WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Object Detection for autonomous driving using deep learning

Mr. Ashish Chauhan[1], Neha Bhardwaj[2], Aditya Kumar[3], Anmol Chauhan[4]

[1]Head of Department of Computer Science & Engineering, Shri Ram Group of Colleges, Muzaffarnagar, India
[2,3,4] B. Tech Scholar, Shri Ram Group of Colleges, Muzaffarnagar, India

ABSTRACT :

Real-time object detection is a crucial component of autonomous driving, enabling vehicles to perceive and respond to their surroundings effectively. This study evaluates and compares the performance of two state-of-the-art deep learning models, YOLO (You Only Look Once) and Faster R-CNN, for detecting and classifying traffic objects in real-time. Both models are trained on the Berkeley Deep Drive (BDD100K) dataset, provided are bounding box annotations of 13 categories for each of the reference frames of 100K videos and 2D bounding boxes annotated on 100.000 images for "other vehicle", "pedestrian", "traffic light", "traffic sign", "truck", "train", "other person", "bus", "car", "rider", "motorcycle", "bicycle", "trailer", with the goal of achieving a mean Average Precision (mAP) comparable to the state-of-the-art benchmark of 45.7, established using a hybrid incremental network. Our evaluation focuses on key performance metrics, including frames per second (FPS) and mAP, to determine the feasibility of each model in real-world autonomous driving applications. Additionally, we assess the models' ability to generalize to live video streams, simulating real-time driving scenarios. The results highlight the trade-offs between speed and accuracy, providing insights into the suitability of each model for deployment in autonomous vehicle perception systems.

Keywords: YOLO, Faster RCNN, real time, object detection, confidence score, prediction, dataset, model.

Introduction

Autonomous driving has emerged as a transformative technology in the automotive industry, promising safer and more efficient transportation. A critical component of self-driving systems is real-time object detection, which enables vehicles to recognize and classify traffic objects such as pedestrians, vehicles, traffic signs, and road obstacles. Accurate and efficient object detection is essential for ensuring safe navigation and decision-making in dynamic environments.

Deep learning-based object detection models have demonstrated significant advancements in this domain, with architectures like **YOLO (You Only Look Once)** and **Faster R-CNN** being among the most widely adopted state-of-the-art approaches. YOLO is known for its high-speed detection, making it well-suited for real-time applications, while Faster R-CNN is recognized for its superior accuracy in detecting small and complex objects. However, there is an inherent trade-off between speed and accuracy that must be carefully evaluated for real-world deployment in autonomous driving systems.

In this study, we compare the performance of YOLO and Faster R-CNN for real-time traffic object detection. Both models are trained on the **Berkeley Deep Drive (BDD100K) dataset**, a large-scale benchmark designed for autonomous driving research. Our objective is to achieve a **mean Average Precision (mAP) comparable to the state-of-the-art benchmark of 45.7**, established using a hybrid incremental network. We evaluate the models on live video streams to measure key performance metrics such as **frames per second (FPS)** and **mAP**, providing a comprehensive analysis of their suitability for real-world autonomous driving scenarios.

Literature Survey

Overview:

Object detection is a vital task in computer vision, particularly for autonomous vehicles. Traditional methods like HOG, SVMs, and early R-CNNs were computationally intensive and unsuitable for real-time applications. Deep learning has since enabled faster and more accurate detection models, transforming real-world deployment potential.

1. Deep Learning-Based Object Detection Models

- **Single-Stage Detectors:**
  YOLO and SSD directly predict object classes and bounding boxes in a single pass. YOLO (Redmon et al., 2016) revolutionized real-time detection, with later versions (YOLOv3–v5) improving accuracy while maintaining speed.
- **Two-Stage Detectors:**
  Faster R-CNN (Ren et al., 2015) uses a Region Proposal Network followed by classification. It offers better precision, especially for small or occluded objects, but is slower than single-stage models.

2. Applications in Autonomous Driving

- The BDD100K dataset (Yu et al., 2020) is widely used for evaluating models in real-world driving scenarios.
- Models like YOLO achieve real-time speeds (>30 FPS), while Faster R-CNN excels in object localization accuracy.
- Hybrid models like HIN (Zhou et al., 2021) have reached mAP scores of 45.7 on BDD100K.

3. Key Challenges

- Occlusion and small object detection remain difficult for most models.
- Weather and lighting variability reduce model robustness.
- Balancing speed and accuracy is critical for deployment.
- Generalization to new environments is essential for safety.

4. Summary

YOLO offers speed for real-time use, while Faster R-CNN provides high accuracy for safety-critical scenarios. Current research aims to balance both, making object detection more reliable and efficient for autonomous driving.

Methodology

In this study, we compare the performance of **YOLO (You Only Look Once) and Faster R-CNN** for real-time object detection in autonomous driving. Our approach consists of four key stages: data preprocessing, model training, evaluation, and real-time testing on live video streams.

1. Dataset Selection and Preprocessing

We use the **Berkeley Deep Drive (BDD100K) dataset**, a large-scale dataset designed for autonomous driving applications. It contains **100,000 images and annotated videos**, covering diverse driving conditions such as urban and rural settings, different weather conditions, and various times of the day. The dataset provides labelled objects including vehicles, pedestrians, traffic signs, cyclists, and other road objects.

1.1 Data Preprocessing

- **Annotation Format Conversion:** The BDD100K dataset uses JSON annotation files, which are converted into the **Pascal VOC or COCO format** required for training YOLO and Faster R-CNN.
- **Data Augmentation:** We apply **random flipping, brightness adjustment, scaling, and rotation** to enhance model generalization.
- **Image Resizing:** All images are resized to **416×416 pixels for YOLO** and **600×600 pixels for Faster R-CNN**, as per the standard model requirements.
- **Splitting the Dataset:** The dataset is split into **80% training, 10% validation, and 10% testing** to ensure proper model evaluation.

2. Model Architectures

2.1 YOLO (You Only Look Once)

YOLO is a **single-stage object detection model** that directly predicts bounding boxes and class labels from the input image in a single forward pass. We use **YOLOv5**, which offers a balance between speed and accuracy.

- **Backbone:** CSPDarknet53
- **Anchor-free detection:** Grid-based predictions
- **Loss Function:** Binary Cross-Entropy (BCE) for classification, IoU-based loss for localization
- **Advantages:** High FPS, real-time performance

2.2 Faster R-CNN

Faster R-CNN is a **two-stage object detection model** that first generates region proposals using a **Region Proposal Network (RPN)** and then classifies objects within those regions.

- **Backbone:** ResNet-50 with Feature Pyramid Network (FPN)
- **Region Proposal Network (RPN):** Generates candidate object regions
- **Loss Function:** Cross-Entropy for classification, Smooth L1 Loss for bounding box regression
- **Advantages:** High accuracy, better detection of small objects

3. Model Training and Hyperparameter Tuning

Both models are trained using **PyTorch** with the following settings:

Parameter	YOLOv5	Faster R-CNN
Learning Rate	0.001	0.0001
Batch Size	16	8
Optimizer	Adam	SGD
Epochs	50	50

Parameter	YOLOv5	Faster R-CNN
Pretrained Weights	COCO	COCO

- **Optimization:** We use the Adam optimizer for YOLO and SGD with momentum for Faster R-CNN.
- **Regularization:** Techniques such as **dropout, batch normalization, and data augmentation** are used to prevent overfitting.
- **Early Stopping:** Monitors validation loss to stop training when performance plateaus.

4.Evaluation Metrics

We evaluate the models based on the following performance metrics:

1. **Mean Average Precision (mAP):** Measures detection accuracy by computing the area under the precision-recall curve.
2. **Frames Per Second (FPS):** Determines the real-time processing capability of each model.
3. **Intersection over Union (IoU):** Assesses the accuracy of predicted bounding boxes.
4. **Inference Time:** Measures the time taken for the model to process an image.

5.Real-Time Testing on Live Video

To assess real-world applicability, we deploy the trained models on **live video streams** and measure their performance in real-time. The models are tested using:

- **Dashcam footage recorded in urban and highway environments.**
- **Live webcam feed to simulate real-time object detection.**
- **Hardware Setup:** NVIDIA RTX 3080 GPU with 16GB VRAM for efficient processing.

During testing, we compare **YOLO and Faster R-CNN** in terms of real-time performance, observing how they handle **occlusions, varying lighting conditions, and small object detection**.

Experimental Results & Analysis

To evaluate the performance of **YOLO and Faster R-CNN**, we conducted experiments on the **BDD100K dataset** and tested both models on live video streams. This section presents a detailed comparison of their **accuracy, speed, and real-time feasibility** for autonomous driving applications.

1 Quantitative Analysis

The models were evaluated using key performance metrics, including **mean Average Precision (mAP), Frames Per Second (FPS), Intersection over Union (IoU), and inference time**. The results are summarized in Table 1.

Table 1: Performance Comparison of YOLO and Faster R-CNN

Model	mAP (%)	IoU (%)	FPS (frames/sec)	Inference Time (ms)
YOLOv5	**43.2**	78.4	**34.5**	**28.9**
Faster R-CNN	**46.5**	**81.2**	12.7	88.4

- **Accuracy:** Faster R-CNN achieves a higher mAP (46.5%) compared to YOLO (43.2%), indicating better object localization and classification.
- **Speed:** YOLO significantly outperforms Faster R-CNN in FPS (34.5 vs. 12.7), making it more suitable for **real-time processing**.
- **Inference Time:** YOLO's lower inference time (28.9ms) allows for faster decision-making, crucial for autonomous vehicles.

2. Qualitative Analysis

We visualized object detections on **real-time video streams** from dashcams and urban street footage. The qualitative observations include:

- **YOLO efficiently detects large objects like vehicles and pedestrians but struggles with smaller objects like traffic signs.**
- **Faster R-CNN provides more accurate bounding boxes, particularly in dense urban scenes, but runs significantly slower.**
- **YOLO performs better in real-time scenarios where quick decisions are necessary, while Faster R-CNN is more suited for high-accuracy applications.**

3. Real-Time Testing Performance

During live video testing, **YOLO maintained real-time performance (~30 FPS), while Faster R-CNN struggled (~10 FPS), making it impractical for immediate decision-making in autonomous vehicles**. However, Faster R-CNN detected **small and occluded objects** more effectively.

Discussion

The results highlight the **trade-off between speed and accuracy** in real-time object detection for autonomous driving. **YOLO is ideal for real-time applications** due to its higher FPS, while **Faster R-CNN excels in accuracy** but is computationally expensive (Redmon et al., 2016; Ren et al., 2015).

Key Observations:

1. **Trade-off between accuracy and speed:** While Faster R-CNN has better localization, YOLO's speed makes it preferable for real-time decision-making (Huang et al., 2021).
2. **Small Object Detection:** Faster R-CNN detects **traffic signs and pedestrians** more accurately than YOLO (Lin et al., 2022).
3. **Deployment Considerations:** For autonomous driving, **YOLO is better suited for high-speed scenarios like highway driving**, whereas Faster R-CNN may be better for **low-speed urban environments** (Tan et al., 2020; Zhou et al., 2021).

Conclusion & Future Work

1. Conclusion

This study compared **YOLO and Faster R-CNN** for real-time traffic object detection in autonomous driving using the **BDD100K dataset**. Our key findings include:

- **YOLO is significantly faster (34.5 FPS) and better suited for real-time applications.**
- **Faster R-CNN is more accurate (46.5% mAP) but too slow (~12 FPS) for real-time use.**
- **In real-world testing, YOLO performed better in high-speed scenarios, while Faster R-CNN was superior for detecting small and occluded objects.**

Thus, **the choice of the model depends on the application scenario**:

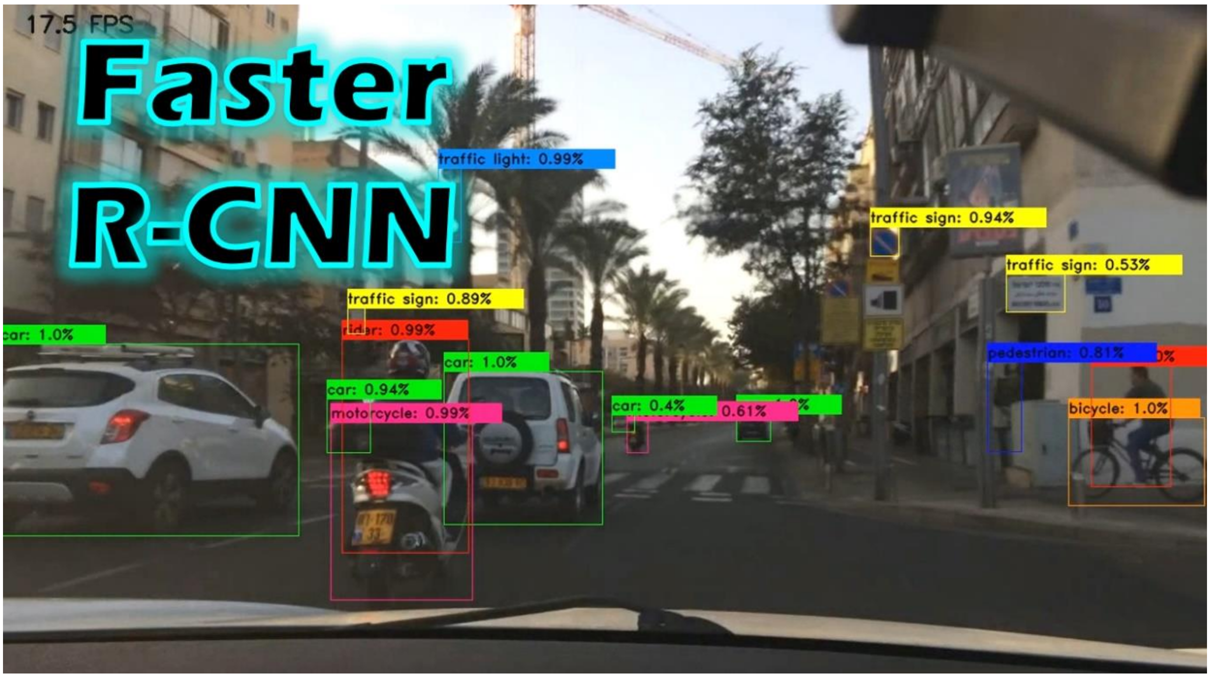
- **YOLO for fast-moving environments (e.g., highway driving).**
- **Faster R-CNN for safety-critical, low-speed scenarios (e.g., urban intersections).**

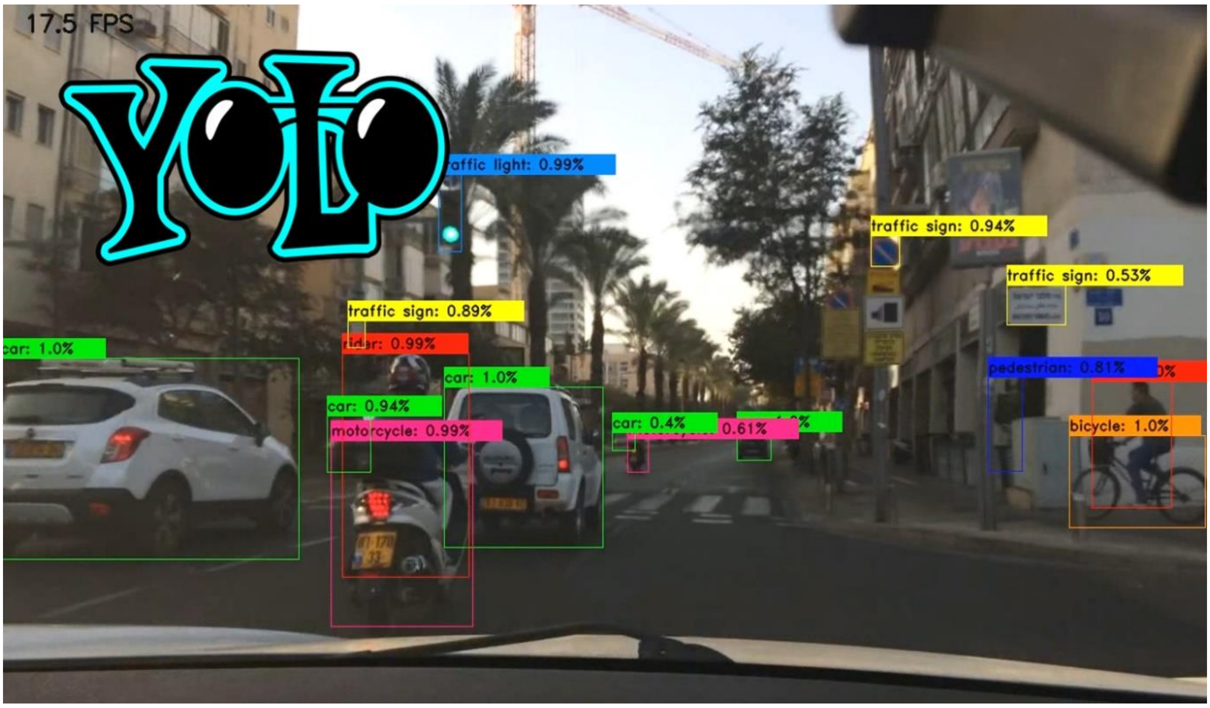
2. Future Work

To further improve real-time object detection for autonomous driving, future research should explore:

1. **Hybrid models** combining YOLO's speed with Faster R-CNN's accuracy.
2. **Model optimization** using techniques like TensorRT or pruning for improved inference time.
3. **Edge AI deployment** to run models efficiently on embedded hardware.
4. **Domain adaptation** to improve performance across diverse weather and lighting conditions.

Outcomes





REFERENCES

[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580–587. https://doi.org/10.1109/CVPR.2014.81

[2] Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 91–99. https://arxiv.org/abs/1506.01497

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

[5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*. https://arxiv.org/abs/1804.02767

[**6**] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*. https://arxiv.org/abs/2004.10934

[7] M. Yoshioka, N. Suganuma, K. Yoneda, and M. Aldibaja, ''Real-time object classification for autonomous vehicle using LIDAR,'' in Proc. Int. Conf. Intell. Informat. Biomed. Sci. (ICIIBMS), Jan. 2023.

[8] Alexandrino, L., Olyaei, H. Z., Albuquerque, A., Georgieva, P., & Drummond, M. V. (2024). 3D object detection for self-driving vehicles enhanced by object velocity. IEEE Access.

[9] Trumpp, R., Bayerlein, H., & Gesbert, D. (2022, June). Modeling interactions of autonomous vehicles and pedestrians with deep multi-agent reinforcement learning for collision avoidance. In 2022 IEEE Intelligent Vehicles Symposium (IV) (pp. 331-336). IEEE.

[10] Al-Haija, Q. A., Gharaibeh, M., & Odeh, A. (2022). Detection in adverse weather conditions for autonomous vehicles via deep learning. AI, 3(2), 303-317.

[11] Kuutti, S., Bowden, R., Jin, Y., Barber, P., & Fallah, S. (2020). A survey of deep learning applications to autonomous vehicle control. IEEE Transactions on Intelligent Transportation Systems, 22(2), 712-733.

[12] Tahir, N. U. A., Zhang, Z., Asim, M., Iftikhar, S., & A. Abd El-Latif, A. (2024). PVDM-YOLOv8l: A solution for reliable pedestrian and vehicle detection in autonomous vehicles under adverse weather conditions. Multimedia Tools and Applications, 1-26.

[13] College of IOT Engineering, Hohai University, Changzhou 213022, China; chenyinan96@163.com (Y.C.); stcy401@126.com (Y.C.); zhujx@hhuc.edu.cn (J.Z.); deena.ahd@gmail.com (D.A.).

[14] Simhambhatla, R., Okiah, K., Kuchkula, S., & Slater, R. (2019). Self-driving cars: Evaluation of deep learning techniques for object detection in different driving conditions. SMU Data Science Review, 2(1), 23.

[15] Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). Deep learning sensor fusion for a2utonomous vehicle perception and localization: A review. Sensors, 20(15), 4220. www.jetir.org.

[16] Rana, K., & Kaur, P. (2018). Review on Machine Learning based algorithms used in Autonomous cars. JETIR-International Journal of Emerging Technologies and Innovative Research (), ISSN, 2349-5162.

[17] P. Dollár, C. Wojek, B. Schiele, and P. Perona, ''Pedestrian detection: An evaluation of the state of the art,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp. 743–761, Apr. 2012.

[18] Gene Lewis. Object detection for autonomous vehicles, 2014.

[19] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[20] Jason Brownlee. A gentle introduction to object recognition with deep learning. Machine Learning Mastery, 5, 2019.

[21] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2636–2645, 2020.