iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73397

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Real-Time Restaurant Table Occupancy Monitoring System

Vasudeva Rao Paradesi[1], Mrs. Ambati Tulasi[2] Assistant Professor (ADHOC)
[1,2]*Department of Information Technology & Computer Applications, Andhra University College of Engineering, Visakhapatnam, India*

Abstract: ***In the fast-evolving landscape of intelligent restaurant management, adopting automation technologies is pivotal to improve operational efficiency and customer experience. Traditional methods of table occupancy detection, which rely on physical sensors, often suffer from high installation costs, limited flexibility, and poor scalability.***
This research proposes an innovative, vision-based real-time monitoring system for restaurants. The system leverages the YOLOv8 object detection model to accurately identify tables and human presence, integrated seamlessly with a Django-powered multi-tenant web dashboard. Each restaurant can register and configure its custom table layout, which is then updated in real-time based on live camera feeds.
By replacing expensive hardware with a robust AI-driven system, the solution enhances accessibility and scalability. The dashboard offers a user-friendly interface, real-time status updates, and support for simultaneous multi-user access. Extensive testing under varying lighting conditions and scenarios demonstrates the system's accuracy, reliability, and real-world applicability. The system aims to provide a comprehensive and cost-effective alternative to traditional sensor-based solutions, making advanced occupancy monitoring accessible to a wider range of restaurant businesses.
Keywords: Table occupancy detection, YOLOv8, real-time monitoring, Django, computer vision, restaurant automation, object detection, Oracle SQL, deep learning, web dashboard.

I. INTRODUCTION

The study on this topic presents a detailed framework that combines the high-speed object localization capabilities of YOLOv8 with a Django-based dashboard. The system is designed for multi-tenant use, enabling restaurants to independently configure their table layouts and monitor occupancy data in real-time through a seamless, unified interface. This addresses a critical need in the hospitality sector for efficient and dynamic table management.

Traditional occupancy detection solutions often rely on motion or pressure sensors, which, while effective in small-scale environments, are financially impractical and inflexible for dynamic restaurant settings. These systems typically involve significant upfront investment in hardware, complex wiring, and ongoing maintenance, limiting their adoption by smaller or independent establishments.

By leveraging advancements in deep learning and computer vision, this system eliminates the need for physical sensors, offering a more scalable, adaptable, and cost-efficient solution. This paradigm shift not only reduces initial setup costs but also provides greater flexibility for reconfiguring layouts without physical hardware adjustments, thus supporting dynamic business operations. The integration of a web dashboard further empowers restaurant staff with immediate insights, enabling proactive management decisions

II. LITERATURE REVIEW / RELATED WORK

A. Sensor-Based Occupancy Detection

Early approaches to occupancy monitoring relied heavily on infrared, ultrasonic, or pressure sensors embedded in furniture. While effective for binary detection (occupied/unoccupied), these systems were costly to install and maintain. The installation often required specialized technicians and disruption to existing infrastructure. Moreover, their inflexible infrastructure limited adaptability to different seating arrangements and real-time feedback capabilities. Any change in table layout necessitated re-installation or recalibration of sensors, making them unsuitable for dynamic restaurant environments where layouts frequently change for events or different dining experiences. Data collection was also often limited to simple occupied/unoccupied states, lacking the richness required for advanced analytics.

©IJRASET: All Rights are Reserved

B. RFID and IOT-Based Systems

Some systems implemented RFID-tagged furniture or wearable devices for customers. Though offering improved granularity in data collection, these solutions required users or furniture to be tagged and monitored continuously, which posed practicality and privacy concerns in casual dining environments. Customers might find it intrusive to wear RFID tags, and the cost of tagging every piece of furniture and managing the RFID infrastructure can be prohibitive. Furthermore, the accuracy of RFID systems can be affected by signal interference and the density of tags, leading to unreliable data in busy settings. The constant monitoring also raised ethical questions regarding customer privacy, which is a significant consideration for public-facing businesses.

C. Vision-Based Detection Approaches

Recent developments in deep learning have introduced computer vision models such as YOLOv3 and YOLOv5 for occupancy detection in cafes, offices, and waiting areas. These models demonstrated the potential of using cameras for detecting people and objects, offering a non-intrusive alternative to sensors. While effective in static monitoring setups, many lacked a real-time dynamic interface and multi-user dashboard support, limiting their broader adoption in commercial restaurants. Previous vision- based systems often focused on single-camera setups or were not designed with the multi-tenant architecture required for diverse restaurant businesses. Our system builds upon these advancements with YOLOv8 and Django, enabling responsive monitoring across independent tenants by providing a centralized yet customizable platform. This allows each restaurant to maintain its unique layout and data while benefiting from a robust underlying technology.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system adopts a modular and scalable architecture, leveraging a combination of cutting-edge deep learning for object detection and a robust web framework for data management and visualization. This design ensures both high accuracy in real-time detection and a user-friendly, secure interface for restaurant administrators.

A. User Registration And Table Configuration

This system provides a secure registration form and authentication form using Django built-in authentication module. This ensures that only authorized persons can access and manage restaurant-specific data. Restaurant administrators can log in, define their table layouts by assigning names, coordinates, and seating capacity. This intuitive configuration process allows for easy setup and modification of floor plans. These configurations are stored securely in the Oracle SQL database, which provides a reliable and scalable backend for managing diverse restaurant data. The database schema is designed to efficiently store and retrieve table information, user credentials, and historical occupancy data, ensuring data integrity and quick access.

B. Real-Time Detection Using Yolov8

YOLOv8, a state-of-the-art object detection framework developed by Ultralytics, is central to the system's real-time monitoring capabilities. It is trained to identify two key classes: people and tables. Frames captured from the live video feed from strategically placed cameras are processed in real time using OpenCV and custom backend logic. OpenCV facilitates video capture, frame processing, and image manipulation, making it an ideal choice for this application. The system then analyzes the bounding boxes of detected objects (people and tables) to determine occupancy using an Intersection over Union (IoU) function.

```
def calculate_iou(rect1, rect2):
left = max(rect1[0], rect2[0]) top = max(rect1[1], rect2[1]) right = min(rect1[2], rect2[2]) bottom = min(rect1[3], rect1[3])
overlap_width1 = max(0, right - left + 1) overlap_height1 = max(0, bottom - top + 1)
overlap_area = overlap_width * overlap_height
area1 = (rect1[2] - rect1[0] + 1) * (rect1[3] - rect1[1] + 1) area2 = (rect2[2] - rect2[0] + 1) * (rect2[3] - rect2[1] + 1)
union_area1 = float(area11 + area12 - overlap_area1 if union_area == 0:
return 0.0
return overlap_area / union_area A threshold IoU value (typically
```

text0.1) is used to determine if a person's bounding box significantly overlaps with a table's bounding box, indicating occupancy. This threshold can be fine-tuned to optimize accuracy based on camera angles and restaurant layouts. Each occupancy status is then sent to the server using a RESTful API endpoint for further processing and dashboard updates.

C. Occupancy Logic And Server Communication

The Python script, yolo_update.py, runs independently to process the detections from the YOLOv8 model. This script computes the occupancy status for each configured table based on the IoU calculations and sends HTTP POST requests to the Django backend with updated status values. Each request includes essential information such as the restaurant ID, table ID, and the current occupancy status (e.g., occupied or free). This asynchronous communication ensures that the detection and dashboard update processes are decoupled, improving overall system responsiveness. The server then processes these requests and updates the corresponding table status in the Oracle SQL database, which in turn reflects on the web dashboard. This design minimizes latency and ensures near real-time updates.

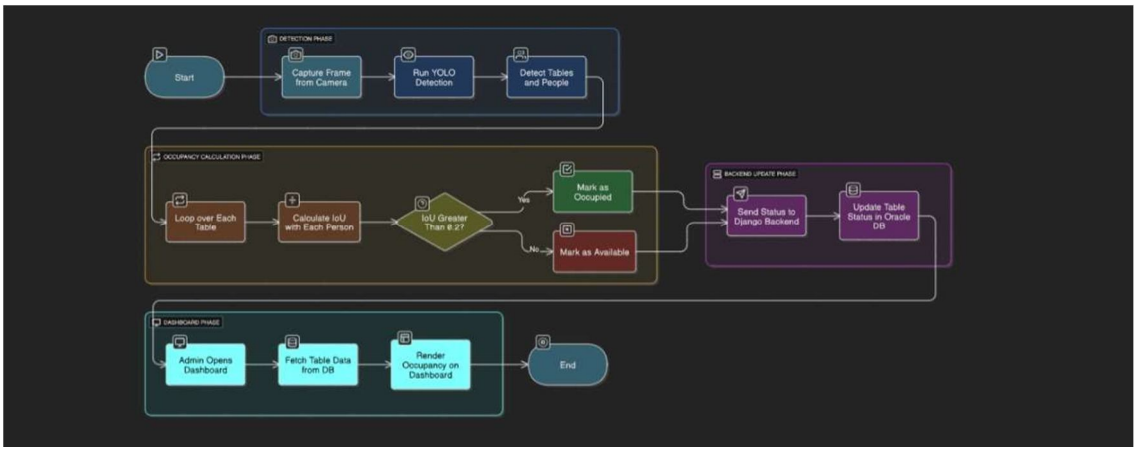


Figure 1: Control flow of Table Occupancy Detection

D. Real-Time Dashboard Visualization

Django serves a dynamic web page dashboard that visually represents each restaurant's table layout. This dashboard is the primary interface for restaurant staff to monitor table occupancy. Color codes are used to provide instant visual feedback: green for free, red for occupied. This intuitive color-coding allows for quick assessment of the restaurant's current status. Additional planned states include orange for reserved (allowing staff to pre-assign tables) and blue for cleaning (indicating a table is temporarily unavailable). The dashboard supports live updates through technologies like AJAX (and planned WebSockets for even faster updates), ensuring that the displayed information is always current. Beyond real-time status, the dashboard also offers historical analytics features, enabling managers to review past occupancy patterns, identify peak hours, and optimize staffing. Furthermore, administrative features allow for user management, camera configuration, and system settings adjustments, making it a comprehensive management tool.

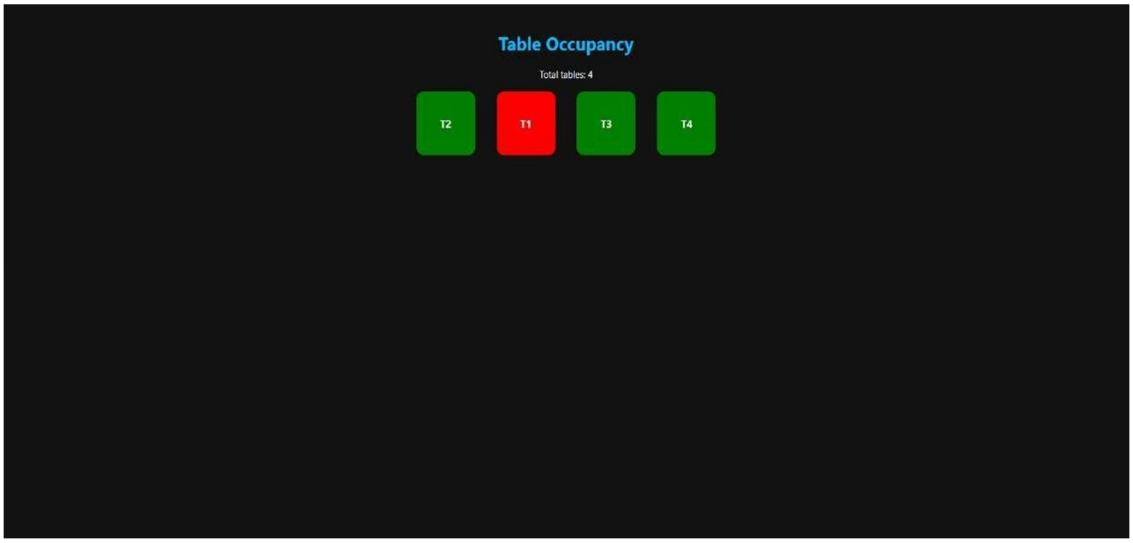


Figure 2: Dashboard view

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

IV. IMPLEMENTATION AND TECHNOLOGIES USED

This system's implementation relies on a carefully selected stack of modern technologies,durability, ensuring robustness, scalability, and ease of maintenance.

Component	Technology	Description
Web Framework	Django (Python)	User authentication, routing, API backend
Frontend	HTML/CSS/JavaScript	Dashboard UI and layout setup
Object Detection	YOLOv8model (Ultralytics)	Real-time object detection
Database	Oracle SQL	Storage for layouts, logs, and status tracking
Live Feed Access	OpenCV	Captures and streams camera frames
API Integration	Django REST Framework	Endpoint /update/ for occupancy data

V. PERFORMANCE EVALUATION

To validate the system's effectiveness, extensive performance evaluations were conducted under various real-world conditions. These tests focused on detection accuracy, adaptability, responsiveness, and stability of the occupancy status.

A. Detection Accuracy

Tests were conducted across various lighting conditions and seating arrangements, mimicking diverse restaurant environments. The results demonstrate high accuracy rates, validating YOLOv8's capability in different scenarios.

Scenario	Tables	Accuracy	Frame Time
Real Environment (6)	6	93.2%	0.9 sec
Static Image (18)	18	91.7%	1.1 sec
Mixed Lighting (12)	12	88.5%	1.4 sec

The "Real Environment" scenario involved live video feeds from a functioning restaurant setting, highlighting the system's practical performance. "Static Image" tests involved a larger number of tables in a controlled environment, demonstrating scalability. "Mixed Lighting" evaluated robustness under challenging conditions, where shadows or varying light levels could potentially affect detection. The results indicate that the system maintains a high level of accuracy even in less-than-ideal circumstances.

B. Adaptability And Responsiveness

The system demonstrates robustness under variable conditions including partial occlusion (e.g., a person partially hidden by another object) and inconsistent lighting. This is crucial for real-world restaurant environments where lighting can change throughout the day and customer movements can be unpredictable. Though frame processing ranges between text0.9–1.4 seconds, it suffices for near real-time updates. While instantaneous updates are ideal, a delay of slightly over a second is generally acceptable for table occupancy monitoring, as table status typically doesn't change every split second. This performance allows restaurant staff to make informed decisions without significant lag.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

C. Fluctuation Filtering

To avoid rapid status shifts due to transient frame inconsistencies (e.g., a person briefly walking past a table or a momentary detection error), occupancy decisions are buffered over several frames. This means the system doesn't immediately change a table's status based on a single frame's detection. For example, if a person exits a table view briefly, the system waits for consistent absence over a predefined number of consecutive frames before marking it unoccupied. This temporal filtering mechanism significantly enhances the stability and reliability of the reported occupancy status, preventing "flickering" on the dashboard and providing more dependable information to users.

VI. SECURITY AND DATA INTEGRITY

Security and data integrity are paramount, especially in a multi-tenant system dealing with potentially sensitive operational data. The system incorporates several measures to protect user information and maintain data accuracy.

Security measures include:

Hashed password storage: User passwords are not stored in plain text. Instead, they are hashed using strong cryptographic algorithms (e.g., PBKDF2 with SHA256) and salted, making them resistant to brute-force attacks and dictionary attacks even if the database is compromised.

CSRF (Cross-Site Request Forgery) token validation for form-based actions: Django's built-in CSRF protection is utilized for all web forms, preventing malicious websites from tricking authenticated users into submitting requests they did not intend.

Input sanitization: All user inputs are carefully sanitized to prevent common web vulnerabilities such as SQL injection and cross- site scripting (XSS) attacks. This ensures that malicious code cannot be injected into the system through user input fields.

A. Tenant Data Isolation

Only authenticated users can access their restaurant’s data, ensuring strict tenant data isolation. The system is designed to logically separate data for each registered restaurant, preventing one restaurant's administrators from viewing or modifying another's data. This multi-tenancy model is crucial for commercial deployment, guaranteeing privacy and data security for each client.

B. Api Endpoint Security

The /update/ API endpoint, which receives occupancy data from the YOLOv8 processing script, is optimized for performance and might bypass standard CSRF token validation for efficiency in real-time data submission. However, it incorporates robust internal validation mechanisms. It validates incoming data structure and IDs (e.g., restaurant ID, table ID, and occupancy status) before any database access. This ensures that only well-formed and legitimate data can be processed, mitigating risks associated with malformed or unauthorized requests. This balance between performance and security is critical for a real-time table detection system.

VII. CONCLUSION AND FUTURE WORK

The proposed system significantly improves table occupancy monitoring in restaurants using an efficient, cost-effective, and scalable approach. By replacing traditional, expensive, and inflexible sensor-based systems with an AI-driven vision-based solution, it offers a modern and adaptable alternative for restaurant management. YOLOv8’s accuracy in object detection combined with Django’s versatility in web development results in a robust platform suitable for real-world deployment. The multi-tenant architecture, real-time dashboard, and secure data handling capabilities make it a comprehensive solution for enhancing operational efficiency and customer experience in the hospitality industry.

A. Future Developments Include

Live status updates via AJAX/WebSockets: Transitioning from periodic HTTP POST requests to more persistent connections like WebSockets will provide instantaneous updates on the dashboard, significantly improving responsiveness and user experience.

Heatmap analytics to understand table usage patterns: Implementing a feature to visualize table usage over time through heatmaps can provide valuable insights into popular tables, peak hours, and underutilized areas, aiding in layout optimization and marketing strategies.

©IJRASET: All Rights are Reserved

Facial recognition integration for guest check-in (with privacy safeguards): While this offers advanced personalization and streamlined check-ins, it would require stringent adherence to privacy regulations (e.g., GDPR, CCPA) and clear consent mechanisms to address ethical concerns.

Mobile app for on-the-go access by restaurant managers: Developing a dedicated mobile application would allow managers to monitor table occupancy, receive alerts, and manage settings remotely, providing greater flexibility and responsiveness.

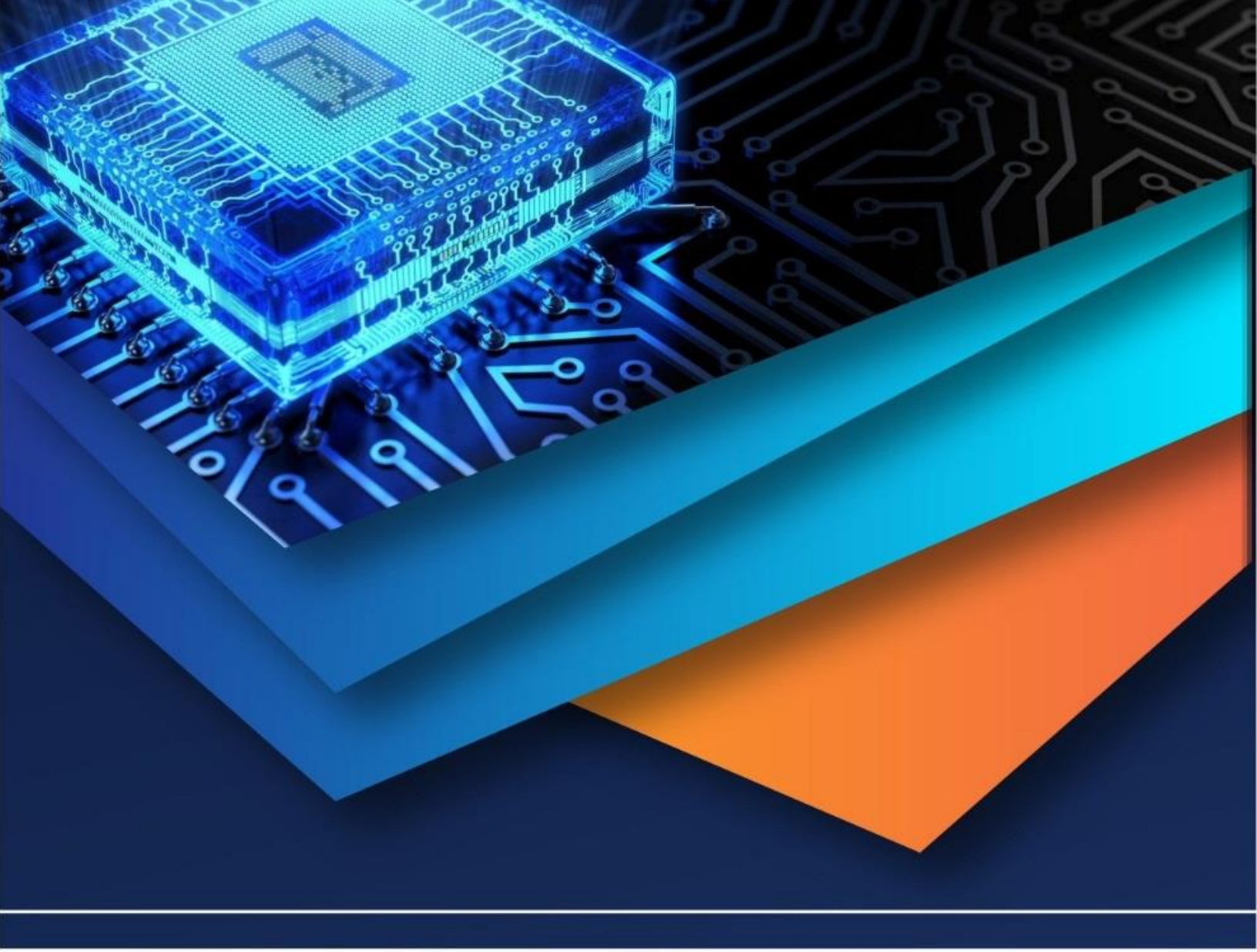
Multi-camera coordination for large or multi-floor premises: Implementing algorithms to intelligently combine data from multiple cameras across larger spaces or different floors would enable comprehensive monitoring of expansive restaurant environments. This would involve managing overlapping camera views and stitching them for a unified view.

POS integration for auto-status change on order/payment events: Connecting the system with Point of Sale (POS) systems could enable automatic updates to table status (e.g., marking a table as occupied when an order is placed, or free after payment is processed), reducing manual intervention and improving accuracy.

AI-based forecasting of occupancy trends: Leveraging historical data with machine learning models to predict future occupancy trends could help restaurants optimize staffing, inventory, and marketing efforts, leading to more efficient operations and increased revenue. This could involve predicting busy periods or identifying days with lower footfall.

REFERENCES

[1] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767 Ultralytics YOLOv8 Documentation
[2] Django Project Documentation Oracle SQL Developer Help
[3] Liu, B., et al. (2021). Vision-Based Seat Occupancy in Smart Cafes, IEEE Access, 9, 87455–87463



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)