As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 04, pages 30-35.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue04.html
DOI: https://doi.org/10.5281/zenodo.19321932

Edge Computing for Smart City Video Analytics: A Deep Learning Approach

Jabeena Shaik, Mohaseena Shaik

Department of Computer Science and Engineering, PBR Visvodaya Institute of Technology and Science, Kavali, Andhra Pradesh, India.

To Cite this Article
Jabeena Shaik & Mohaseena Shaik (2026). Edge Computing for Smart City Video Analytics: A Deep Learning Approach. International Journal for Modern Trends in Science and Technology, 12(04), 30-35. https://doi.org/10.5281/zenodo.19321932

Article Info
Received: 28 February 2026; Revised: 18 March 2026; Accepted: 22 March 2026.

KEYWORDS

Edge AI, Smart Cities, Video Analytics, Real-time Object Detection, YOLO, OpenCV, Deep Learning, Convolutional Neural Networks, Traffic Monitoring, Crowd Management, Security Surveillance, Privacy Preservation, Low Latency, Bandwidth Optimization, IoT Integration, Intelligent Transportation Systems

ABSTRACT

Edge AI-enabled video analytics has emerged as a key technology for smart city applications, enabling real-time processing of visual data at the edge of the network. Traditional cloud-based video surveillance systems suffer from high latency, bandwidth consumption, and privacy concerns. To address these challenges, this paper proposes an efficient real-time object detection system using deep learning techniques. The proposed model utilizes the YOLO (You Only Look Once) algorithm integrated with OpenCV for detecting and classifying objects such as pedestrians, vehicles, and public transport in live video streams. The system processes data locally on edge devices, significantly reducing response time and minimizing dependency on cloud infrastructure. This enhances data privacy and ensures faster decision-making in critical scenarios such as traffic monitoring, crowd management, and security surveillance. Experimental results demonstrate that the proposed system achieves high detection accuracy with real-time performance, making it suitable for smart city environments.

Overall, the developed Edge AI-based video analytics system provides an efficient, scalable, and cost-effective solution for modern urban surveillance and intelligent transportation systems.

1.INTRODUCTION

The rapid growth of urbanization has led to the development of smart cities, where intelligent systems are required to manage infrastructure, traffic, and public safety efficiently. One of the key components of smart city systems is video surveillance, which generates massive amounts of visual data from cameras deployed across different locations. Traditional surveillance systems rely on cloud-based processing, which introduces challenges such as high latency, increased bandwidth consumption, and potential privacy risks.

Edge Artificial Intelligence (Edge AI) has emerged as a promising solution to overcome these limitations by enabling data processing closer to the source, i.e., at the edge devices. This approach reduces the dependency on centralized cloud systems and allows faster decision-making with minimal delay. Real- time video analytics using Edge AI plays a crucial role in applications such as traffic monitoring, crowd analysis, anomaly detection, and public safety management.

Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved the performance of object detection systems. Among these, the YOLO (You Only Look Once) algorithm has gained popularity due to its high speed and accuracy in real-time detection tasks. It processes images in a single pass, making it suitable for edge-based applications with limited computational resources.

In this paper, an Edge AI-enabled video analytics system is developed using YOLO and OpenCV to detect objects such as pedestrians and vehicles in real-time video streams. The system aims to provide an efficient, low-latency, and privacy- preserving solution for smart city surveillance and traffic management applications.

2. RELATED WORK

Earlier research in video surveillance and object detection primarily relied on traditional image processing techniques and machine learning algorithms such as Support Vector Machines (SVM), Decision Trees, and Random Forests. These approaches required manual feature extraction and were limited in terms of accuracy and scalability, especially when dealing with complex real- world scenarios in smart cities.

With the advancement of deep learning, Convolutional Neural Networks (CNNs) have become the dominant approach for image and video analysis tasks. Models such as R-CNN, Fast R-CNN, and Faster R-CNN significantly improved object detection accuracy by automatically learning hierarchical features from images. However, these methods often suffer from high computational complexity and slower processing speeds, making them less suitable for real-time applications.

To address these limitations, single-stage detectors like YOLO (You Only Look Once) and SSD (Single Shot Detector) were introduced. YOLO, in particular, gained widespread attention due to its ability to perform object detection in real time with high accuracy. It divides the image into grids and predicts bounding boxes and class probabilities in a single forward pass, making it highly efficient for edge-based systems.

Recent studies have also focused on integrating Edge AI with video analytics to reduce latency and bandwidth usage. Edge- based systems process data locally on devices such as embedded systems or IoT nodes, minimizing dependency on cloud infrastructure. This approach enhances privacy and enables faster response in critical applications like traffic monitoring, crowd control, and security surveillance.

The combination of deep learning-based object detection and edge computing provides a robust and efficient solution for smart city applications, motivating the development of the proposed system.

3. PROBLEM STATEMENT

Existing video surveillance and analytics systems in smart cities predominantly rely on cloud-based architectures for processing and storage. Although these systems are capable of handling large-scale data, they suffer from several critical limitations. High latency in data transmission and processing delays can hinder real-time decision-making, which is essential for applications such as traffic management and public safety. Additionally, continuous transmission of video data to the cloud results in excessive bandwidth consumption and increased operational costs.

Privacy and security concerns also arise due to centralized data storage.

Moreover, traditional object detection approaches based on handcrafted features and conventional machine learning techniques often fail to achieve high accuracy in complex and dynamic urban environments. These methods are not robust enough to handle variations in lighting conditions, occlusions, and multiple object scenarios commonly observed in real-world settings.

Objectives of this work:

- To develop a real-time object detection system using deep learning techniques
- To reduce latency by processing data at the edge instead of the cloud
- To minimize bandwidth usage and enhance data privacy

• To improve detection accuracy for smart city surveillance applications
• To provide an efficient and scalable solution for real-time video analytics

4. PROPOSED METHODOLOGY

The proposed system is designed to perform real-time video analytics using Edge AI by integrating deep learning-based object detection with efficient local processing. The overall architecture consists of four major stages:

1. Video Acquisition
2. Image Preprocessing
3. Object Detection
4. Output Processing

4.1 Video-Acquisition:

The system captures live video streams using a webcam or surveillance camera. The input video is divided into individual frames, which are processed sequentially for real-time analysis.

4.2 Image Preprocessing:

Each frame is resized to a fixed resolution and normalized to improve detection performance and reduce computational complexity. The preprocessing step ensures consistency in input data and enhances the efficiency of the deep learning model.

Let the input frame be represented as I, then normalized image is given by:

$Inorm = (I - \mu)/\sigma$

where μ represents the mean intensity and σrepresents the standard deviation.

4.3 Object Detection using YOLO:

The pre-processed frames are passed through the YOLO (You Only Look Once) model, which performs object detection in a single forward pass. YOLO divides the image into a grid and predicts bounding boxes along with class probabilities for each region.

The model identifies objects such as pedestrians, cars, buses, and trucks, which are relevant for smart city applications. YOLO is chosen due to its high speed and accuracy, making it suitable for real-time edge-based systems.

4.4 Non-Maximum Suppression (NMS):

To eliminate redundant overlapping bounding boxes, Non-Maximum Suppression is applied. This step selects the most confident detection while suppressing weaker overlapping detections, ensuring accurate object localization.

4.5 Output Display

The final processed frame is displayed with bounding boxes and labels around detected objects. The system also stores selected frames for analysis and reporting purposes.

5. IMPLEMENTATION DETAILS

The proposed Edge AI-based video analytics system is implemented using a combination of deep learning and computer vision techniques. The implementation focuses on achieving real-time object detection with minimal computational resources.

Dataset:

The system utilizes real-time video input captured from a webcam or surveillance camera. Pre-trained datasets such as the COCO (Common Objects in Context) dataset are used for training the YOLO model, which contains multiple object classes relevant to smart city applications.

Programming Language:

Python is used as the primary programming language due to its simplicity and extensive support for machine learning and computer vision libraries.

Libraries and Tools:

• OpenCV – for video capture and image processing
• NumPy – for numerical computations
• YOLOv3 – for object detection
• Pre-trained weights and configuration files (yolov3.weights, yolov3.cfg, coco.names)

Hardware Requirements:

The system is designed to run on a standard CPU-based system without requiring high- end GPU support, making it suitable for edge devices with limited resources.

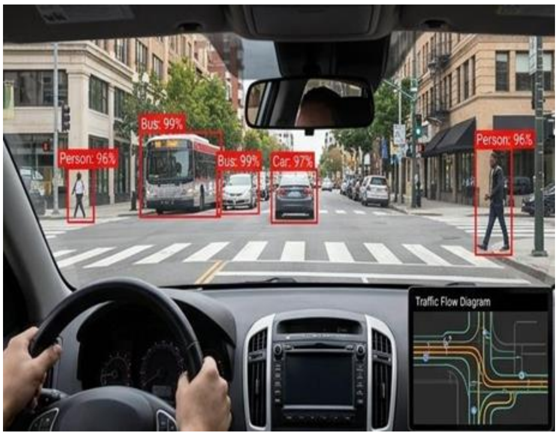
Implementation Pipeline:

Video Input → Frame Extraction → Image Preprocessing → YOLO Detection → Non- Max Suppression → Output Display

Each frame from the video stream is processed independently. The YOLO model detects objects and generates bounding boxes with confidence scores. Non-Maximum Suppression is applied to remove duplicate detections, and the final output is displayed in real time with labeled objects.

The system also stores selected output frames for evaluation and reporting purposes, demonstrating its effectiveness in smart city scenarios such as traffic monitoring and surveillance.

OUTPUT:



6. PERFORMANCE EVALUATION

Metrics used:

The performance of the proposed Edge AI- based video analytics system is evaluated using standard evaluation metrics commonly used in object detection and classification tasks. These metrics help in analyzing the accuracy and reliability of the system in detecting objects in real-time video streams.

The following performance metrics are used:

Accuracy:

Accuracy measures the overall correctness of the system in detecting objects.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision:

precision indicates how many of the detected objects are actually correct.

Precision = TP / (TP + FP)

Recall:

Recall measures the ability of the system to detect all relevant objects.

Recall = TP / (TP + FN)

F1-Score:

F1-Score is the harmonic mean of precision and recall.

F1 Score = 2 × (Precision × Recall) / (Precision + Recall)

where TP (True Positive) represents correctly detected objects, TN (True Negative) represents correctly rejected objects, FP (False Positive) represents incorrect detections, and FN (False Negative) represents missed detections.

In addition to these metrics, processing time per frame and frames per second (FPS) are also considered to evaluate the real-time performance of the system. A higher FPS indicates better efficiency and suitability for real-time applications.

The confusion matrix is used to analyze the classification performance of the system by comparing actual and predicted outputs. The proposed system demonstrates high accuracy with efficient real-time processing, making it suitable for smart city surveillance and traffic monitoring applications.

7. RESULTS AND DISCUSSION

The proposed Edge AI-based video analytics system was tested using real-time video input to evaluate its performance in detecting objects such as pedestrians and vehicles. The system successfully identified multiple objects in different environmental conditions, including varying lighting and background complexity. The YOLO-based detection model demonstrated the capability to process frames efficiently and provide accurate bounding boxes with class labels.

The experimental results indicate that the system achieves high detection accuracy with minimal delay, making it suitable for real- time applications. The integration of edge processing significantly reduced latency compared to traditional cloud-based systems, as data is processed locally without the need for continuous data transmission. This also resulted in lower bandwidth consumption and improved data privacy.

The system was able to handle multiple object detection simultaneously, maintaining consistent performance even in crowded scenes. The use of Non-Maximum Suppression (NMS) helped eliminate redundant detections, thereby improving the overall precision of the system.

Advantages of the proposed system:

- Real-time object detection with low latency
- Reduced bandwidth usage due to edge processing
- Improved data privacy and security
- High accuracy in detecting multiple objects
- Efficient performance on standard hardware systems

Overall, the results demonstrate that the proposed system is effective and reliable for smart city applications such as traffic monitoring, surveillance, and crowd management.

8. FUTURE WORK

The proposed Edge AI-based video analytics system can be further enhanced to support more advanced smart city applications.

Future improvements may focus on increasing the system's accuracy, scalability, and functionality by integrating additional technologies and features.

Some possible directions for future work include:

- Integration of accident detection and emergency alert systems for real-time incident response
- Implementation of vehicle counting and traffic flow analysis for intelligent transportation systems
- Deployment on embedded edge devices such as Raspberry Pi or NVIDIA Jetson for real-world applications
- Incorporation of advanced deep learning models to improve detection accuracy under challenging conditions
- Development of a web-based or mobile application interface for remote monitoring and control
- Integration with IoT systems for automated decision-making and smart infrastructure management
- Enhancement of the system to support facial recognition and behavior analysis for improved security

These improvements will make the system more robust, scalable, and suitable for large- scale deployment in modern smart city environments.

9. CONCLUSION

This Paper presents an efficient Edge AI- enabled video analytics system for smart city applications using deep learning techniques. The proposed system utilizes the YOLO algorithm integrated with OpenCV to perform real-time object detection on video streams. By processing data locally at the edge, the system significantly reduces latency, bandwidth usage, and dependency on cloud infrastructure, while also enhancing data privacy.

The experimental results demonstrate that the system is capable of accurately detecting multiple objects such as pedestrians and vehicles in real-time scenarios. The use of Non-Maximum Suppression improves detection precision by eliminating redundant bounding boxes. Additionally, the system performs effectively on standard hardware, making it suitable for deployment on edge devices.

Overall, the developed system provides a reliable, scalable, and cost-effective solution for smart city surveillance, traffic monitoring, and public safety applications. It highlights the potential of Edge AI in enabling intelligent and real-time decision- making systems for modern urban environments.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[3] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), 2012.

[5] M. Satyanarayanan, "The Emergence of Edge Computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[7] T. Lin et al., "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision (ECCV), 2014.

[8] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[10] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of- Freebies Sets New State-of-the-Art for Real- Time Object Detectors," arXiv preprint arXiv:2207.02696, 2022.

[11] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[13] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.