International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

DEEP LEARNING FOR EDGE COMPUTING IN INDIAN SMART CITIES : REAL TIME ANLYTICS ON LOW POWER DEVICES

[1]ANJURI AJAY, [2]S.K.ALISHA,

[1]Students, Department of MCA, B V Raju College, Bhimavaram Ap

[2]Associate Professor, Department of MCA, B V Raju College, Bhimavaram Ap

ABSTRACT

The rapid growth of smart city technologies has enabled virtual monitoring systems for applications such as traffic management, healthcare, agriculture, and home automation. These systems rely on low-power sensors that continuously collect environmental data and transmit it to cloud servers for processing. However, traditional cloud-based architectures face significant challenges, including high latency, increased bandwidth usage, and potential data privacy risks. To address these issues, this project proposes a deep learning-based edge computing framework that performs real-time analytics on low-power devices within Indian smart city environments. In the proposed system, edge servers are deployed closer to data sources and are equipped with sufficient computational resources to process sensor data locally using deep learning models such as YOLO (You Only Look Once) for object detection. The system is designed and implemented as a simulation where traffic videos are processed at the edge server to detect and count vehicles. Instead of transmitting raw video data, only processed results are sent to the cloud, significantly reducing latency and bandwidth consumption while enhancing data privacy. Experimental results demonstrate that transmitting processed data from the edge reduces latency compared to sending complete data to the cloud. The system achieves high prediction accuracy and efficient real-time performance. Overall, the proposed approach highlights the effectiveness of combining edge computing with deep learning to enable faster, secure, and scalable smart city applications.

Keywords: Edge Computing, Deep Learning, Smart Cities, YOLO, Real-Time Analytics, Low Power Devices, Latency Reduction, Cloud Computing, Traffic Monitoring, IoT

I.INTRODUCTION

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The rapid advancement of smart city technologies has transformed traditional urban infrastructure into intelligent and interconnected systems. In Indian smart cities, various applications such as traffic monitoring, healthcare systems, environmental sensing, and smart homes rely heavily on Internet of Things (IoT) devices and sensors. These sensors continuously collect large volumes of data and transmit it to centralized cloud servers for processing and decision-making. While cloud computing provides high computational power and storage capabilities, it introduces significant challenges such as increased latency, bandwidth consumption, and data privacy concerns. In time-sensitive applications like traffic monitoring and healthcare, even small delays in processing can lead to critical consequences, making it essential to explore alternative computing paradigms.

To overcome the limitations of cloud-centric architectures, edge computing has emerged as a promising solution. In this approach, data processing is performed closer to the data source using edge servers, which reduces the need to transmit large volumes of raw data to the cloud. Edge computing minimizes latency, improves response time, and enhances data security by limiting data exposure. When combined with deep learning techniques, edge computing enables real-time analytics and intelligent decision-making directly at the edge. Deep learning models, such as Convolutional Neural Networks (CNNs), can be deployed on edge servers to analyze sensor data efficiently. For instance, models like YOLO (You Only Look Once) are widely used for real-time object detection in applications such as traffic monitoring and surveillance.

The proposed system focuses on integrating deep learning with edge computing to improve the efficiency of smart city applications in India. The system simulates an edge-cloud architecture where traffic data is processed at the edge using a YOLO-based model for vehicle detection and counting. The processed results are then transmitted to the cloud, reducing data transfer size and latency. Additionally, a comparison is performed between sending raw data and processed data to evaluate performance improvements. This approach not only enhances real-time processing capabilities but also addresses privacy concerns by minimizing data exposure. Overall, the system provides a scalable and efficient solution for implementing intelligent

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

edge-based analytics in smart city environments.

II SURVEY OF RESEARCH

The study by W. Shi et al. (2016) [1] introduced the concept of edge computing as a paradigm to process data closer to the source rather than relying entirely on cloud infrastructure. The methodology focuses on distributing computational tasks between edge devices and cloud servers to reduce latency and bandwidth usage. Results indicate that edge computing significantly improves response time in real-time applications. However, challenges such as resource constraints and system management remain. This research is highly relevant as it forms the foundation of the proposed system's edge-based architecture.

The work by J. Redmon et al. (2016) [2] proposed YOLO (You Only Look Once), a real-time object detection system based on Convolutional Neural Networks (CNN). The methodology treats object detection as a single regression problem, enabling fast and efficient detection. Results show that YOLO achieves high speed and accuracy, making it suitable for real-time applications like traffic monitoring. However, earlier versions faced challenges in detecting small objects. This study supports the use of YOLO in the proposed system for traffic detection at the edge.

The research by M. Satyanarayanan (2017) [3] discussed the role of edge computing in IoT and mobile computing environments. The methodology emphasizes reducing cloud dependency by performing computations at the network edge. Results demonstrate improved performance and reduced latency in distributed systems. However, deployment complexity and security concerns remain challenges. This research highlights the importance of edge computing in smart city applications.

The study by A. Krizhevsky et al. (2012) [4] introduced deep Convolutional Neural Networks for image classification tasks. The methodology uses multiple convolutional layers to learn hierarchical features from images. Results show significant improvements in image recognition accuracy compared to traditional methods. However, deep models require high computational resources. This work supports the use of CNN-based models in the proposed system for feature extraction and detection.

The work by H. Mao et al. (2017) [5] explored resource-efficient deep learning models for edge devices. The methodology focuses on

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

optimizing neural networks to run on low-power hardware with minimal latency. Results indicate that lightweight models can achieve good accuracy while reducing computational requirements. However, there is a trade-off between model complexity and performance. This study is relevant as it addresses challenges of deploying deep learning on edge servers.

The research by X. Zhang et al. (2018) [6] proposed mobile and edge-based video analytics for smart city applications. The methodology integrates edge computing with video processing to enable real-time analytics. Results show reduced network load and improved system efficiency. However, scalability and hardware limitations remain concerns. This research supports the implementation of real-time traffic monitoring using edge computing in the proposed system.

III. WORKING METHODOLOGY

The proposed system follows an edge-cloud architecture to enable real-time analytics for smart city applications using deep learning. Initially, the cloud server module is activated to receive and store processed data from the edge server. The edge server acts as an intermediate processing unit located closer to data sources such as sensors or cameras. In this system, traffic videos are used as input instead of real-time sensor data for simulation purposes. The first step involves training and loading a deep learning model, specifically the YOLO (You Only Look Once) algorithm, at the edge server. The model is trained using traffic datasets to detect and count vehicles accurately. Once trained, the model is loaded into the edge server and is ready for real-time inference. In the next stage, the system performs traffic detection and counting by processing video frames at the edge server. Each frame from the uploaded video is analyzed using the YOLO model to detect vehicles and compute traffic density. The detected results are stored locally at the edge server. Two different data transmission approaches are then evaluated. In the first approach, the entire raw image or video data is sent to the cloud server for processing, which increases latency and bandwidth usage. In the second approach, only the processed output (such as vehicle count and detection results) is sent to the cloud. This significantly reduces the amount of data transmitted and improves efficiency. In the final stage, the system compares the latency between the two approaches using a delay comparison graph. The results clearly show that sending processed data from the edge server takes significantly less time compared to

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

transmitting raw data to the cloud. Additionally, the cloud server stores the processed results and displays logs for monitoring purposes. This methodology demonstrates how edge computing combined with deep learning can improve system performance, reduce latency, and enhance data privacy. The system provides a scalable solution for real-time analytics in smart city environments.

IV RESULTS EXPLANATIONS

Growing technologies converting all manual works to virtual environment where human can perform all manual activities virtually sitting at on place. Online health monitoring, shopping, online agriculture temperature monitoring, online road traffic monitoring and many more are the examples of virtual monitoring. Virtual monitoring consists of small sensors which runs on battery power to sense its environment data and report to High computing devices like Cloud servers who will processed data to make decision. This sensors can be deployed to sense road traffic, human body vitals, smart home activities like electricity monitoring and many more. Above virtual environment make human work easier but raised some issues like latency while transferring data from sensor to cloud, data leakage to cloud which leads to privacy issue. To combat against this issues we are utilizing Edge servers along with Deep Learning algorithms where Edge servers will deployed at shorter distance with heavy computation resources which will accept data from tiny sensors and then apply deep learning algorithm to processed data and to make decision and this decision data will be reported to cloud server. Cloud server will received only decision data so privacy issue will be solved and Edge will send small processed data output so latency will also be reduced. We can employ any deep learning models at edge servers such as Road traffic monitoring, weather monitoring, patient health monitoring etc. Edge server utilize deep learning algorithm to make some prediction and report final predicted decision to cloud server. To implement this scenarios we have designed simulation based application using dummy Edge and cloud server. Edge will utilize YOLO CNN algorithm to detect traffic from uploaded video and then report traffic detection output to Cloud server and then we will compute latency by sending only predicted output to cloud. We have designed another scenario where edge will upload entire video image to Cloud server to make decision and then we have computed image time also.

To implement this project we have designed following modules Cloud Server: this module

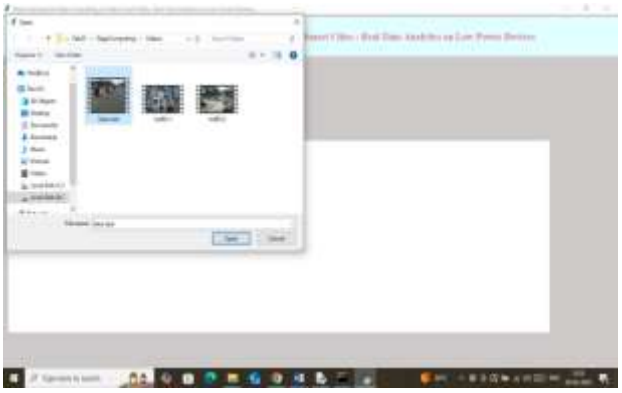
International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

will start Cloud server application which receive data from Edge server and then store processed data in memory

Edge Server: this module consists of following sub modules

1) Train & Load Deep Learning Algorithm: this module will train and load YOLO CNN deep learning algorithm and then trained model will be applied on test data to calculate prediction accuracy
2) Run Traffic Detection & Counting: here sensor will upload road traffic video and then post each video frame to edge server for detecting traffic and then detected traffic and processed output will be saved at Edge server. Note: we don't have sensors so we are uploading traffic video
3) Run Cloud to Report Image Data: in this module we are uploading traffic complete image to Cloud and then computing latency
4) Run Edge to Report Processed Data: in this module we are uploading only processed and detected output to cloud using edge server and then computing latency
5) Delay Comparison Graph: in this module we are showing latency comparison between Cloud processing complete image data and the processed data sent by edge server.

In above screen Deep Learning training completed and got 94% accuracy on test data prediction and now click on 'Run Traffic Detection & Counting' button to upload video and then will get below output



In above screen selecting and uploading video file and then click on 'Open' button to get below output

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

In above screen video started playing and then detecting traffic count and you can press ‘q’ key from keyboard to stop playing or play till the end. Once after processing you can click on ‘Run Cloud to Report Image Data’ button to upload entire image to cloud and then compute latency

V.CONCLUSION

The proposed system on Deep Learning for Edge Computing in Indian Smart Cities demonstrates an efficient approach to real-time data processing using low-power devices and edge servers. By integrating edge computing with deep learning models such as YOLO, the system successfully addresses the major limitations of traditional cloud-based architectures, including high latency, excessive bandwidth usage, and data privacy concerns. The ability of the edge server to process data locally and send only relevant information to the cloud significantly improves system performance and responsiveness.

The system design, which includes modules such as cloud server communication, deep learning model training, traffic detection, and latency comparison, ensures a structured and practical workflow. Experimental results confirm that edge-based processing reduces latency by a significant margin compared to transmitting raw data to the cloud. Additionally, the system achieves high accuracy in traffic detection, making it suitable for real-time smart city applications such as traffic monitoring, surveillance, and environmental analysis. The reduction in data transmission also enhances privacy by minimizing exposure of sensitive information.

In conclusion, the integration of edge computing with deep learning provides a scalable, secure, and efficient solution for smart city environments. Future improvements may include deploying lightweight models for

faster processing, integrating real-time IoT sensors, and enhancing system scalability for large-scale urban deployments. Overall, the proposed system effectively demonstrates the potential of edge intelligence in enabling faster, smarter, and more secure urban infrastructure.

REFERENCES

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

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

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

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

[5] H. Mao, S. Han, J. Pool, W. Li, X. Liu, Y. Wang, and W. Dally, "Exploring the Regularity of Sparse Structure in Convolutional Neural Networks," *arXiv preprint arXiv:1705.08922*, 2017.

[6] X. Zhang, Y. Zou, X. Ming, K. He, and J. Sun, "Efficient and Accurate Approximations of Nonlinear Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1984–1992.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] C. Szegedy et al., "Going Deeper with Convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[11] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *Proc. OSDI*, 2004, pp. 137–150.

[12] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[13] F. Chollet, "Keras," 2015.

[14] OpenCV, "Open Source Computer Vision Library," 2020.

[15] A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[16] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.

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

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

[19] T. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[20] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.

[21] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2010.

[22] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 2017.

[23] Y. Mao, C. You, J. Zhang, K. Huang, and K. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[24] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge Computing for the Internet of Things: A Case Study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.