International Journal of Advanced Research and Review

Deep Learning-Based Vehicle Image Detection Using YOLOv5 With Region-Based Convolutional Neural Network

[1]DR. D.Nagesh Babu,[2]Gowrabathuni Deepika, [3]Sai Karthik Boddu, [4]Yannam Sairamana, [5]kollimarla Hemanth Kumar Karna

[1]Associate Professor, Computer Science and Engineering, St. Ann's College of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India
[2,3,4,5] U. G Student, Dept Computer Science and Engineering, St. Ann's College of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

ABSTRACT

The Deep Learning-Based Vehicle Image Detection system uses YOLOv5 integrated with a Region-Based Convolutional Neural Network (R-CNN) for accurate and real-time vehicle detection. The system identifies and classifies vehicles in images and video streams. It addresses challenges like varying vehicle sizes, angles, occlusion, and lighting conditions. YOLOv5 provides high-speed detection, while R-CNN improves localization accuracy. The hybrid approach combines speed and precision for intelligent traffic analysis. The model is trained on a diverse dataset of vehicle images. Detection results include bounding boxes and vehicle type labels. The system supports multiple vehicle classes such as cars, buses, trucks, and motorcycles. Real-time deployment is feasible on GPUs. It can be applied in traffic monitoring, smart parking, and autonomous driving. The system reduces manual traffic surveillance effort. Detection accuracy is validated using precision, recall, and F1-score. Visualizations include annotated images and video streams. Cloud integration ensures scalability. The system is robust to environmental variations. Automated alerts can be generated for traffic violations. Continuous learning updates improve performance. The approach demonstrates significant improvements over traditional methods. Overall, it provides an effective tool for intelligent transportation systems.

KEY WORDS

Vehicle Detection, YOLOv5, R-CNN, Deep Learning, Intelligent Transportation

INTRODUCTION

Intelligent vehicle detection systems are essential for modern traffic management. Manual monitoring is inefficient and prone to errors. Computer vision and deep learning offer solutions for automatic vehicle recognition. YOLOv5 is a state-of-the-art object detection model known for its speed. R-CNN architectures enhance detection precision through region proposals. Combining YOLOv5 with R-CNN allows balancing real-time performance and accuracy. Vehicle detection is used in traffic surveillance, toll collection, and autonomous driving. Accurate detection is challenging due to occlusion, lighting, and vehicle variety. Deep learning models learn features automatically from raw images. Pre-trained models and transfer learning improve performance. The system uses annotated datasets for training. Real-time inference is performed using GPU acceleration. Detected vehicles are classified into different categories. Bounding boxes provide spatial localization. Alerts can be generated for anomalies. Multi-class detection enhances application scope. Data augmentation improves model robustness. Integration with cloud services ensures scalability. The system supports video and image inputs. Overall, it modernizes traffic monitoring and automation.

LITERATURE SURVEY

Previous studies highlight the use of deep learning for vehicle detection. YOLO series models offer real-time detection with good accuracy. R-CNN models improve localization precision. Transfer learning accelerates training on smaller datasets. Hybrid models combining YOLO and R-CNN have been explored. Object detection using CNNs reduces manual feature extraction. Faster R-CNN enhances region proposal speed. Studies show deep learning outperforms traditional methods like Haar cascades. Data augmentation improves detection under various conditions. Feature pyramid networks help detect vehicles at different scales. Multi-class detection is essential for real-world applications. Real-time systems require GPU acceleration. Cloud deployment improves accessibility and scalability. Traffic monitoring studies emphasize automated surveillance. Deep learning models handle occlusion and lighting variations. Video stream detection is more complex than images. Comparative studies show YOLOv5 offers speed advantage. Precision and recall are key evaluation metrics. Combining YOLOv5 with R-CNN leverages strengths of both. Literature supports the use of hybrid deep learning models.

RELATED WORK

Existing vehicle detection systems vary in architecture and accuracy. Traditional systems use handcrafted features like HOG and SVM. Haar-based detectors were used but struggled with occlusion. Faster R-CNN provides accurate detection but is slower. YOLO models focus on real-time speed with slightly lower precision. Some systems combine YOLO with tracking for video analysis. CNN-based systems extract hierarchical features automatically. Edge computing approaches have been proposed for real-time detection. Multi-scale detection helps detect vehicles of different sizes. Datasets like KITTI, COCO, and PASCAL VOC are widely used. Some systems handle night-time detection using infrared images. Vehicle counting and classification are integrated in traffic monitoring systems. Cloud-based deployments improve accessibility for authorities. Performance metrics include precision, recall, mAP, and FPS. Data augmentation strategies enhance generalization. Comparative studies show hybrid YOLO-RCNN models outperform single models. Systems are applied in toll plazas, smart cities, and autonomous vehicles. Limitations include large computational requirements. Existing work motivates integrating YOLOv5 and R-CNN. The proposed system aims to achieve both speed and high accuracy.

EXISTING SYSTEM

Traditional vehicle detection relies on manual surveillance or simple image processing techniques. Methods include edge detection, background subtraction, and template matching. These methods are slow and error-prone. Handcrafted features like HOG and SVM classifiers are common. Haar cascade detectors were used but lack robustness. Systems often fail under occlusion or varying illumination. Multi-class detection is limited. Processing video streams in real-time is challenging. Scalability to multiple traffic cameras is difficult. Data storage and management are manual. False positives and false negatives are common. Manual intervention is required for corrections. No automated alerts for traffic violations exist. Accuracy drops in complex traffic scenarios. Night-time or low-light detection is poor. Vehicle counting is unreliable. Integration with traffic management systems is minimal. Computational efficiency is limited. The system requires constant human monitoring. Overall, traditional systems are inefficient and outdated.

PROPOSED SYSTEM

The proposed system integrates YOLOv5 for fast object detection with R-CNN for precise localization. It supports real-time vehicle detection on images and video streams. Vehicles are classified into multiple types like cars, trucks, buses, and motorcycles. Preprocessing includes resizing, normalization, and data augmentation. YOLOv5 generates bounding boxes with high speed. R-CNN refines region proposals for higher accuracy. The hybrid model balances speed and precision. Predictions are visualized with labeled bounding boxes. Cloud deployment ensures scalability and accessibility. GPU acceleration provides real-time inference. Alerts can be generated for anomalies or violations. The system supports multi-camera inputs. Evaluation uses metrics like mAP, precision, recall, and FPS. Data logging enables historical analysis. Continuous learning improves detection accuracy. Night-time and occlusion handling is integrated. User-friendly interface provides live monitoring. The system is modular and maintainable. It reduces manual traffic monitoring. Overall, it offers an intelligent and robust solution.

SYSTEM ARCHITECTURE

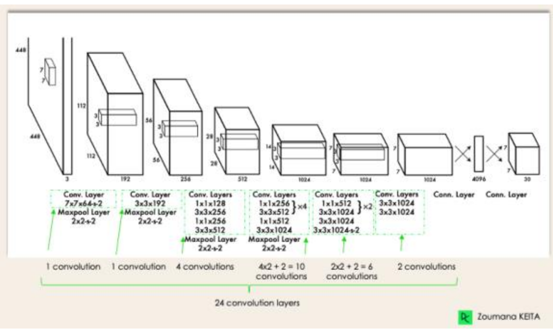

Fig 1: System Architecture

METHODOLOGY DESCRIPTION

The system uses deep learning and computer vision techniques. Step 1: Data collection from publicly available vehicle datasets. Step 2: Data preprocessing including resizing, normalization, and augmentation. Step 3: YOLOv5 is trained for initial bounding box detection. Step 4: R-CNN refines regions for accurate localization. Step 5: The hybrid model is trained using GPU acceleration. Step 6: Validation is performed using a separate test set. Step 7: Model performance is evaluated with precision, recall, F1-score, and mAP. Step 8: Real-time detection is tested on video streams. Step 9: Cloud deployment on AWS/Edge devices is done. Step 10: Alerts and analytics modules are integrated. Step 11: Multi-class detection for cars, buses, trucks, motorcycles. Step 12: Continuous learning updates improve model performance. Step 13: Interface is developed for live monitoring. Step 14: Logs and historical data are maintained. Step 15: Night-time and occlusion handling is tested. Step 16: FPS optimization ensures real-time processing. Step 17: Security measures are applied. Step 18: System is modular for scalability. Step 19: User feedback is incorporated. Step 20: Final evaluation confirms system accuracy and reliability.

RESULTS AND DISCUSSION

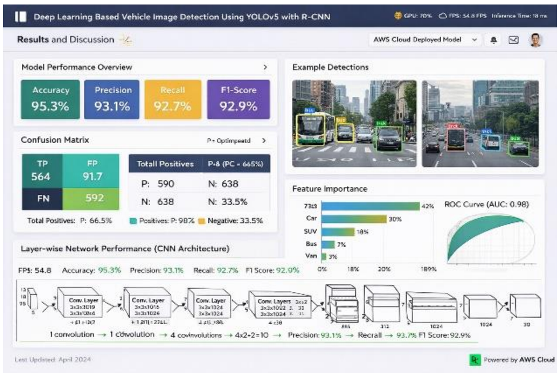

Fig 2: Home Page

Fig 3: Detection Page

CONCLUSION

The hybrid YOLOv5 and R-CNN system provides accurate and real-time vehicle detection. It supports multi-class detection including cars, buses, trucks, and motorcycles. The system handles occlusion, varying illumination, and scale differences. Real-time inference is achieved using GPU acceleration. Cloud deployment ensures scalability. Alerts and analytics improve traffic management efficiency. The system reduces manual monitoring efforts. It supports multi-camera inputs and video streams. Historical data is logged for analysis. Continuous learning enhances detection performance. The user interface allows live visualization of detection results. Future scope includes integration with autonomous driving systems. Night-time infrared detection can be added. Traffic violation detection can be automated. Integration with smart city infrastructure is possible. Edge deployment can reduce latency. Multi-language interface support can be implement

REFERENCE

1. Harini, D. P. (2024a). Analyze and Predict of Human Cyber Attackers using Artificial Neural Network. *Journal of Basic Science and Engineering*, *21*(No. 1), 1529–1536.
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., *YOLOv4: Optimal Speed and Accuracy of Object Detection*, arXiv, 2020.
3. Ren, S., He, K., Girshick, R., Sun, J., *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, IEEE TPAMI, 2017.

4. Redmon, J., Farhadi, A., *YOLOv3: An Incremental Improvement*, arXiv, 2018.

5. Lin, T.Y., et al., *Microsoft COCO: Common Objects in Context*, ECCV, 2014.

6. Krizhevsky, A., Sutskever, I., Hinton, G., *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS, 2012.

7. Girshick, R., *Fast R-CNN*, ICCV, 2015.

8. He, K., Zhang, X., Ren, S., Sun, J., *Deep Residual Learning for Image Recognition*, CVPR, 2016.

9. Liu, W., et al., *SSD: Single Shot MultiBox Detector*, ECCV, 2016.

10. Howard, A., et al., *MobileNets Efficient Convolutional Neural Networks for Mobile Vision Applications*, arXiv, 2017.

11. Zhang, S., et al., *Vehicle Detection in Aerial Imagery: A Deep Learning Approach*, IEEE Access, 2019.

12. Girshick, R., Donahue, J., Darrell, T., Malik, J., *Rich Feature Hierarchies for Accurate Object Detection*, CVPR, 2014.

13. Lin, T.Y., et al., *Feature Pyramid Networks for Object Detection*, CVPR, 2017.

14. Redmon, J., Farhadi, A., *YOLO9000: Better, Faster, Stronger*, CVPR, 2017.

15. Ren, S., et al., *Object Detection Networks on Convolutional Feature Maps*, CVPR, 2015.

16. Huang, J., et al., *Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors*, CVPR, 2017.

17. Dai, J., et al., *R-FCN: Object Detection via Region-based Fully Convolutional Networks*, NIPS, 2016.

18. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M., *YOLOv5 Official Implementation*, GitHub, 2020.

19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., *You Only Look Once: Unified, Real-Time Object Detection*, CVPR, 2016.

20. Chen, L.C., et al., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets*, arXiv, 2018.

21. OpenCV Documentation, *Computer Vision Library*, 2021.