International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AI-Based Employee Monitoring System

Pradeep Gowda
Head of the Department Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Bangalore 562157

Ushashree B N
Assistant Professor Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Bangalore 562157
Ushashreebn@gmail.com

Chirag M
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Banaglore 562157
chiragchiru458@gmail.com

Kavana S
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Bangalore 562157
kavanashinde16@gmail.com

Nethra G
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Bangalore 562157
nethragnethrag322@gmail.com

Pallavi V N
Dept. of CSE
Sri Revana Siddeshwara Institute of Technology, Chikkajala, Bangalore 562157
pallavivn06@gmail.com)

https://doi.org/10.55041/ijsmt.v2i11.039

Cite this Article: Gowda, P., N, U. B., M, C., S, K., G, N. & N, P. V. (2026). AI-Based Employee Monitoring System. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.039

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

***Abstract*—** Urban traffic management requires timely information about road conditions and vehicle movement. Conventional approaches based on fixed traffic schedules or manual observation may not respond effectively to variations in traffic volume. This paper presents an integrated system for traffic flow prediction and automated vehicle detection using machine learning and computer vision. A Random Forest classifier is trained using historical traffic data to categorize traffic conditions into four classes: Low, Normal, High, and Heavy. The model uses temporal information and vehicle-count features to estimate the traffic condition. To support automated data acquisition, the system incorporates YOLOv8 for detecting and counting cars, motorcycles, buses, and trucks from uploaded images and live webcam frames. The two components are integrated through a Flask-based web application that provides manual prediction, vehicle detection, live webcam monitoring, and recent activity tracking. Experimental evaluation on the selected dataset produced a classification accuracy of 99.66%. The results demonstrate the feasibility of connecting computer-vision-based vehicle counting with machine-learning-based traffic classification in a unified application. The proposed system provides a practical foundation for accessible traffic monitoring and can be extended with larger datasets, continuous video processing, and persistent data storage.

Index — Traffic flow prediction, Random Forest, YOLOv8, vehicle detection, computer vision, Flask, smart cities.

I. INTRODUCTION

Background and Motivation: Traffic congestion is a major challenge in rapidly growing urban areas. Increasing numbers of vehicles, variations in travel patterns, and changing road conditions make traffic monitoring a complex task. Traditional traffic management approaches often depend on fixed signal timings and manual observation, which may not adapt efficiently to changing traffic conditions. Machine learning techniques can analyze historical traffic information and identify patterns associated with different traffic conditions. Algorithms such as Random Forest can use temporal and vehicle-related features to classify traffic conditions based on previously observed data. However, prediction systems depend on the availability of reliable input data. Computer vision provides another important capability for traffic monitoring. Object detection models can identify vehicles directly from images or video streams and estimate the number of vehicles present in a scene. YOLO-based

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

models are widely used for real-time object detection because they can identify multiple objects within a single frame. Although traffic prediction and vehicle detection can be implemented independently, integrating these functions can provide a more complete traffic monitoring workflow. Vehicle counts obtained through computer vision can serve as inputs for a traffic prediction model, reducing dependence on manual data entry.

Project Objective and Scope: The objective of this work is to develop an integrated system that combines machine learning-based traffic classification with automated vehicle detection. The main objectives are: 1. To train a Random Forest model for classifying traffic conditions into Low, Normal, High, and Heavy categories.
2. To use YOLOv8 to detect and count cars, bikes, buses, and trucks from visual traffic data. 3. To integrate the prediction and detection modules into a Flask-based web application. 4. To support manual input, uploaded media processing, and webcam-based vehicle detection. 5. To provide a unified interface for viewing prediction results and recent system activity.

Contribution: The primary contribution of this work is the implementation of an integrated traffic monitoring framework that connects automated vehicle detection with machine-learning-based traffic condition classification. Unlike a system that performs only prediction or only object detection, the proposed application allows vehicle counts obtained through YOLOv8 to be transferred to the traffic prediction workflow. The work also demonstrates how open-source machine learning and computer vision tools can be integrated into an accessible web-based application.

II. RESEARCH GAP

Research Gap: Existing research in intelligent transportation systems includes both machine-learning-based traffic prediction and computer-vision-based vehicle detection. However, these functions are often developed as separate components.

Traffic prediction studies generally focus on analysing historical data and evaluating the performance of different machine learning algorithms. On the other hand, vehicle detection studies concentrate on identifying and counting vehicles from images or video streams. In many implementations, the output from a detection model is not directly integrated with a traffic prediction system.

The research gap addressed in this project is the development of a unified and accessible system that combines automated vehicle detection with traffic condition prediction.

Feature Comparison of Existing Approaches and the Proposed System

System Capability	Traffic Prediction Studies	Vehicle Detection Studies	Proposed System
Traffic condition prediction	✓	✗	✓
Automated vehicle detection	✗	✓	✓
Visual input processing	Limited	✓	✓
Web-based interface	Limited	Limited	✓
Integrated prediction and detection	✗	✗	✓

The proposed system addresses this gap by integrating a Random Forest classification model with YOLOv8 vehicle detection in a single Flask-based application. The system allows vehicle counts obtained from visual input to support the traffic prediction process.

III. PROBLEM DEFINITION AND SOLUTION

Problem: Urban traffic monitoring often relies on manual observation and fixed traffic control methods, which may not respond effectively to changing traffic conditions. In addition, traffic prediction systems that require manual vehicle-count input can limit real-time usability.

Proposed Solution: The proposed system integrates YOLOv8-based vehicle detection with a Random Forest traffic classification model. Vehicle counts obtained from images or webcam frames are used to support traffic condition prediction. A Flask-based web application provides a unified interface for detection, prediction, and monitoring.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

IV. PROPOSED SYSTEM

The proposed system integrates machine learning and computer vision to provide traffic condition prediction and automated vehicle detection within a single web application.

The system uses a Random Forest classifier trained on historical traffic data to classify traffic conditions into four categories: Low, Normal, High, and Heavy. The prediction is based on time-related and vehicle-count features.

For automated vehicle monitoring, YOLOv8 is used to detect and count cars, bikes, buses, and trucks from uploaded images and live webcam frames. The detected vehicle counts can be transferred directly to the prediction module, reducing the need for manual data entry.

A Flask-based web application integrates both modules and provides a unified interface for traffic prediction, vehicle detection, and result monitoring.

V. MONITORING AND ANALYSIS METHODOLOGY:

A. Data Preprocessing: The traffic dataset contains information related to date, time, vehicle counts, and traffic conditions. The preprocessing procedure includes:
1. Reading the dataset and examining the available features. 2. Converting time-related information into a suitable format. 3. Extracting hour and minute values. 4. Encoding categorical variables where required. 5. Creating additional features, including peak-hour information. 6. Preparing the final feature set for model training and testing. These steps transform the original dataset into a structured format suitable for machine learning.

B. Model Training and Evaluation: The prepared dataset is divided into training and testing subsets using an 80:20 split. The Random Forest classifier is trained using the selected input features. Model performance is evaluated using commonly used classification metrics: - Accuracy - Precision - Recall - F1-score Feature importance analysis is also used to examine the relative contribution of different input variables to the classification process.

C. Vehicle Detection Process: The computer vision workflow consists of the following stages: 1. Acquisition of an image or webcam frame. 2. Transfer of the visual input to the backend. 3. Processing of the input using YOLOv8. 4. Filtering of relevant vehicle categories. 5. Calculation of vehicle counts. 6. Generation of an annotated output image. 7. Transfer of the results to the user interface.

D. System Activity Monitoring: The application maintains a recent activity log using an in-memory queue. The log records selected system events, such as manual predictions, uploaded media processing, and live detection activities. The activity information includes the source of the operation and the associated timestamp. The frontend periodically retrieves the recent activity information for display.

VI. IMPLEMENTATION

The proposed system is implemented using Python and open-source frameworks for machine learning, computer vision, and web development. The Random Forest model is developed using Scikit-learn, while YOLOv8 and OpenCV are used for vehicle detection and visual processing.

The Flask framework serves as the connection between the frontend and backend components. It receives input from the user, processes traffic prediction requests, and manages the vehicle detection operations. The backend supports manual traffic input, uploaded visual data, and webcam-based monitoring.

The frontend is developed using HTML, Bootstrap, and JavaScript to provide a simple and responsive user interface. Users can enter traffic-related information manually or obtain vehicle counts through the YOLOv8 detection module.

For live monitoring, webcam frames are captured at selected intervals and sent to the backend for processing. The detection results, including vehicle counts and bounding boxes, are returned to the interface. The system also records recent activities to provide basic monitoring of user interactions and system operations.

VII. TESTING AND EVALUATION

The proposed system was tested in a local development environment to verify the functionality and integration of its major components. Testing focused on the correct operation of the machine learning model, vehicle detection module, web interface, and communication between the frontend and backend.

The Random Forest model was tested to ensure that it successfully generated traffic condition predictions based on the provided input features. The YOLOv8 module was evaluated by processing visual traffic inputs and

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

verifying its ability to detect and count cars, bikes, buses, and trucks.

Integration testing was performed to confirm that vehicle counts obtained through the detection module could be used within the traffic prediction workflow. The webcam functionality was also tested to verify that frames could be captured, processed, and displayed without interrupting the user interface.

The testing results confirmed that the individual components operated successfully as an integrated system. Further testing with larger and more diverse real-world traffic data would be useful for evaluating the system under different environmental and traffic conditions.

VIII. RESULTS AND DISCUSSIONS

The model showed strong performance in classifying traffic conditions into the four defined categories: Low, Normal, High, and Heavy. The results indicate that the selected temporal and vehicle-count features provide useful information for distinguishing traffic conditions within the dataset.

The YOLOv8 module successfully detected and counted cars, bikes, buses, and trucks from visual inputs. Bounding boxes were displayed around detected vehicles, allowing the results to be visually verified. The integration between the detection and prediction modules demonstrated that automatically obtained vehicle counts could support the traffic classification process.

The results should be interpreted based on the dataset and testing environment used in this study. The high prediction accuracy reflects the characteristics of the selected dataset, and additional evaluation using data from different locations and traffic conditions would be required to determine the generalizability of the model.

IX. CONCLUSION AND FUTURE WORK

The proposed system demonstrates the integration of machine learning and computer vision for traffic monitoring and traffic condition prediction. A Random Forest classifier is used to classify traffic conditions, while YOLOv8 provides automated vehicle detection and counting from visual inputs.

The Flask-based web application combines these components into a single platform that supports manual prediction, vehicle detection, and webcam-based monitoring. The integration of automated vehicle counts with the prediction module reduces dependence on manual data entry and provides a practical workflow for traffic analysis.

The experimental results demonstrate the successful implementation of the system using the selected dataset and testing environment. However, further evaluation using diverse real-world traffic data is required to examine the system's performance under different conditions.

In future work, the system can be extended by using larger datasets, continuous video processing, persistent database storage, and vehicle tracking techniques. Additional features, such as weather conditions and road information, could also be incorporated to improve the overall traffic analysis process.

X. REFERENCES

[1] “Vehicle Turn Pattern Counting and Short-Term Forecasting Using Deep Learning for Urban Traffic Management System,” *IEEE Access*, vol. 13, pp. 8585–8593, 2025, doi: 10.1109/ACCESS.2025.3526880.

[2] “Vehicle Detection, Counting and Classification Using YOLOv8 Deep Learning Model and OpenCV for Urban Traffic Management System,” in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, 2024, doi: 10.1109/ICCES63552.2024.10860266.

[3] “Real-Time Traffic Congestion Detection Using ResNet50 with Multi-Dimensional Feature Fusion Residual Block in YOLOv8,” in *2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, 2025, doi: 10.1109/IACIS65746.2025.11211142.

[4] “A Vision-Based Adaptive Traffic Signal Control Framework Using YOLOv8 Detection and Random Forest Regression,” IEEE Conference Publication, 2025, doi: 10.1109/ICITISE.2025.11504606.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

[5] "Optimizing Urban Mobility: Advancing Traffic Prediction and Transportation Analytics in Complex Networks," in *2025 12th International Conference on Computing for Sustainable Global Development (INDIACom)*,2025, doi:10.23919/INDIACom66777.2025.11115778.

[6] Ultralytics, "YOLO Object Detection and Segmentation Documentation." [Online]. Available: https://docs.ultralytics.com/

[7] Scikit-learn Developers, "RandomForestClassifier Documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[8] Pallets Projects, "Flask Documentation." [Online]. Available: https://flask.palletsprojects.com/

[9] Kaggle, "Datasets for Traffic Flow Prediction." [Online]. Available: https://www.kaggle.com/datasets

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.