A Performance Comparison of Object Detection Algorithms on Traffic Scenes in Indian Roads

Bhakti Paranjape
Dr. Vishwanath Karad MIT World Peace University, Pune, Bharat, India
bhakti.paranjape@mitwpu.edu.in (corresponding author)

Apurva Naik
Dr. Vishwanath Karad MIT World Peace University, Pune, Bharat, India
apurva.naik@mitwpu.edu.in

S. Perumal Sankar
Toc H Institute of Science and Technology, Ernakulam, Bharat, India
spsankar2004@gmail.com

Received: 23 March 2025 | Revised: 30 April 2025, 19 May 2025, 6 June 2025, 15 June 2025, and 19 June 2025 | Accepted: 21 June 2025

 DOI: https://doi.org/10.48084/etasr.11105

ABSTRACT

Machine learning-based object detection allows machines to decipher visual information and recognize objects in digital images or videos using localization and classification techniques. This study focuses on applying object detection techniques to traffic images from Indian roads, which are unstructured and have complicated traffic patterns. Taking into account the high number of traffic accidents in India, it is imperative to develop intelligent systems for traffic analysis and management. This study uses four cutting-edge object detection algorithms, SSD, YOLO, Faster R-CNN, and CenterNet, previously trained on the popular COCO and PASCAL VOC datasets. These algorithms are tested on the DATS dataset, created to represent road conditions in India, to examine the capacity of these models to manage the complexities of Indian traffic situations. In terms of mAP, the results show that CenterNet had the lowest score (69.5%) and YOLOv3 the highest (81.5%).

Keywords-DATS; deep learning; object detection; traffic scenes

I. INTRODUCTION

Object detection is a fundamental task in computer vision that involves the identification and localization of multiple objects within an image or video. It not only determines what objects are present but also where they are located, using bounding boxes. Powered by deep learning models, object detection has enabled significant advancements in various domains, one of the most impactful being intelligent transportation, which includes traffic surveillance and road safety. Object detection can address challenges by tracking key road elements such as vehicles, pedestrians, and traffic signs, helping to monitor traffic, detect violations, and improve safety. These systems are central to ADAS and intelligent transportation, requiring high-quality and diverse datasets for effective real-world performance.

The Ministry of Road Transport and Highways reports that around 80,000 people die in road accidents annually in India, mainly due to high speeds, unsafe roads, and distracted driving [1]. India's vast road network of over five million kilometers handles 90% of passenger and 65% of freight traffic [2]. Unlike structured traffic in developed countries, Indian roads often exhibit unstructured traffic with diverse conditions, highlighting the need for a dedicated dataset capturing these complexities. The Dataset for Indian Traffic Scenes (DATS) is specifically curated to address these challenges [3, 4].

Object detection algorithms are broadly classified into two types: two-stage detectors and single-stage detectors. Two-stage detectors, such as Region-based Convolutional Neural Networks (R-CNN) and Faster R-CNN [5, 6], first generate region proposals where objects might be located and then perform classification and bounding box regression on those regions. These methods are generally more accurate but computationally intensive. In contrast, single-stage detectors such as You Only Look Once (YOLO) and Single Shot multiBox Detector (SSD) perform object localization and classification in a single forward pass through the network, making them significantly faster and suitable for real-time applications, although sometimes with a slight trade-off in accuracy. The evolution of object detection literature reflects ongoing efforts to balance accuracy, speed, and computational efficiency for various practical use cases. In [7], a deep CNN used Region of Interest (RoI) pooling to extract features, enabling simultaneous object classification and bounding box

regression in one pass. YOLO [8] uses a fully convolutional approach for end-to-end detection, predicting object positions and classifications in a single shot. An enhanced version of YOLO [9] introduces an inception model and a pooling pyramid layer. SSD [10] simplifies detection by merging the proposal and resampling steps into one network, using multiple bounding boxes across feature maps for efficient real-time performance. In [11], it was highlighted that two-stage algorithms offer higher accuracy but slower speeds, while single-shot models are faster with competitive accuracy, along with a comprehensive discussion of detection methods, applications, and datasets. Specialized tasks such as the detection of salient objects, faces, and pedestrians were explored in [12], with suggestions for improving performance in each domain. Recent innovations include SpineNet [13], a backbone architecture that optimizes intermediate features and cross-scale connections using a neural network search. In [5, 14-25], the rapid evolution of the field of object detection was explored, with ongoing research exploring novel architectures and optimizations, reflecting diverse approaches to improve detection speed, accuracy, and real-time performance.

The key contributions of this study are as follows:

- Uses a domain-specific dataset, DATS, with real-world images of Indian roads that were captured using high-resolution cameras on Android devices. The dataset was collected across Maharashtra, Goa, and Gujarat, covering diverse road types (urban, rural, highways, unpaved roads, etc.), was manually annotated with LabelIMG [26], and features unique object categories, such as rickshaws, bullock carts, animals, and pedestrians.
- Implemented and evaluated YOLO, SSD, CenterNet, and Faster R-CNN with various combinations of backbone networks (e.g., ResNet, MobileNet, Inception) and input resolutions.
- DATS includes 53 object classes covering a wide range of objects seen in Indian traffic images, and is more comprehensive than other Indian traffic datasets for object detection.

II. DATS DATASET

A comprehensive dataset with a large number of images is crucial for successful deep learning algorithms [11, 14]. The DATS dataset contains more than 15,000 high-resolution images captured with the cameras of Redmi Note 8 Pro and Note 5 Pro smartphones. It includes scenes from urban and rural areas in Maharashtra, Goa, and Gujarat, covering unstructured roads, dense traffic, various road types, stray animals, rule-breaking pedestrians, and poorly maintained or unpaved roads. The images in Figure 1 depict common scenes on Indian roads. There are different classes of vehicles that are mostly found only in Indian regions. Indian roads feature domestic animals, such as cows, bullocks, dogs, goats, sheep, and horses, since they are often taken for grazing, particularly in rural agricultural areas. The images are preprocessed through normalization, resizing, and augmentation.

III. EXPERIMENTATION

This study employed state-of-the-art object detection models, namely SSD, YOLO, Faster R-CNN, and CenterNet, to evaluate their performance on the DATS dataset. Multiple pretrained variants of these models were downloaded from the GluonCV Model Zoo [27-28] and used for inference. These variants differ in input resolution, backbone architectures, and training datasets. These models were originally trained on the PASCAL VOC [29] and Common Objects in Context (COCO) [30] datasets, which are generic and not tailored to any specific domain, but instead consist of everyday objects in natural scenes. The models were tested on DATS, which includes region-specific object classes.

Their performance was evaluated using mean Average Precision (mAP) for models trained on PASCAL VOC, Box Average Precision (BoxAP) for models trained on COCO, and inference time to reflect real-time usability. To compute the mAP values, pretrained models from the GluonCV model Zoo were tested on the DATS dataset. The models predicted bounding boxes, class labels, and confidence scores, which were then compared against ground-truth annotations using Intersection over Union (IoU), typically with a threshold of 0.5. Precision and recall were calculated for each class, followed by calculating the Average Precision (AP) from the precision-recall curves. The mean of these APs across all classes yielded the final mAP. GluonCV is a deep learning toolkit that provides pre-trained models for computer vision tasks such as object detection. Pycocotools is a Python API for loading, visualizing, and evaluating COCO-format datasets commonly used for computing mAP in object detection tasks. A drop in mAP was observed for objects such as rickshaws and tempos, which were absent from the training datasets, leading to misclassifications.

All models were tested on the complete DATS dataset. For each object detected in an image, the corresponding confidence score generated by the model was recorded. This process was repeated across all images, and for each model, the average confidence score per object class was computed. These averaged confidence scores provide an overall measure of the model's confidence in identifying various objects within the dataset. Box AP values represent the model's ability to localize and predict objects in images based on the COCO dataset. They are derived by evaluating the precision and recall of predicted bounding boxes against ground truth boxes at various IoU thresholds, typically 0.5 and 0.75. Table II shows three AP values: the first is for IoU=0.5, the second for IoU=0.75, and the third is mAP, which averages precision across multiple IoU thresholds. Higher AP values indicate better object detection and localization performance. The implementation was carried out on Google Colab, which provided access to a cloud-based NVIDIA Tesla T4 GPU (16 GB VRAM), with 12 GB of system RAM. The models were tested using MXNet version 1.9.1 and GluonCV version 0.10.5, without any retraining or fine-tuning. The runtime provided approximately 12 GB of RAM and a temporary disk space of 70 GB.

Fig. 1. Examples of unstructured roads, vehicle categories such as bikes, bullock carts, and autorickshaws, and different animals seen on roads

IV. RESULT

Among the models, YOLOv3 Darknet53 (416×416) achieved the best balance with a good confidence score (0.84) and the highest mAP (81.5%), making it highly accurate and reliable. SSD_512_ResNet50 models (original and int8) showed very high confidence (0.92–0.93) but slightly lower mAP (80.1%), indicating high certainty but marginally less accuracy than YOLOv3. CenterNet models had lower confidence (0.57-0.7) but still delivered reasonable mAP (78-79%), suggesting cautious but fairly accurate predictions. MobileNet-based YOLO and SSD models maintained moderate confidence (0.8) with lower mAP (73-75%), favoring lightweight and faster inference. Several models used in this study were int8 quantized versions, which are optimized for faster inference and reduced memory footprint by using 8-bit integer precision instead of standard 32-bit floating-point representations. These quantized models show minimal performance loss while improving efficiency, making them suitable for real-world deployment.

YOLOv3 Darknet53 (608×608) achieved the highest Box AP (37.0/58.2/40.1) with moderate confidence (0.74), while Faster R-CNN models showed the best overall accuracy, with Faster R-CNN ResNet101_v1d reaching the highest Box AP (40.8/62.4/44.7) with decent confidence (0.64-0.82).). YOLOv3 models (416×416) balanced good confidence (0.80-0.85) with high Box AP (36.0/57.2/38.7). SSD models performed moderately (Box AP: 25-30) with higher confidence (up to 0.88). CenterNet models were less confident (0.51-0.64) but steadily improved accuracy with deeper backbones and DCNv2. Overall, Faster R-CNN offers top precision, YOLOv3 balances speed and accuracy, and SSD suits faster but less accurate needs.

Table I presents the confidence scores and mAP for models pre-trained on the PASCAL VOC dataset, and Table II presents the Box AP and confidence scores for the models trained on the COCO dataset. Figures 2 and 3 show the time taken by each model, trained on PASCAL VOC and COCO datasets, respectively, to generate detections. ResNet models took the longest to generate detections due to their deeper, more complex architecture, offering higher accuracy but requiring more resources. As shown in Figures 4 and 5, models pre-trained on COCO generated the most detections. YOLOv3 and Faster R-CNN detected all eight objects, but Faster R-CNN had some misdetections, while YOLOv3 performed error-free, highlighting its robustness in complex traffic scenarios. Lighter models, such as MobileNet and VGG, are more efficient but less accurate. Models trained on the PASCAL VOC dataset performed better than those trained on COCO, likely due to the VOC dataset's focus on 20 object categories, which align better with the DATS traffic scene focus, whereas COCO's 80 categories led to more misclassifications.

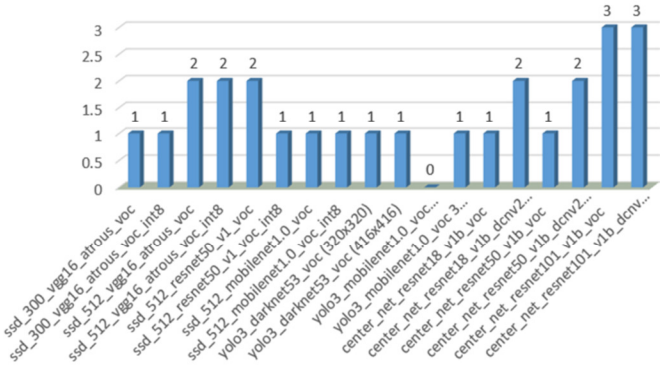

Fig. 2. Time required by each model trained on the Pascal-VOC dataset for object detection on each image.

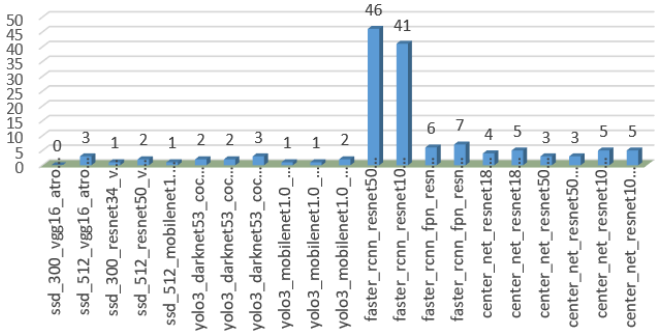

Fig. 3. Time required by each model trained on the COCO dataset for object detection on each image.

TABLE I. MAP AND PREDICTION SCORES FOR DIFFERENT MODELS TRAINED ON THE PASCAL-VOC DATASET AND TESTED ON THE DATS DATASET

Models pre-trained on the VOC dataset	Confidence scores	mAP (%)
yolo3_darknet53_voc (416×416)	0.84	81.5
ssd_512_resnet50_v1_voc_int8	0.92	80.16
ssd_512_resnet50_v1_voc	0.93	80.1
yolo3_darknet53_voc (320×320)	0.87	79.3
center_net_resnet101_v1b_dcnv2_voc	0.66	79.2
ssd_512_vgg16_atrous_voc	0.91	79.2
center_net_resnet50_v1b_dcnv2_voc	0.70	78.7
ssd_512_vgg16_atrous_voc_int8	0.91	78.39
center_net_resnet101_v1b_voc	0.57	78.2
ssd_300_vgg16_atrous_voc	0.90	77.6
ssd_300_vgg16_atrous_voc_int8	0.90	77.46
center_net_resnet50_v1b_voc	0.58	76.1
yolo3_mobilenet1.0_voc3 (416×416)	0.82	75.8
ssd_512_mobilenet1.0_voc	0.82	75.4
ssd_512_mobilenet1.0_voc_int8	0.83	75.04
center_net_resnet18_v1b_dcnv2_voc	0.62	74.7
yolo3_mobilenet1.0_voc (320×320)	0.74	73.9
center_net_resnet18_v1b_voc	0.60	69.5

TABLE II. MAP AND PREDICTION SCORES FOR DIFFERENT MODELS TRAINED ON THE COCO DATASET AND TESTED ON THE DATS DATASET

Models pre-trained on the COCO dataset	Confidence scores	Box AP
ssd_300_vgg16_atrous_coco	0.69	25.1/42.9/25.8
ssd_300_resnet34_v1b_coco	0.81	25.1/41.7/26.2
ssd_512_mobilenet1.0_coco	0.79	21.7/39.2/21.3
yolo3_mobilenet1.0_coco(320×320)	0.79	26.7/46.1/27.5
yolo3_mobilenet1.0_coco(416×416)	0.85	28.6/48.9/29.9
ssd_512_resnet50_v1_coco	0.80	30.6/50.0/32.2
yolo3_darknet53_coco(320×320)	0.85	33.6/54.1/35.8
yolo3_darknet53_coco(416×416)	0.80	36.0/57.2/38.7
yolo3_mobilenet1.0_coco(608×608)	0.81	28.0/49.8/27.8
ssd_512_vgg16_atrous_coco	0.88	28.9/47.9/30.6
yolo3_darknet53_coco(608×608)	0.74	37.0/58.2/40.1
center_net_resnet50_v1b_coco	0.64	32.1/33.4
center_net_resnet50_v1b_dcnv2_coco	0.64	34.0/35.3
center_net_resnet18_v1b_coco	0.51	26.6/28.1
center_net_resnet18_v1b_dcnv2_coco	0.55	28.9/30.3
center_net_resnet101_v1b_coco	0.61	34.5/35.8
center_net_resnet101_v1b_dcnv2_coco6	0.60	35.8/37.1
faster_rcnn_fpn_resnet50_v1b_coco	0.70	38.4/60.2/41.6
faster_rcnn_fpn_resnet101_v1d_coco	0.64	40.8/62.4/44.7
faster_rcnn_resnet101_v1d_coco	0.82	40.1/60.9/43.3
faster_rcnn_resnet50_v1b_coco	0.66	37.0/57.8/39.6

As shown in Figures 4 and 5, models trained on the COCO dataset tend to detect more objects, but this comes at the cost of accuracy. For example, while YOLOv3 and Faster R-CNN detected all eight objects in the given image, Faster R-CNN exhibited misdetections due to the wide variety of classes in the COCO dataset, which may lead to confusion during object classification. On the other hand, YOLOv3 showed better accuracy, detecting all objects without misclassification, which indicates its robustness in handling complex real-world scenarios, such as those present in the DATS dataset. CenterNet, when trained on the COCO dataset, performed poorly compared to when trained on the PASCAL VOC dataset. This may be due to the increased complexity and wider variety of object categories in COCO, which could detract from the model's ability to accurately detect objects relevant to the DATS dataset, such as vehicles and pedestrians. Furthermore, it was found that as the number of layers in the network architecture increases, the models tend to provide better detections. Faster-RCNN models required more time to generate detections compared to other models due to the inherently slower nature of the R-CNN family, which involves more computational steps for region proposal generation and classification.

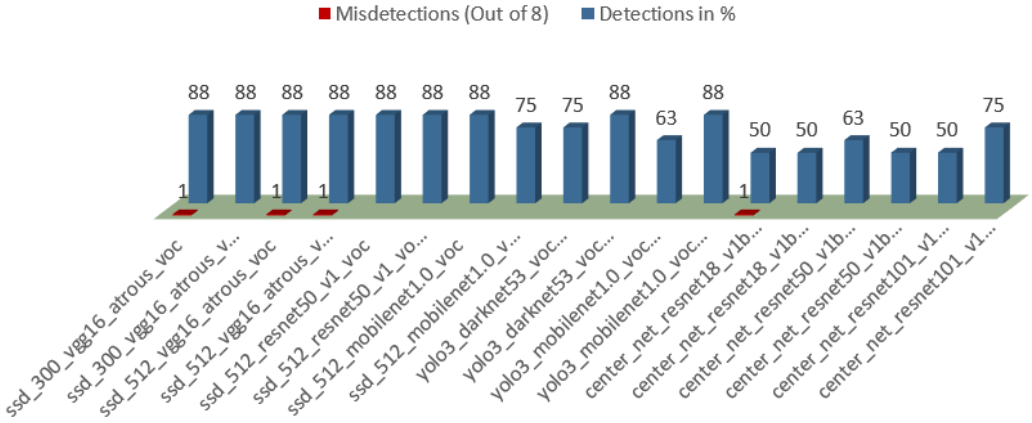

Fig. 4. Detections in percentage and misdetections by each model trained on the Pascal-VOC dataset.

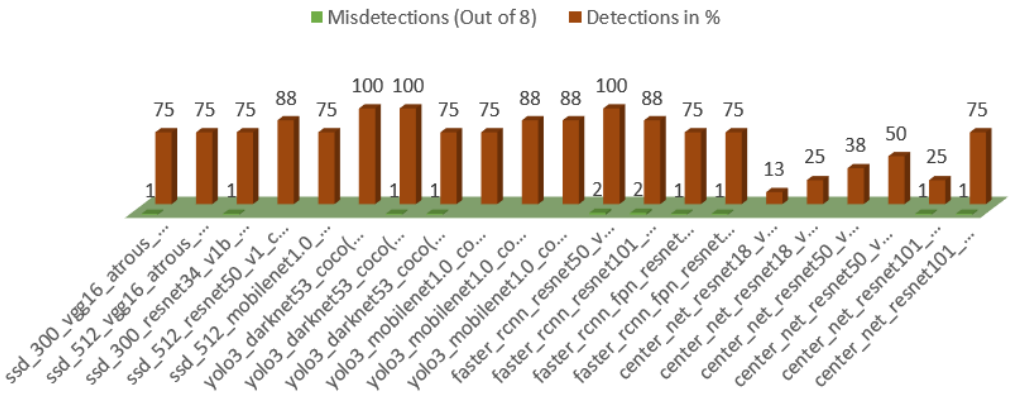

Fig. 5. Detections in percentage and misdetections by each model trained on the COCO dataset.

Figure 6 illustrates the output of the SSD model trained on both the COCO and Pascal VOC datasets. The image size used for testing is 300 for the COCO-trained SSD model and 512 for the Pascal VOC-trained SSD model. The SSD model trained on the VOC dataset successfully detected the objects in the image, whereas the model trained on the COCO dataset misclassified some objects when tested with an image of size 300. However, when the image size increased to 512, the SSD model trained on the COCO dataset was able to detect most objects correctly.

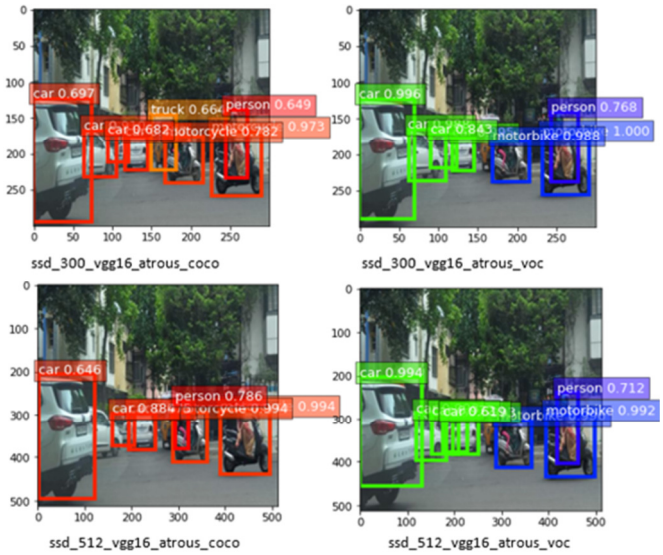

Fig. 6. Images of the SSD model trained with VGG16 on images of sizes 300×300 and 512×512 from the Pascal-VOC and COCO datasets.

This suggests that increasing the input image size can improve detection performance of models trained on more complex datasets such as COCO. Figure 7 shows the output of CenterNet with ResNet as its backbone network, using 18, 50, and 101 layers. When an image from the DATS dataset was input into the model, it was observed that only a few objects were detected. For instance, a car was misclassified as a truck. However, as the number of layers in the ResNet backbone increased, the number of detected objects also improved. With the ResNet-101 model, all objects in the image were correctly detected, indicating that deeper networks with more layers are better equipped to capture complex features and provide more accurate object detection. This shows the benefit of using deeper architectures for improved detection in traffic scene applications, where more intricate and varied objects need to be identified.

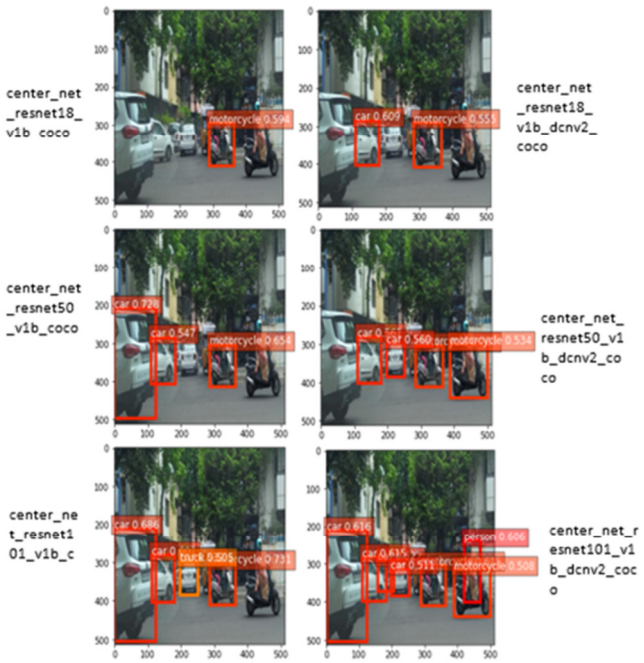

Fig. 7. Images of the CenterNet model trained with the ResNet network with 18, 50, and 101 layers on images sized 512×512 from the COCO dataset.

Table III provides a comparative study of the pre-trained models, summarizing various object detection models (SSD, YOLOv3, CenterNet, and Faster R-CNN), detailing their architecture, year of introduction, backbone networks, number of layers, training image sizes, and the number of classes supported in the three datasets.

TABLE III. MODELS WITH DIFFERENT BACKBONE NETWORKS, LAYERS, AND NUMBER OF TRAINING IMAGES

Model	Year	Backbone network	Layers	Size of the training images	Number of classes in the Pascal-VOC Dataset	Number of classes in the COCO Dataset	Number of classes in the DATS dataset
SSD	2016	VGG16	16	300×300	20 (Person: person Animal: bird, cat, cow, dog, horse, sheep. Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train. Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor. Train/validation/test: 9,963 images containing 24,640 annotated objects)	80 (Natural images that reflect everyday scenes with multiple objects and providing contextual information. 165,482 train, 81,208 validation, and 81,434 test images. 2,500,000 labeled instances in 328,000 images)	54 (Images from typical Indian roads covering all objects seen in traffic scenarios. 15,000+ images containing 7,971 annotations) (In a process of collecting images to increase the dataset)
				512×512			
		ResNet	50	512×512			
		Mobilenet	28	512×512			
YOLOv3	2016	Darknet	53	320×320			
				416×416			
		Mobilenet	28	320×320			
				416×416			
CenterNet	2019	ResNet	18	512×512			
			50	512×512			
			101	512×512			
Faster-RCNN	2016	ResNet	50	512×512			
			101	512×512			

V. CONCLUSION

This study focused on the comparative evaluation of four object detection models, namely SSD, YOLO, Faster-RCNN, and CenterNet, utilizing various backbone networks such as VGG16, ResNet (18, 50, 101 layers), MobileNet, and Darknet. The models were trained on Pascal VOC and COCO datasets with image sizes of 300×300, 416×416, and 512×512, and tested on the DATS dataset. YOLO outperformed others with an mAP of 81.5% and the highest number of correct detections. The study suggests that training on a custom dataset tailored to traffic scenes could further improve performance, highlighting the need for more realistic datasets. The study identified challenges such as accurately detecting similar-looking objects (e.g., cars, trucks, buses) and smaller objects (e.g., pedestrians, bicycles, and traffic signs). Misclassifications often occur with partially obscured objects or in complex environments, affecting the reliability of the detection. Addressing these issues is crucial to improving the reliability and accuracy of object detection models in dynamic and cluttered traffic environments, such as those found on Indian roads.

REFERENCES

[1] "About Road Safety," *Ministry of Road Transport & Highways, Government of India*. https://morth.nic.in/en.

[2] "Topic: Road accidents in India," *Statista*. https://www.statista.com/topics/5982/road-accidents-in-india/.

[3] B. Paranjape, "DATS_2020." Mendeley Data, Apr. 29, 2021, https://doi.org/10.17632/nfc34n8svj.1.

[4] B. A. Paranjape and A. A. Naik, "DATS_2022: A Versatile Indian Dataset for Object Detection in Unstructured Traffic Conditions," *Data in Brief*, vol. 43, Aug. 2022, Art. no. 108470, https://doi.org/10.1016/j.dib.2022.108470.

[5] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, "Object Detection Using Convolutional Neural Networks," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South), Oct. 2018, pp. 2023–2027, https://doi.org/10.1109/TENCON.2018.8650517.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, https://doi.org/10.1109/TPAMI.2016.2577031.

[7] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[9] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. U. Haq, "Object Detection through Modified YOLO Neural Network," *Scientific Programming*, vol. 2020, pp. 1–10, Jun. 2020, https://doi.org/10.1155/2020/8403262.

[10] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 21–37.

[11] L. Jiao *et al.*, "A Survey of Deep Learning-Based Object Detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, https://doi.org/10.1109/ACCESS.2019.2939201.

[12] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, https://doi.org/10.1109/TNNLS.2018.2876865.

[13] X. Du *et al.*, "SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11589–11598, https://doi.org/10.1109/CVPR42600.2020.01161.

[14] P. Wang, X. Wang, Y. Liu, and J. Song, "Research on Road Object Detection Model Based on YOLOv4 of Autonomous Vehicle," *IEEE Access*, vol. 12, pp. 8198–8206, 2024, https://doi.org/10.1109/ACCESS.2024.3351771.

[15] Y. Zhou and H. Li, "A Survey of Dense Object Detection Methods Based on Deep Learning," *IEEE Access*, vol. 12, pp. 179944–179961, 2024, https://doi.org/10.1109/ACCESS.2024.3507820.

[16] H. Wang, Y. Xu, Z. Wang, Y. Cai, L. Chen, and Y. Li, "CenterNet-Auto: A Multi-object Visual Detection Algorithm for Autonomous Driving Scenes Based on Improved CenterNet," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 742–752, Jun. 2023, https://doi.org/10.1109/TETCI.2023.3235381.

[17] S. Ennaama, H. Silkan, A. Bentajer, and A. Tahiri, "Enhanced Real-Time Object Detection using YOLOv7 and MobileNetv3," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19181–19187, Feb. 2025, https://doi.org/10.48084/etasr.8777.

[18] W. Zhou, C. Wang, J. Xia, Z. Qian, and Y. Wu, "Monitoring-Based Traffic Participant Detection in Urban Mixed Traffic: A Novel Dataset and A Tailored Detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 189–202, Jan. 2024, https://doi.org/10.1109/TITS.2023.3304288.

[19] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, "YOLOv8-QSD: An Improved Small Object Detection Algorithm for Autonomous Vehicles Based on YOLOv8," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–16, 2024, https://doi.org/10.1109/TIM.2024.3379090.

[20] P. Mutabarura, N. Muchuka, and D. Segera, "Comparative Evaluation of YOLO Models on an African Road Obstacles Dataset for Real-Time Obstacle Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19045–19051, Feb. 2025, https://doi.org/10.48084/etasr.9135.

[21] A. Danti, J. Y. Kulkarni, and P. S. Hiremath, "An Image Processing Approach to Detect Lanes, Pot Holes and Recognize Road Signs in Indian Roads," *International Journal of Modeling and Optimization*, pp. 658–662, 2012, https://doi.org/10.7763/IJMO.2012.V2.204.

[22] A. Padiath, L. Vanajakshi, S. C. Subramanian, and H. Manda, "Prediction of traffic density for congestion analysis under Indian traffic conditions," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2009, pp. 1–6, https://doi.org/10.1109/ITSC.2009.5309716.

[23] M. J. Shaifee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video," *Journal of Computational Vision and Imaging Systems*, vol. 3, no. 1, Oct. 2017, https://doi.org/10.15353/vsnl.v3i1.171.

[24] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection," in *2018 15th Conference on Computer and Robot Vision (CRV)*, Toronto, Canada, May 2018, pp. 95–101, https://doi.org/10.1109/CRV.2018.00023.

[25] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 10778–10787, https://doi.org/10.1109/CVPR42600.2020.01079.

[26] "HumanSignal/labelImg." HumanSignal, Jun. 29, 2025, [Online]. Available: https://github.com/HumanSignal/labelImg.

[27] J. Guo *et al.*, "GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing," *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.

[28] "Model Zoo — gluoncv 0.11.0 documentation." https://cv.gluon.ai/model_zoo/index.html.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge,"

International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun. 2010, https://doi.org/10.1007/s11263-009-0275-4.

[30] T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 740–755.