Journal of Engineering Science and Technology Review 18 (2) (2025) 125- 131

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

MIRYO: A Hybrid Model for Detecting Vehicles in Noisy Images

Nikita Singhal[1,*] and Lalji Prasad[2]

[1]*Department of Computer Engineering, Army Institute of Technology, Pune, MH, India*
[2]*Institute of Advanced Computing, SAGE University, Indore, MP, India*

Received 4 October 2024; Accepted 4 March 2025

Abstract

Detection of vehicles is a key task in many smart transportation applications, involving traffic management, road infrastructure, autonomous driving, and other challenges that arise due to the daily growth in vehicle numbers. Several deep learning (DL) based techniques have previously been explored and investigated by the researchers for vehicle detection, but vehicle detection in noisy images is still considered a difficult task. Low-light, low-resolution, and other environmental noise have a substantial impact on images, significantly reducing vehicle detection system performance. In this paper, we implemented MIRYO, a hybrid model for detection of vehicles in challenging images based on MirNet-v2 and modified Yolov3. The task of vehicle detection was completed by creating a two-stage pipeline. MIRNet-v2 was used in the first stage of this pipeline to reduce noise and improve image contrast in low-quality challenging images. The second stage of this pipeline, used modified YOLOv3 to detect and localize vehicles in images. The hybrid model MIRYO is evaluated on two baseline datasets: the challenging MIOTCD and the high-resolution Highway dataset, and its performance is compared to that of the Yolov3, Yolov4, and Yolov5 architectures on the same dataset. MIRYO achieved an overall mAP of 76.9% on the MIOTCD dataset, while YOLOv3, YOLOv4, and YOLOv5 achieved 75.1%, 74%, and 75%, respectively, and a mAP of 94.8% on the Highway dataset, while YOLOv3, YOLOv4, and YOLOv5 achieved 94.5%, 94.9%, and 94.8%, respectively.

Keywords: Vehicle detection, MIRYO, deep learning, YOLO, MIRNet

1. Introduction

The rapid rise in vehicle and population on the road placed a significant strain on transportation infrastructure, resulting in a number of issues such as traffic jams, over-crowding, vehicle theft, rule breaking, and so on [1]. To address these issues, many technological advancements in Intelligent Transportation Systems (ITS) have occurred over the last decade. The technological advancement in sensor, computer vision, and deep learning enables substantial advantages in ITS applications including autonomous driving, traffic analysis and management, transportation networks, and so on, and vehicle detection is at the heart of these kind of smart transportation applications.

Deep learning models for vehicle detection performed better than traditional algorithms due to the accessibility of abundant data and cutting-edge hardware. Traditional vehicle detection algorithms require a significant amount of computation time and effort because they rely on hand-crafted features, and these techniques are not suitable for real-time detection. Traditional methods involve three steps: in the first step, it proposes a region of interest (ROI) using techniques such as selective search, sliding window, and so on; in the second step, features are extracted from ROIs using handcrafted-feature extraction methods such as HoG [2], SIFT-like [3], Haar-like [4], and so on; and in the third step, various classifiers such as SVM [5], AdaBoost [6], kNN [7], and so on are used to detect and classify vehicles. Deep learning-based vehicle detectors are classified into two types: 2-stage and 1-stage detectors. 2-stage detectors such as RCNN [8], FRCNN [9], Faster RCNN [10], and others involve two steps: in the first stage of the process, region proposal network (RPN) proposed ROIs, and in the second stage, proposed ROIs features are used to detect vehicles with bounding boxes. 2-stage detectors required more computation power but achieved good detection accuracy. 1-stage detectors, such as YOLO [11] and SSD [12], detect vehicles with bounding boxes in a single phase without region proposal which makes them faster than 2-stage detectors.

Though vehicle detection has received a great deal of attention recently, much more research is required to create systems that are reliable and perform well in real-world conditions. Vehicle detection in images is still plagued by issues such as occlusion, environmental and illumination conditions, a wide range of vehicle types, vehicle orientation and size, and so on [13]. The efficiency of vehicle detection in image is influenced by environmental noise and lighting conditions. To tackle these issues, proposed study developed MIRYO, a hybrid model for detecting vehicles in challenging images based on MirNet-v2 [14] and modified Yolov3. This paper's main contributions are:

- A hybrid model MIRYO for vehicle detection in challenging images that combines MIRNetv2 and a modified YOLOv3.
- To validate the effectiveness of MIRYO, two public benchmark datasets, MIOTCD [15] and Highway [16], are used.
- The hybrid model MIRYO's performance is compared to YOLOv3 [17], YOLOv4 [18], and

*E-mail address: ngupta@aitpune.edu.in
ISSN: 1791-2377
doi:10.25103/jestr.182.16

YOLOv5 [19] in terms of F1-score, precision, recall, and mAP.

The rest of this paper is structured as follows. Section 2 offers the summary of previous works. Section 3 describes, MIRYO, our hybrid model. The experimental setups employed in this work, along with the data sources and hyper-parameters, are detailed in Section 4. Section 5 contains the findings and discussion, and the paper is concluded and recommendations for future investigations are offered in Section 6.

2. Related Work

Many researchers have published various techniques for the detection of vehicles in images, but it is still a difficult task as these images suffer from various environmental noise, complex weather, and illumination conditions, all of which have a substantial impact on images, significantly reducing vehicle detection system performance. Despite the fact that DL models have been found to outperform traditional machine learning (ML) approaches in image-based vehicle detection, numerous latest studies have utilized conventional ML techniques. Xu et al. presented an improved Viola-Jones detection method for aerial imagery vehicle detection [20]. Derrouz et al. used 3D disparity map features and 2D HoG features to classify vehicles using five classifiers: SVM, kNN, Random Forest, Decision Tree, and MLP [21]. Cao et al. developed a multi-instance, weakly supervised algorithm for learning weak labels without formally labelling each object in the image. The density map obtained from the positive-regions was then used to train SVM that classify vehicles [22]. These traditional approaches to vehicle detection are not appropriate for real-time vehicle detection as they heavily rely on hand-crafted features. Deep learning approaches to vehicle detection have outshone traditional approaches and many deep learning algorithms for vehicle detection have emerged. Among these, Faster RCNN 2-stage detector, and YOLO and SDD 1-stage detectors are commonly used by the researchers for vehicle detection. Many 2-stage detectors outperform 1-stage detectors in terms of accuracy, but they are not appropriate for real-time detection due to their slow computation speed, so 1-stage detectors have gained more popularity in recent years. Faster RCNN is a 2-stage vehicle detector that uses RPN in the first stage to generate feature-maps and then uses these feature-maps to detect vehicles in the second stage. Faster RCNNs generally struggled to detect small vehicles; to address this issue, Deng et al. used a two CNN-based architecture known as coupled RCNN [23]. Tayara et al. presented FCRN [24], which not only addressed the issue of small size of vehicle in aerial images, but also overcame the challenges of detecting different types of vehicle and orientation from aerial images. Various backbone networks were also investigated by different researchers in order to enhance the efficiency of Faster RCNN for vehicle detection and classification [25-27]. SSD, on the other side, is a 1-stage detector that detects variable-size vehicles using different scales and aspect ratios. The researcher demonstrated various SSD improvements for detecting vehicles from images, including Oriented-SSD [28], Inception-SSD [29,30], FPES [31] based SSD, and FGSC-SSD [32]. SSD models, like Faster RCNN, are slower than the YOLO family, making YOLO more appropriate for real-time detection of vehicles [33]. YOLOv3, YOLOv4, and YOLOv5 used Darknet53, CSPDarknet53, and CSPDarknet53 with focus layer backbone, respectively. To solve the problem of vehicle orientation in aerial images, the authors [34] presented the YOLOv3 model, which used sloping bounding boxes, and some researchers [35,36] used the Kmeans++ algorithm for calculating anchor boxes and soft-NMS. Ni et al. [37] used depth-wise-separable convolutional blocks instead of residual blocks in the YOLOv4 backbone, and Koay et al. [38] demonstrated an improvement in tiny YOLOv4 for vehicle detection. Daniel et al. proposed an improved tiny and light-weight YOLOv5 [39] vehicle detection mechanism that used a multi-scale mechanism to detect variable-sized vehicles. Benjumea et al. modified the architecture of YOLOv5 and generated series of models to detect extra small objects [40].

It is challenging to determine which member of the YOLO family—YOLOv3, YOLOv4, and YOLOv5—is more accurate due to the variety of experimental setups used by the researchers, including the use of various datasets, dataset complexity, hyper-parameters, etc. We evaluated these three YOLO models, as well as our proposed model MIRYO, in this article using similar hyper-parameters and datasets.

3. Proposed Methodology

Numerous DL based techniques for vehicle detection have previously been used, but vehicle detection in noisy images is still considered a difficult task. Low-light, low-resolution, and other environmental noise have a significant impact on images, significantly reducing the efficiency of vehicle detection systems. This study developed MIRYO, a hybrid DL model based on MirNet-v2 and modified Yolov3 for detection of vehicle in challenging images. The detection task was completed by developing a two-stage pipeline. In the first-stage of this pipeline, pre-trained MIRNet-v2 was used to reduce noise and improve image contrast in low-quality challenging images. We used modified YOLOv3 in the second stage of this pipeline to detect and recognize vehicles in images. Our proposed model MIRYO's fundamental schematic diagram is illustrated in Figure 1.

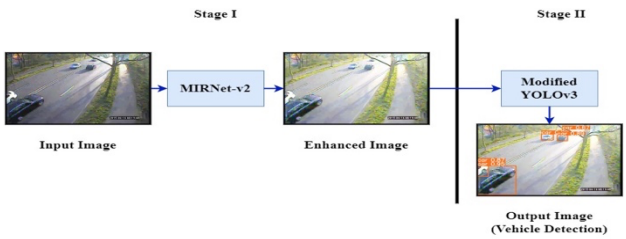

Fig. 1. Fundamental schematic diagram of hybrid model MIRYO

3.1 MIRNet-v2

Image degradations of different degree are frequently encountered as a result of the acquisition process due to the physical constraints of cameras or because of challenging lighting conditions or environmental conditions. The performance of vehicle detection from image has suffered as a result of this degraded low quality image. To address this issue, the quality of noisy images is improved in the first-stage of MIRYO by using pre-trained MIRNet-v2. MIRNetv-2 applies a convolutional layer to a low-quality noisy input image to extract low-level features, which are then passed to various recursive residual groups to extract deep features. These deep features are then passed to a convolutional layer, and the output of the convolutional layer is concatenated with

the input images to produce a restored image as a MIRNet-v2 block output.

3.2 Modified YOLOv3

Figure 2 depicts the detailed architecture of a modified YOLOv3 inspired by the YOLOv3 architecture. The backbone network is responsible for extracting features and consist of stack of bottleneck block and conv2D blocks. Bottleneck block reduce the computational cost of network and over-fitting problem. The structure of these blocks are shown in Figure 3a and 3b. Additionally, the SPPF block is added to Darknet53 backbone in modified YOLOv3, which is faster than SPP block to speed up network performance and increase accuracy in real-time vehicle detection. Figure 3c depicts the SPPF block in which input is passed through 1x1 conv2D layer and then three 5x5 max-pooling layers and the output of max-pooling layers and previous 1x1 conv2D layer is concatenated and passed to 1x1 conv2D layer to produce fixed length output. Instead of ReLU [41] activation, we used the SiLU [42] (Sigmoid-weighted Linear Unit) activation function in the cov2D block of modified YOLOv3.

The neck section of the modified YOLOv3 is identical to the original YOLOv3, which used a feature pyramid network (FPN) [43] to extract feature maps from various stages with different object scales and sizes, which are made up of many bottom-up and top-down paths. A backbone network feature map and our up-sampled features are also combined via concatenation; by doing so, we may extract more insightful semantic information. The detection head consists of three YOLO layers that makes prediction at three different scales. It predicts three boxes at each scale, so the tensor at output layer for MIOTCD is NxNx48 (size at each scale*(3*(5+number of classes)) and NxNx24 for highway dataset.

To produce new anchors as per the datasets used in this research we utilized k-means clustering algorithm that analyzes the datasets and 9 clusters and 3 scales are sorted. The new 9 clusters on MIOTCD were: (22x23), (43x32), (36x65), (74x57), (102x95), (120x169), (169x126), (219x206), (361x328), and on Highway dataset were: (8x13), (13x22), (20x 33), (30, 48), (46, 72), (64, 108), (81,167), (114,131), (131,223). Non-max suppression (NMS) has been utilized to select one bounding box for each object and removes the remaining redundant detected boxes.

YOLOv3 employs three types of losses: class loss for classification, object loss for objectness, and box loss for regression. Class loss and object loss both used binary cross-entropy (BCE) loss; the difference is that class loss only calculates the loss of positive data, whereas object loss calculates the loss of all instances. In YOLOv3, predicted box regression loss is calculated using IoU loss, but when IoU is equal to zero, IoU loss fails to optimize the model, so GIoU [44] is used in the modified YOLOv3 block of the proposed model to overcome this issue. Equations 1 and 2 are used to calculate IoU and IoU_{LOSS}, whereas equations 3 and 4 are used to calculate GIoU and $GIoU_{LOSS}$.

$$IoU = \frac{|B_p U B_{gt}|}{|B_p \cap B_{gt}|} \quad (1)$$

$$IoU_{LOSS} = 1 - IoU \quad (2)$$

$$GIoU = IoU - \frac{|C \setminus (B_p U B_{gt})|}{|C|} \quad (3)$$

$$GIoU_{LOSS} = 1 - GIoU \quad (4)$$

where B_p and B_{gt} are predicted and ground-truth boxes, C is convex shape between B_p and B_{gt}.

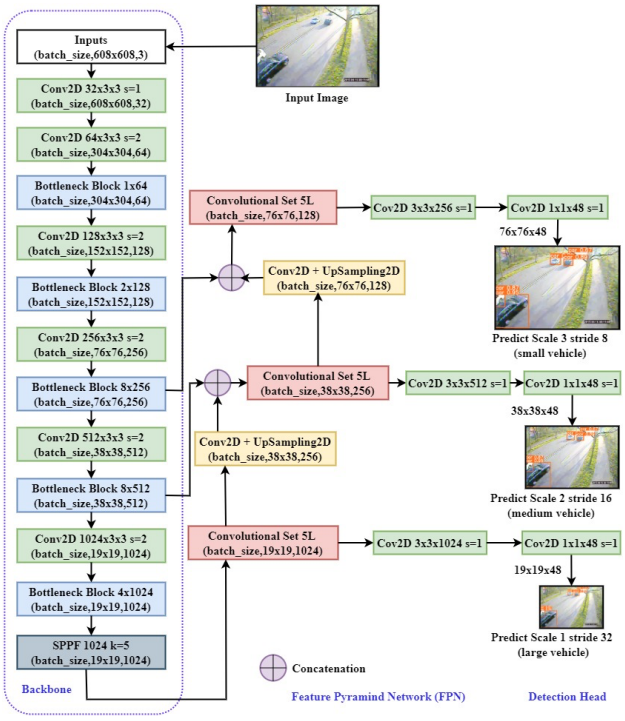

Fig. 2. Modified YOLOv3 block of MIRYO

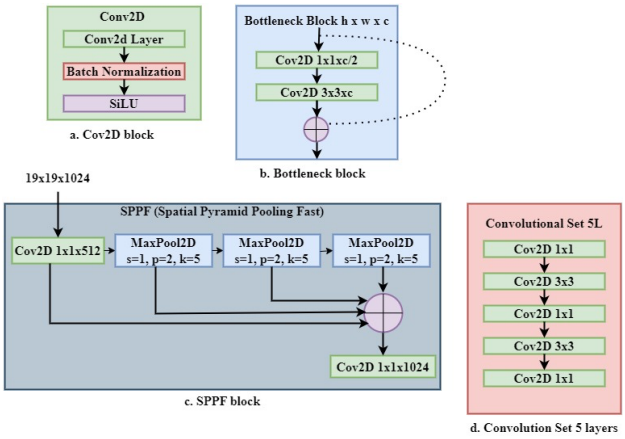

Fig. 3. a. Cov2D block, b. Bottleneck block, c. SPPF block, and d. Convolutional Set

4. Experimental Setup

4.1 Datasets

4.1.1 MIOTCD Dataset

The MIOTCD dataset, published by Luo et al. in 2018, was chosen as the first dataset for evaluating the proposed model. It includes 137,743 images collected by thousands of cameras mounted across Canada and the United States, with a variety of challenges such as complicated weather conditions, traffic density, different time periods, varying image quality, dynamic vehicle size, different lighting conditions, and so on. The dataset include eleven vehicle classes and 416,277 annotation instances of different vehicles sizes. We chose 15,000 images at random from a total of 137,743 for this experiment, with 47,945 annotation instances of various vehicle sizes, as shown in Figure 4.

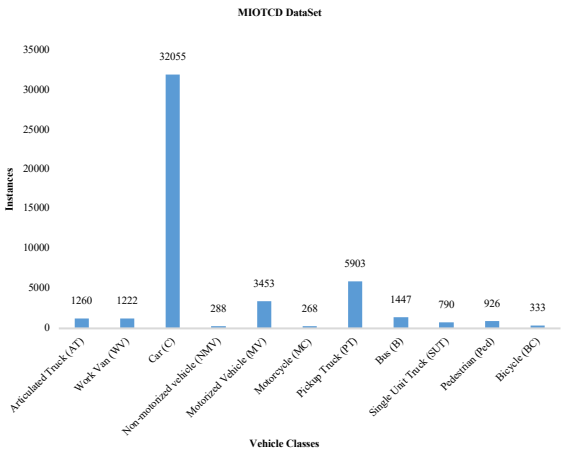

Fig. 4. Number of vehicle instances and classes in a selected dataset1 from MIOTCD

4.1.2 Highway Dataset

The Highway dataset, published by Song et al. in 2019, was chosen as the second dataset for evaluating the proposed model. It is a collection of 11,129 high-quality (1920x1080) colored images captured by 23 roadside cameras on a highway in Hangzhou, China at various times, locations, and lighting conditions. The dataset instances include three vehicle classes and 57,290 annotation instances of different vehicles sizes, as shown in Figure 5.

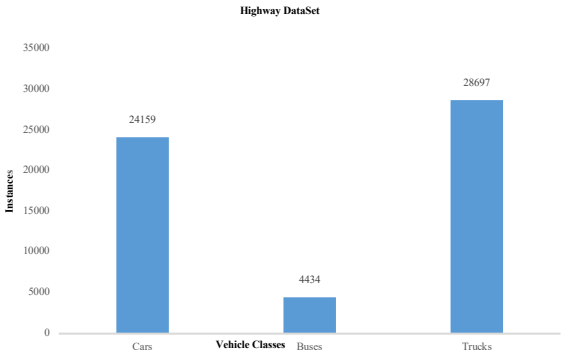

Fig. 5. Number of vehicle instances and classes in Highway dataset

4.2 Experimental Environment

On an Nvidia GTX 1080Ti GPU with 48 GB RAM running Ubuntu, we trained our models MIRYO, YOLOv3, YOLOv4, and YOLOv5. Each network is trained using the identical parameters. We set the input network size to 608x608 pixels during the training stage. Mosaic, left-right scaling and flipped transformations were used for data augmentation. Weights were optimized using the SGD algorithm, which had a momentum of 0.9, a weight decay of 0.0005, and an initial learning rate of 0.01. We trained the network for 80 epochs. We evaluated these models on two different datasets: a high-resolution highway dataset and a complicated MIOTCD dataset. We randomly selected 15,000 images from MIOTCD out of 137,743 images and 11,129 images from the highway dataset to create a train, validation, and test set with a 70%, 20%, and 10% ratio, respectively.

4.3. Evaluation Metrics

We used precision (P), recall (R), F1-score and mean average precision (mAP) [45] metrics to evaluate the MIRYO and compare it to YOLOv3, YOLOv4, and YOLOv5. P measures the model's prediction accuracy and is defined in equation 5. R measures the model's ability to recognize all ground truths and is defined in equation 6. The F1-score is calculated using the harmonic mean of P and R, which is defined in equation 7. The mAP is determined by averaging the average precision (AP) of all classes and is defined in equation 8.

$$P = \frac{TP}{TP+FP} \tag{5}$$

$$R = \frac{TP}{TP+FN} \tag{6}$$

$$F1 - score = 2 * \frac{(P*R)}{(P+R)} \tag{7}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{8}$$

where TP, FP, and FN stand for true-positive, false-positive, and false-negative, respectively.

5. Result and Discussion

We evaluated the effectiveness of proposed hybrid model MIRYO on two datasets, MIOTCD and Highway, and compared its performance with YOLOv3, YOLOv4, and YOLOv5 in terms of P, R, F1-score, and mAP. The performance of these models on two datasets is shown in Table 1 and Table 2, with the highest scores outlined in bold. For training, validation, and testing, 10500, 3000, and 1000 images were chosen from the MIOTCD dataset, respectively, and 7790, 2225, and 1114 images were chosen from the highway dataset, respectively. According to the experimental results, MIRYO achieved the highest precision of 79.7% and mAP_{50} of 76.9% on the MIOTCD dataset. On the highway dataset, MIRYO achieved the highest F1-score of 91% and mAP_{50} of 94.8%, which is extremely close to the mAP_{50} of 94.9% achieved by YOLOv4. The proposed model with 10% of dropout and CIoU loss function achieved 75.5% mAP and 94.4 % mAP on MIOTCD and the highway dataset respectively.

Also, the efficacy of MIRYO evaluated on high resolution highway dataset with inclusion four type of artificial noises Gaussian, SNP, Poisson, and speckle noise which is shown in Table 3. This investigation of noise models is necessary for the removal of noise from images, which is actually needed for better prediction results.

Table 1. Performance of vehicle detection models on MIOTCD Dataset

Models	MIOTCD Dataset (11 Vehicle classes)				Inference Time (ms)
	P	R	F1-Score	mAP_{50}	
YOLOv3	0.763	0.721	0.74	0.751	8.5
YOLOv4	0.79	**0.85**	**0.82**	0.74	14
YOLOv5	0.754	0.718	0.74	0.75	5

MIRYO without MIRNET	0.783	0.691	0.73	0.75	8.3
MIRYO with MIRNET (ours)	**0.797**	0.704	0.75	**0.769**	8.2

Table 2. Performance of vehicle detection models on Highway Dataset

Models	Highway Dataset (3 vehicle classes)				Inference Time (ms)
	P	R	F1-Score	mAP_{50}	
YOLOv3	**0.916**	0.899	**0.91**	0.945	7.5
YOLOv4	0.85	**0.94**	0.89	**0.949**	12
YOLOv5	0.907	0.899	0.90	0.948	4.4
MIRYO without MIRNET	0.907	0.899	0.90	0.946	7.2
MIRYO with MIRNET (ours)	0.914	0.898	**0.91**	0.948	7.2

Table 3. Performance of MIRYO on Highway dataset with different noise

Noise	Gaussian			SNP			Poisson			Speckle		
Class	P	R	mAP_{50}	P	R	mAP_{50}	P	R	mAP_{50}	P	R	mAP_{50}
all	0.917	0.894	0.945	0.914	0.898	0.948	0.905	0.905	0.944	0.885	0.867	0.919
truck	0.926	0.921	0.965	0.926	0.923	0.967	0.919	0.924	0.964	0.907	0.894	0.949
bus	0.911	0.89	0.933	0.906	0.893	0.938	0.899	0.901	0.93	0.874	0.853	0.905
car	0.915	0.871	0.938	0.908	0.879	0.938	0.898	0.891	0.938	0.874	0.853	0.904

The class-wise mAP_{50} and Precision-recall (PR) curve of proposed model MIRYO on MIOTCD is shown in Figure 6. MV category of MIOTCD achieved very less mAP_{50} due to its small physical appearance. The authors of MIOTCD achieved a mAP_{50} of 80.36% using YOLOv4 model assuming that if MV vehicle detected in AT, B, C, PT, SUT, or WV, then it will be treated as true-positive because MV are too small to fit into a particular category. We didn't consider them as true-positive hence achieved comparatively lower mAP_{50}.

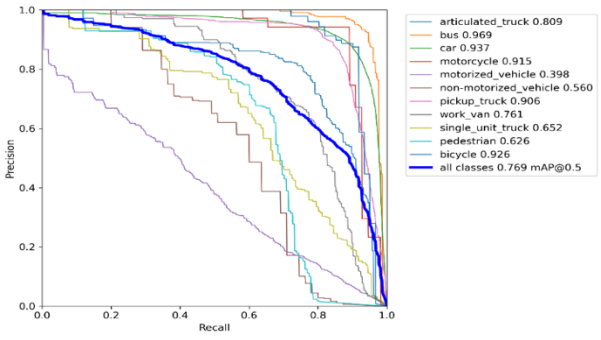

Fig. 6. PR Curve and class-wise mAP_{50} of MIRYO on MIOTCD dataset

The class-wise mAP_{50} and Precision-recall (PR) curve of proposed model MIRYO on highway dataset is shown in Figure 7. The authors of the highway dataset achieved a mAP_{50} of 87.88% using the YOLOv3 model, while we achieved 94.8%. Furthermore, the number of classes in the dataset and the quality of the dataset both have an impact on model performance. MIOTCD dataset images with eleven vehicle classes are low resolution images captured in challenging environmental conditions such as darkness, snow, and clouds, whereas highway dataset images with three vehicle classes are high resolution images. As a result, the overall mAP_{50} of the challenging MIOTCD is lower than that of the high-quality highway dataset.

The Confusion matrix in Figures 8 and Figure 9 depicts the number of correct and incorrect predictions per class made by MIRYO on the MIOTCD and Highway datasets.

The sample real-time detection of vehicle by MIRYO in MIOTCD, Highway and our private data captured by camera installed at our institute (AIT, Pune) main gate is shown in Figure 10.

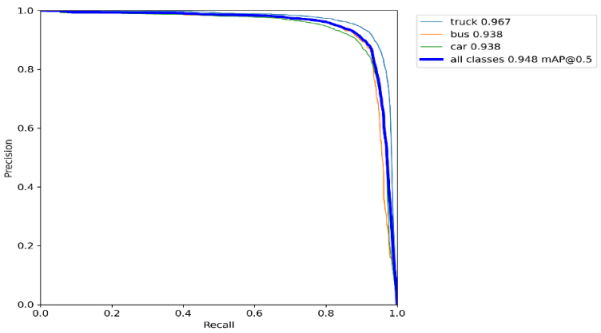

Fig. 7. PR Curve and class-wise mAP_{50} of MIRYO on Highway dataset

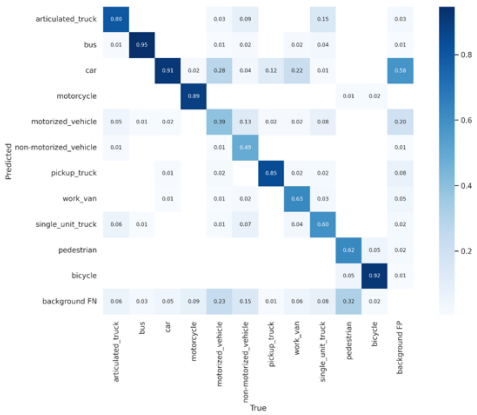

Fig. 8. Confusion Matrix of MIRYO on MIOTCD dataset

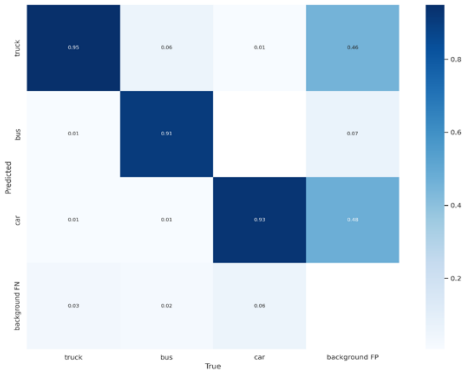

Fig. 9. Confusion Matrix of MIRYO on the Highway dataset

Fig. 10. Sample detection results on MIOTCD, Highway and Private Dataset

6. Conclusion

In this paper, we proposed MIRYO, a hybrid model that used pre-trained MIRNetv-2 and modified YOLOv3 models to detect vehicles in low and high quality images. The hybrid model MIRYO was evaluated on two baseline datasets: the complicated low-quality MIOTCD dataset and the high-resolution Highway dataset. In terms of mAP, the proposed model outperformed existing YOLOv3, YOLOv4, and YOLOv5 models. On the MIOTCD dataset, the proposed model MIRYO achieved an overall mAP of 76.9%, while YOLOv3, YOLOv4, and YOLOv5 achieved 75.1%, 74%, and 75%, respectively, and a mAP of 94.8% on the Highway dataset, while YOLOv3, YOLOv4, and YOLOv5 achieved 94.5%, 94.9%, and 94.8%, respectively. MIRYO's performance on the highway dataset was very close to that of YOLOv3, YOLOv4, and YOLOv5, but on the MOITCD dataset, MIRYO achieved a minimum 2% improvement over YOLOv3, YOLOv4, and YOLOv5. As a result, we can conclude that MIRYO, our hybrid model, detects vehicles more reliably in noisy images. The accuracy of smaller objects still needs to be improved, which we will work on in the future.

References

[1] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing Transportation Systems via Seep Searning: A Survey," *Transp. Res. Part C Emerg. Technol.*, vol. 99, pp. 144–163, Feb. 2019, doi: 10.1016/j.trc.2018.12.004.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Comp. Soc. Conf. Comp. Vision Patt. Recog. (CVPR'05)*, San Diego, CA USA: IEEE, Jun. 2005, pp. 886–893, doi: 10.1109/cvpr.2005.177.

[3] D. G. Lowe, "Object Recognition from Local Scale-invariant Features," in *Proceed. Seventh IEEE Int. Conf. Comp. Vision*, Kerkyra, Greece, Sep. 1999, pp. 1150-1157, doi: 10.1109/ICCV.1999.790410.

[4] T. Mita, T. Kaneko and O. Hori, "Joint Haar-like Features for Face Detection," in *Tenth IEEE Int. Conf. Comp. Vision (ICCV'05)*, Beijing, China, Oct. 2005, pp. 1619-1626, doi: 10.1109/ICCV.2005.129.

[5] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[6] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comp. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: doi.org/10.1006/jcss.1997.1504.

[7] J. M. Keller, M. R. Gray and J. A. Givens, "A Fuzzy K-nearest Neighbor Algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 4, pp. 580-585, July-Aug. 1985, doi: 10.1109/TSMC.1985.6313426.

[8] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conf. Comp. Vision and Patt. Recog. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.

[9] R. Girshick, "Fast R-CNN," in *2015 IEEE Int. Conf. on Comp. Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[10] 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, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conf. Comp. Vision and Patt. Recog. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Comp. Vision – ECCV 2016: 14th Eur. Conf., Amsterdam, The Netherlands, Oct., 2016, Proceedings, Part I 14*, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[13] A. Boukerche, A. J. Siddiqui, and A. Mammeri, "Automated Vehicle Detection and Classification," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 1–39, Sep. 2018, doi: 10.1145/3107614.

[14] S. W. Zamir *et al*., "Learning Enriched Features for Fast Image Restoration and Enhancement," *IEEE Trans. Patt. Anal.s and Mach. Intell.*, vol. 45, no. 2, pp. 1934-1948, Feb. 2023, doi: 10.1109/TPAMI.2022.3167175.

[15] Z. Luo, F. Branchaud-Charron, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, and P.-M. Jodoin, "MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5129-5141, Oct. 2018, doi: 10.1109/TIP.2018.2848705.

[16] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based Vehicle Detection and Counting System using Deep Learning in Highway Scenes," *Eur. Transp. Res. Rev.*, vol. 11, no. 1, Dec. 2019, doi: 10.1186/s12544-019-0390-4.

[17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv.org*, Aug. 2018. doi: abs/1804.02767.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, Apr. 2020, doi: abs/2004.10934.

[19] G. jocher et al, "YOLOv5", Ultralytics, [Online]. Available: https://github.com/ ultralytics/yolov5, [Accessed Apr. 19, 2023].

[20] Y. Xu, G. Yu, X. Wu, Y. Wang and Y. Ma, "An Enhanced Viola-Jones Vehicle Detection Method from Unmanned Aerial Vehicles Imagery," *IEEE Trans. Intell. Transp. Syst.,* vol. 18, no. 7, pp. 1845-1856, July 2017, doi: 10.1109/TITS.2016.2617202.

[21] H. Derrouz, A. Elbouziady, H. Ait Abdelali, R. Oulad Haj Thami, S. El Fkihi and F. Bourzeix, "Moroccan Video Intelligent Transport System: Vehicle Type Classification Based on Three-Dimensional and Two-Dimensional Features," *IEEE Access*, vol. 7, pp. 72528-72537, Jun. 2019, doi: 10.1109/ACCESS.2019.2920740.

[22] L. Cao, F. Luo, L. Chen, Y. Sheng, H. Wang, C. Wang, and R. Ji, "Weakly Supervised Vehicle Detection in Satellite Images via Multi-Instance Discriminative Learning," *Patt. Recogn.*, vol. 64, pp. 417–424, Apr. 2017, doi: 10.1016/j.patcog.2016.10.033.

[23] Z. Deng, H. Sun, S. Zhou, J. Zhao and H. Zou, "Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled

Region-Based Convolutional Neural Networks," in *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 8, pp. 3652-3664, Aug. 2017, doi: 10.1109/JSTARS.2017.2694890.

[24] H. Tayara, K. Gil Soo and K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," *IEEE Access*, vol. 6, pp. 2220-2230, Feb. 2018, doi: 10.1109/ACCESS.2017.2782260

[25] Q. Tan, J. Ling, J. Hu, X. Qin and J. Hu, "Vehicle Detection in High Resolution Satellite Remote Sensing Images Based on Deep Learning," *IEEE Access*, vol. 8, pp. 153394-153402, Aug. 2020, doi: 10.1109/ACCESS.2020.3017894.

[26] G. Yin, M. Yu, M. Wang, Y. Hu, and Y. Zhang, "Research on Highway Vehicle Detection based on Faster R-CNN and Domain Adaptation," *Appl. Intell*, vol. 52, no. 4, pp. 3483–3498, Jul. 2021, doi: 10.1007/s10489-021-02552-7.

[27] Y. Chen and W. Hu, "A Video-Based Method with Strong-Robustness for Vehicle Detection and Classification Based on Static Appearance Features and Motion Features," *IEEE Access*, vol. 9, pp. 13083-13098, Jan. 2021, doi: 10.1109/ACCESS.2021.3051659.

[28] T. Tang, S. Zhou, Z. Deng, L. Lei, and H. Zou, "Arbitrary-Oriented Vehicle Detection in Aerial Imagery with Single Convolutional Neural Networks," *Remote Sens.*, vol. 9, no. 11, Nov. 2017, Art, no. 1170, doi: 10.3390/rs9111170.

[29] W. Chen, Y. Qiao, and Y. Li, "Inception-SSD: An Improved Single Shot Detector for Vehicle Detection," *J. Ambient Intell. Hum. Comput.*, vol. 13, pp. 5047-5053, Jun. 2020, doi: 10.1007/s12652-020-02085-w.

[30] P. M. Harikrishnan, A. Thomas, V. P. Gopi, P. Palanisamy, and K. A. Wahid, "Inception Single Shot Multi-Box Detector with Affinity Propagation Clustering and their Application in Multi-Class Vehicle Counting," *Appl. Intell.*, vol. 51, no. 7, pp. 4714–4729, Jan. 2021, doi: 10.1007/s10489-020-02127-y.

[31] M. Zhao, Y. Zhong, D. Sun, and Y. Chen, "Accurate and Efficient Vehicle Detection Framework based on SSD Algorithm," *IET Image Proc.*, vol. 15, no. 13, pp. 3094–3104, Jun. 2021, doi: 10.1049/ipr2.12297.

[32] M.-H. Sheu, S. M. S. Morsalin, J.-X. Zheng, S.-C. Hsia, C.-J. Lin, and C.-Y. Chang, "FGSC: Fuzzy Guided Scale Choice SSD Model for Edge AI Design on Real-Time Vehicle Detection and Class Counting," *Sensors*, vol. 21, no. 21, Nov. 2021, Art. no. 7399, doi: 10.3390/s21217399.

[33] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs," *Sensors*, vol. 22, no. 2, Jan. 2022, Art. no. 464, doi: 10.3390/s22020464.

[34] X. Zhang and X. Zhu, "An Efficient and Scene-Adaptive Algorithm for Vehicle Detection in Aerial Images Using an Improved YOLOv3 Framework," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 11 ,Oct. 2019, Art. no. 483, doi: 10.3390/ijgi8110483.

[35] X. Luo, X. Tian, H. Zhang, W. Hou, G. Leng, W. Xu, H. Jia, X. He, M. Wang, and J. Zhang, "Fast Automatic Vehicle Detection in UAV Images using Convolutional Neural Networks," *Remote Sens.*, vol. 12, no. 12, Jun. 2020, Art. no. 1994, doi: 10.3390/rs12121994.

[36] S. Jamiya and E. Rani, "LittleYOLO-SPP: A Delicate Real-time Vehicle Detection Algorithm," *Optik*, vol. 225, Jan. 2021, Art. no. 165818, doi: 10.1016/j.ijleo.2020.165818.

[37] Z. Ni, T. Liu, K. Li, Y. Bai, and Z. Zhu, "Real-time Vehicle Detection and Computer Intelligent Recognition through Improved YOLOv4," *J. Phys.: Conf. Ser.*, vol. 2083, no. 4, Nov. 2021, Art. no. 042006, doi: 10.1088/1742-6596/2083/4/042006.

[38] H. V. Koay, J. H. Chuah, C.-O. Chow, Y.-L. Chang, and K. K. Yong, "YOLO-RTUAV: Towards Real-Time Vehicle Detection through Aerial Images with Low-Cost Edge Devices," *Remote Sens.*, vol. 13, no. 21, pp. 4196–4196, Oct. 2021, doi: 10.3390/rs13214196.

[39] D. Padilla Carrasco, H. A. Rashwan, M. Á. García and D. Puig, "T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks," *IEEE Access*, vol. 11, pp. 22430-22440, Mar. 2023, doi: 10.1109/ACCESS.2021.3137638.

[40] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving Small Object Detection in YOLOv5 for Autonomous Vehicles," *arXiv:2112.11798 [cs]*, Dec. 2021, Available: abs/2112.11798

[41] A. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv.org*, Aug. 2018. doi: abs/1803.08375.

[42] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning," *Neural Networks*, vol. 107, pp. 3–11, Nov. 2018, doi: 10.1016/j.neunet.2017.12.012.

[43] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conf. Comp. Vision and Patt. Recog. (CVPR)*, Honolulu, HI, USA, July 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.

[44] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *2019 IEEE/CVF Conf. Comp. Vision Patt. Recog. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 658-666, doi: 10.1109/CVPR.2019.00075.

[45] A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review," *Remote Sens.*, vol. 13, no. 13, Jun. 2021, Art. no. 2450, doi: 10.3390/rs13132450.