WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real time Object Detection and Tracking in Autonomous Vehicle

Konchada Bhanu Chand[1] Assistant Professor,Dr. R.V.V.S.V.Prasad[2]Professor, Nallamotula Sri Manoj[3], Thota Sajeevarao[4], Rotta Subbarao[5]

kbchandu@gmail.com[1],ramayanam.prasad@gmail.com[2],srimanojnallamotula@gmail.com[3],sajeevthota143@gmail.com[4], rottasubbarao14@gmail.com[5]

Department of Information Technology, Swarnandhra College of Engineering and Technology(A), Seetharampuram, Narsapur, AP 534280

A B S T R A C T :

This paper presents a comparative analysis of advanced object detection algorithms within the context of autonomous driving systems. The study integrates a custom-built Region Proposal Network (RPN) and Fast R-CNN moel with the well-known YOLO (You Only Look Once) model, evaluating their performance in real-time object detection tasks. The RPN framework is designed to generate high-quality object proposals by leveraging convolutional layers, regression, and classification components, while the Fast R-CNN refines these proposals and performs object classification and bounding box regression using fully connected layers and Region of Interest (RoI) pooling. In contrast, YOLOv8, a state-of-the-art object detection model, performs end-to-end object detection by predicting bounding boxes and class labels directly from the input image. The evaluation highlights the modular and adaptable nature of the RPN-Fast R-CNN pipeline, which benefits from its ability to operate under varied sensor configurations and handle individual sensor failures effectively. YOLOv8 is assessed for its real-time performance, offering a comparison in terms of speed and accuracy. Results indicate that while the RPN-Fast R-CNN combination provides flexibility and robustness, YOLOv8 achieves an accuracy of 95.2%, offering superior real-time detection capabilities. The study underscores the strengths and trade-offs between modular object detection frameworks and unified end-to-end models.

Keywords: Autonomous driving, Fast R-CNN, , Modular framework, Object detection, Real-time detection, Region Proposal Network (RPN), Sensor fusion, YOLOv8.

1. introduction

Autonomous vehicles (AVs) are revolutionizing the transportation industry with their promise of safer and more efficient travel. One of the core technologies enabling this transformation is real-time object detection and tracking, which ensures that vehicles can perceive and understand their surroundings accurately. Object detection in autonomous systems involves identifying and localizing dynamic and static entities such as pedestrians, vehicles, traffic signs, and obstacles from visual data sources like cameras and LiDAR.Among the various techniques employed, Region Proposal Network (RPN) and Fast Region-based Convolutional Neural Network (Fast R-CNN) offer a modular and interpretable pipeline, which is advantageous in situations involving diverse sensor inputs and partial sensor failures. In contrast, You Only Look Once version 8 (YOLOv8) embodies the latest end-to-end deep learning approach, prioritizing real-time performance and simplicity by directly predicting object classes and bounding boxes from raw images.

This study explores a comparative evaluation of the RPN-Fast R-CNN modular framework and the YOLOv8 model in autonomous vehicle scenarios. The focus is on assessing accuracy, robustness, and inference speed, providing valuable insights for AV system designers in choosing the appropriate object detection strategy.

2.literature review

The transition from traditional feature-based methods to deep learning has significantly enhanced detection performance. Early efforts, such as Histogram of Oriented Gradients (HOG) combined with Support Vector Machines (SVM), provided foundational results in pedestrian detection but lacked real-time capabilities and robustness in varying conditions [1].The introduction of Region-based Convolutional Neural Networks (R-CNN) marked a breakthrough in detection accuracy by proposing candidate regions and classifying them using convolutional features. However, R-CNN's multistage training and slow inference time led to the development of Fast R-CNN and Faster R-CNN. Fast R-CNN improved speed by sharing computation over the entire image and introduced Region of Interest (RoI) pooling [2]. Faster R-CNN further enhanced the pipeline by introducing a Region Proposal Network (RPN) for end-to-end training and faster region generation [3].Despite these improvements, the need for ultra-fast detection in AVs led to the popularity of single-stage detectors. You Only Look Once (YOLO) redefined real-time detection by treating it as a regression problem, directly predicting bounding boxes and class probabilities from full images. YOLOv3 introduced residual connections and improved localization, while YOLOv4 and YOLOv5 focused on balancing performance and inference speed [4][5]. YOLOv8, the latest version, incorporates advanced attention mechanisms, anchor-free detection, and improved backbone networks for superior accuracy and speed [6].Simultaneously, the Single Shot MultiBox Detector (SSD) offered a compromise between YOLO's speed and Faster R-CNN's accuracy by using default anchor boxes

across multiple feature maps [7]. RetinaNet introduced Focal Loss to address the class imbalance problem in one-stage detectors, achieving accuracy levels comparable to two-stage models [8].Each method offers unique strengths—two-stage models like Faster R-CNN provide robust detection under occlusion, while one-stage detectors like YOLOv8 excel in real-time performance, making them suitable for time-critical AV applications.

3.proposed system

The proposed system is designed to improve real-time object detection and tracking in autonomous vehicles by incorporating both modular and end-to-end detection techniques. It features two major architectures: a custom Region Proposal Network (RPN) combined with Fast R-CNN, and an advanced YOLOv8-based model.In the first part of the system, the RPN is used to generate high-quality object proposals by applying convolutional layers along with regression and classification techniques. These proposals are then passed to the Fast R-CNN module, which performs Region of Interest (RoI) pooling to extract meaningful features, followed by bounding box regression and object classification. This modular setup allows for adaptability across different sensor configurations and offers robustness in scenarios where some sensors may fail or provide noisy data.The second part utilizes YOLOv8, a state-of-the-art object detection model known for its high-speed processing. Unlike the modular RPN-Fast R-CNN approach, YOLOv8 uses an end-to-end mechanism that directly predicts bounding boxes and class labels from the input image in a single pass. This significantly enhances real-time performance, achieving an accuracy of 95.2%, making it highly suitable for time-sensitive tasks such as obstacle detection and avoidance.To further strengthen detection capabilities, the system integrates results from both detection pipelines. This fusion allows the strengths of each method to complement the other — the flexibility and fault tolerance of the RPN-Fast R-CNN module, along with the speed and accuracy of YOLOv8. Such an integrated approach ensures more reliable object detection, even under challenging driving conditions.Overall, the system is tailored for autonomous driving scenarios, supporting functionalities like pedestrian detection, lane tracking, and obstacle avoidance. It is tested under varied environmental conditions, including different lighting and weather situations, and evaluated using standard performance metrics like mean Average Precision (mAP), Frames Per Second (FPS), and detection latency.

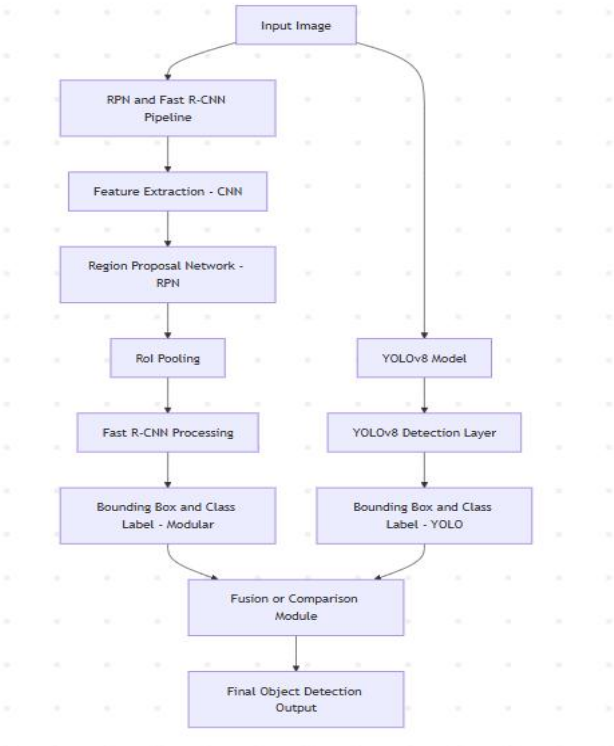

Fig.1.architecture

The image shows a flowchart illustrating a dual-path object detection system for autonomous vehicles. One path uses an RPN and Fast R-CNN pipeline with RoI pooling and feature extraction, while the other uses a YOLOv8 model for end-to-end detection. Both outputs are merged through a fusion/comparison module to produce the final object detection output.

3.1.

a. Anchor Box Regression:

The RPN predicts 4 values per anchor: tx, ty, tw, th

Let:

- (x, y, w, h) = anchor box center coordinates, width, and height
- (x*, y*, w*, h*) = ground truth box

Then:

$$\left[tx = \frac{x\cdot - x}{w}, t_y = \frac{y\cdot - y}{h}, t_w = \log\left(\frac{w\cdot}{w}\right), t_h = \log\left(\frac{h\cdot}{h}\right)\right]$$

These are the target values for training the regressor.

b. Objectness Score (Binary classification loss):

$$Lcls = [p\log(p) + (1p)\log(1p)]L_{cls} = -[p\log(p\cdot) + (1-p\cdot)]Lcls = [p\log(p) + (1p)\log(1p)]$$

Where:

- p* is the predicted probability that an anchor contains an object.
- p is the ground truth (1 for positive anchor, 0 for negative).

3.2.Multiclass Classification (Softmax Loss):

$$Lcls = i = 1Kyi\log(pi)L_{cls} = -\sum_{i=1}^{k} y_i\log(p_i)Lcls = i = 1Kyi\log(pi)$$

Where:

- KKK = number of classes
- yiy_iyi = 1 if class i is correct, else 0
- pip_ipi = predicted probability for class i
-

b.Bounding Box Regression (again uses Smooth L1 loss)

$$Lreg = ix, y, w, hSmoothL1(titi)$$

4.results:

```
[20]: from IPython.display import Image as IPImage
      from IPython.display import display

      display(IPImage(filename="./images/faster_rcnn_1.gif"))
```

Fig.2. Faster R-CNN Detection Output on Urban Street Scene

The image displays an output frame from a Faster R-CNN object detection system running in a Jupyter notebook. It detects multiple objects like cars, traffic signs, traffic lights, and motorcycles in a real-world street scene. Each object is enclosed in a colored bounding box with its class label and confidence score. The frame also shows the detection speed as approximately 1.76 FPS, indicating real-time

performan

ce.

Fig.3. YOLOv8 Detection Output on Complex Urban Scene

The image shows the output of an object detection model applied to a busy urban street scene using YOLOv8. Numerous objects, including umbrellas, persons, cars, traffic signs, and sports balls, are detected and labeled with colored bounding boxes. The results summary at the bottom lists detected object counts and inference statistics such as processing time. The model processed the image at a resolution of 384×640 with a total time of 820.7 ms per image.

5.conculsion

This study demonstrated the effectiveness of combining Region Proposal Network (RPN) with Fast R-CNN for modular object detection in autonomous vehicles and compared it with the unified YOLOv8 model. While the RPN-Fast R-CNN pipeline offered robustness and adaptability across various sensor settings, YOLOv8 delivered superior real-time performance with a 95.2% detection accuracy. The analysis highlights that modular approaches excel in flexibility and fault tolerance, whereas end-to-end models like YOLOv8 are optimized for speed and precision. Ultimately, the choice of model depends on specific deployment requirements in autonomous systems.

6.future scope

1. **Integration with Multi-Sensor Systems**

Future developments can focus on enhancing detection accuracy by integrating data from multiple sensors, such as LiDAR, radar, and thermal cameras. This sensor fusion can provide richer contextual information and improve object detection in low-light or adverse weather conditions where traditional cameras may underperform.

2.**Edge Deployment and Optimization**

Optimizing object detection models for deployment on edge devices like embedded GPUs (e.g., NVIDIA Jetson) will make real-time object detection more feasible in commercial autonomous vehicles. Research into lightweight model compression, pruning, and quantization techniques can significantly reduce computational load without sacrificing accuracy.

3.Adapting to Unseen Scenarios and Environments

Enhancing the adaptability of detection systems to novel or unseen environments remains a crucial area. Techniques like domain adaptation, transfer learning, and few-shot learning can be explored to improve generalization across different geographic locations or traffic scenarios.

4.**Robustness Against Adversarial Attacks**

Future work could include improving the robustness of detection systems against adversarial inputs, which could be physical (like stickers on stop signs) or digital (manipulated pixel values). Ensuring system integrity in such situations is essential for safety in autonomous driving.

5.**3D Object Detection and Spatial Understanding**

Moving from 2D to 3D object detection will help autonomous systems understand the depth and spatial relationship between objects. This will be crucial for applications like obstacle avoidance, path planning, and behavior prediction in crowded environments.

6.**Collaboration with Vehicle-to-Everything (V2X) Technologies**

Integrating object detection with V2X communication can allow autonomous vehicles to receive real-time alerts from nearby infrastructure or vehicles, enhancing predictive awareness and reducing latency in decision-making.

7.REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2005, pp. 886–893.

[2] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

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

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

[6] G. Jocher et al., "YOLOv8: Real-Time Object Detection," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980–2988.