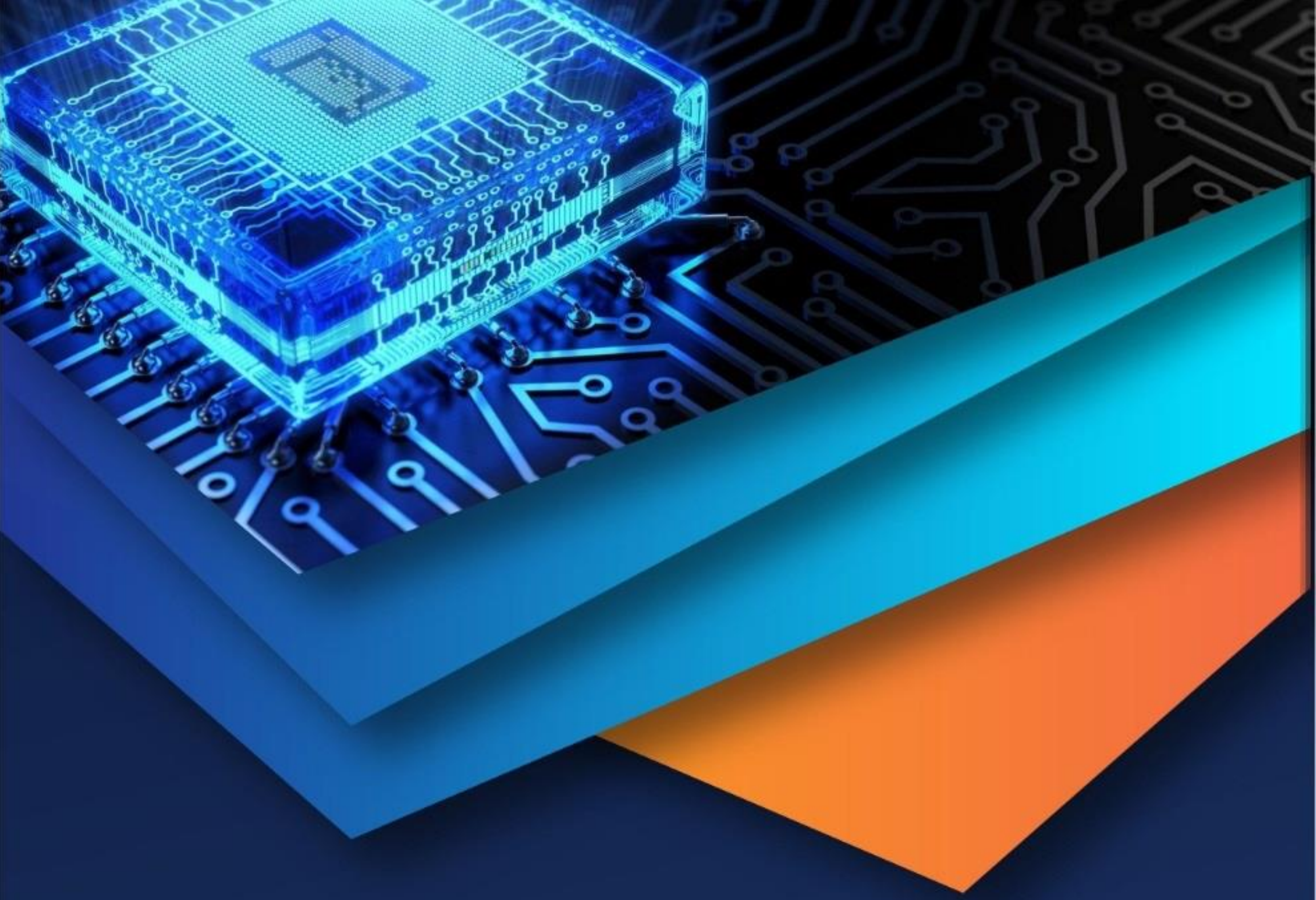
Although the proposed Smart Temple Crowd and Queue Management System demonstrate an effective approach for monitoring and managing temple crowds, there are several opportunities for further improvements and enhancements. Future work can focus on expanding the system capabilities, improving accuracy, and integrating additional smart technologies to make the system more efficient and scalable for large temple environments. One possible enhancement is the development of a dedicated mobile application for devotees. Through the mobile app, visitors could check realtime crowd status, book tokens for temple entry, receive notifications about their scheduled time slots, and track their token status. This would provide greater convenience for devotees and reduce the need for physical queues at the temple.

Another area for future improvement is the integration of predictive analytics for crowd forecasting. By analysing historical crowd data stored in the database, machine learning models could be used to predict peak visiting hours and expected crowd levels on specific days or festivals. This predictive capability would help temple authorities plan better crowd control strategies and allocate resources more efficiently.

The system can also be enhanced by supporting multiple camera inputs and multi-location monitoring within the temple premises. Large temples often have several entry points and worship areas. Integrating multiple cameras and sensor systems would allow the system to monitor crowd distribution across different sections of the temple and provide more detailed crowd analysis. Future research could also explore the implementation of automated alert and notification systems. When crowd density reaches a critical level, the system could automatically send alerts to temple administrators through SMS, email, or mobile notifications. This would allow authorities to take immediate action to control the situation and ensure visitor safety. Additionally, further improvements can be made by enhancing the accuracy and efficiency of the AI detection models. Training the detection system with more diverse datasets and optimizing the model for real-time performance could improve detection reliability in crowded environments. Finally, the system could be extended to support integration with smart city infrastructure or government monitoring platforms. Such integration would enable better coordination during large religious gatherings and festivals where crowd management becomes critical for public safety. Overall, these future enhancements would make the system more robust, scalable, and user-friendly, ultimately contributing to safer and more efficient crowd management in temples and other large religious gathering places.

REFERENCES

[1] YOLOv8. YOLOv8 Documentation. Ultrathick. Available: https://docs.ultralytics.com
[2] OpenCV. Open-Source Computer Vision Library. Available: https://opencv.org
[3] Flask. Flask Web Development Documentation. Available: https://flask.palletsprojects.com
[4] MySQL. MySQL 8.0 Reference Manual. Oracle Corporation. Available: https://dev.mysql.com/doc
[5] ESP32. ESP32 Technical Reference Manual. Expressive Systems. Available: https://www.espressif.com
[6] S. Zhang, R. Benenson, and B. Schiele, "City Persons: A Diverse Dataset for Pedestrian Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
[8] A. Krizhevsky, I. Stever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, 2012.
[9] M. Rouse and D. Wigmore, "Internet of Things (IoT): Definition and Applications," TechTarget Research, 2021.
[10] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.



10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY