BRAIN. Broad Research in Artificial Intelligence and Neuroscience
e-ISSN: 2067-3957 | p-ISSN: 2068-0473
Covered in: Web of Science (ESCI); EBSCO; JERIH PLUS (hkdir.no); IndexCopernicus; Google Scholar; SHERPA/RoMEO; ArticleReach Direct; WorldCat; CrossRef; Peeref; Bridge of Knowledge (mostwiedzy.pl); abcdindex.com; Editage; Ingenta Connect Publication; OALib; scite.ai; Scholar9; Scientific and Technical Information Portal; FID Move; ADVANCED SCIENCES INDEX (European Science Evaluation Centre, neredatatics.org); ivySCI; exaly.com; Journal Selector Tool (letpub.com); Citefactor.org; fatcat!; ZDB catalogue; Catalogue SUDOC (abes.fr); OpenAlex; Wikidata; The ISSN Portal; Socolar; KVK-Volltitel (kit.edu)
2025, Volume 16, Issue 4, pages: 169-184
Submitted: June 24th, 2025 | *Accepted for publication:* October 12th, 2025

CARL-ODD: A Vision Benchmark Dataset of Asia for On-Road Vehicle Detection and Recognition

Sarmad Shafique*
Department of Computer Science, National University of Modern Languages, Mirpur AJK Campus, Pakistan.
Sarmad.shafique@numl.edu.pk
https://orcid.org/0000-0002-7213-6447

Samia Abid
Control, Automotive and Robotics Lab affiliated lab of National Centre of Robotics and Automation (NCRA HEC Pakistan).
samia.shah355@gmail.com
https://orcid.org/0000-0003-1032-1724

Faisal Riaz
Control, Automotive and Robotics Lab affiliated lab of National Centre of Robotics and Automation (NCRA HEC Pakistan), and with the Department of Computer Science and Information Technology, Mirpur University of Science and Technology (MUST), Mirpur AJK, 10250, Pakistan.
Faisal.riaz@must.edu.pk
https://orcid.org/0000-0003-2005-3815

Bilal Ahmed
Control, Automotive and Robotics Lab affiliated lab of National Centre of Robotics and Automation (NCRA HEC Pakistan).
chbilalahmed0059@gmail.com
https://orcid.org/0000-0003-2396-6820

Mubeen Javaid Khan
Control, Automotive and Robotics Lab affiliated lab of National Centre of Robotics and Automation (NCRA HEC Pakistan). Mubeenkhan270396@gmail.com
https://orcid.org/0009-0002-5039-8734

Usama Younis
Department of Computer Science, National University of Modern Languages, Mirpur AJK Campus, Pakistan.
usama.younis@numl.edu.pk, https://orcid.org/0009-0004-9067-4195

Ameer Hamza
Department of Computer Science, National University of Modern Languages, Mirpur AJK Campus, Pakistan.
ameerhamza@numl.edu.pk, https://orcid.org/0000-0002-3148-674X

**Corresponding Author*

Abstract: *The advancement of self-driving vehicles' ability to perceive their environment is highly dependent on computer vision-based object detection and recognition. Over the past few decades, many substantial object detection and recognition datasets have been proposed, including the Karlsruhe Institute of Technology and Toyota Technological (KITTI) dataset, the Beijing Institute of Technology (BIT) vehicle dataset, etc. However, these datasets consist of smooth and organised urban road traffic of Western nations, ignoring the congested and disordered traffic of Asian nations (i.e., Pakistan, India, etc.). To overcome the above-mentioned issues, a substantial dataset named Control Automotive and Robotics Lab- Object Detection Dataset (CARL-ODD) has been proposed for self-driving vehicles to navigate safely while having a detailed perception of their surroundings in an Asian environment. CARL-ODD dataset was collected from over eighty cities of Pakistan containing highways, motorways, and congested as well as diverse traffic scenarios of urban, rural, and hilly areas. The use of both mono and stereo-vision-based driving videos in the dataset provided a considerable advantage over the state-of-the-art datasets. The collected data were then labelled, and 20,000 representative images were selected for the perception module. Moreover, to establish a baseline for CARL-ODD, we have also compared the various pre-trained single-stage and multistage object detection and recognition algorithms with the dataset. These pre-trained object detection algorithms, such as YOLO and Faster R-CNN, have been implemented to detect and classify various on-road vehicles present in the Asian environment. This approach can save time and resources compared to training a model from scratch. The major findings as a result of the analysis showed that the CARL-ODD dataset demonstrated strong performance in detecting and recognising surrounding vehicles, while achieving 91% average precision for single-stage detectors and 93% average precision for multistage object detectors. This work outperforms the state-of-the-art datasets in terms of vehicle classes and diverse traffic scenarios.*

Keywords: *autonomous vehicles; dataset; intelligent perception; vision-benchmark; CARL-ODD; Faster R-CNN.*

How to cite: Shafique, S., Abid, S., Riaz, F., Ahmed, B., Khan, M. J., Younis, U., & Hamza, A. (2025). CARL-ODD: A vision benchmark dataset of Asia for on-road vehicle detection and recognition. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, *16*(4), 169-184. https://doi.org/10.70594/brain/16.4/10

1. Introduction

Traffic congestion is the most excruciating experience for people living in the modern cities of Asia. Every day, thousands of people die in Asia because of road accidents due to congested and unorganised traffic (World Health Organization, 2018).

This requires the development of intelligent solutions to avoid such ravaging situations. Therefore, autonomous driving research provides an imperative premise for the perception of the road environment, which has become the research focus in intelligent vehicles.

Autonomous driving research requires in-depth and rigorous analysis of scene understanding. For this purpose, it is paramount to collect highly accurate data to feed the perception module that helps detect and identify on-road objects. To date, several benchmark datasets, such as COCO (Lin et al., 2014), PASCAL VOC (Everingham et al., 2010) and KITTI (Geiger et al., 2013), have been introduced for object detection techniques. These datasets have been applied in autonomous driving for the detection of vehicles (Zhou et al., 2025; Vishwakarma & Jain, 2024), pedestrians (Amine et al., 2022; Zhang et al., 2018), road signs (Mohammad, Szalay, & Török, 2021; Saxena et al., 2024), traffic lights (Lin et al., 2022; Han et al., 2024), etc. Despite the diversity of existing datasets in the literature, none of them provides the dataset of Asia's local vehicles and pedestrians.

For the perception of a self-driving vehicle in an Asian environment, we have proposed a vision benchmark suite, "CARL-ODD", to detect objects around the ego-vehicle. The proposed dataset exhibits diverse Asian traffic scenarios collected during peak and off-peak traffic hours. CARL-ODD contains four object categories, including three classes of vehicles and pedestrians. This dataset encompasses 2-wheeled, 3-wheeled, and 4-wheeled vehicles These vehicles include motorbikes, rickshaws, minibuses, cars, trucks, and vans.. Moreover, the features and the dressing culture of Asian pedestrians are different as compared to western people. Thus, the perception module is susceptible to failure if trained on the existing dataset. Hence, our proposed dataset will include images of local vehicles, objects, traffic lights, road signs, and pedestrians in Asia. To overcome the problems in existing datasets, our contributions are as follows:

For the advancement of the perception module of self-driving vehicles, this article proposes a substantial dataset named CARL-ODD, which helps in detecting and recognising surrounding on-road vehicles and objects. CARL-ODD consists of driving videos covering over 1,100 km collected from more than 80 cities and includes congested traffic routes, urban and rural highways, hilly terrain, and motorways. This dataset consists of almost 20,000 labelled images with approximately 55,000 labels and is presented in two types of annotation formats which include: YOLO-based annotations (in .txt format) and PASCAL VOC-based annotations (in .xml format). Our proposed dataset includes a variety of class categories, including two-wheeler (motorbikes), three-wheeler (rickshaws), and four-plus-wheeler (cars, buses, and trucks).. Additionally, one-stage and two-stage object detectors are assessed using the proposed dataset to show their accuracy and effectiveness. Finally, a thorough comparison of our proposed benchmark suite with the most recent object detection datasets for self-driving cars demonstrates its diversity and efficacy. Table 1 includes the abbreviation and their meanings.

Table 1. Abbreviation and Meaning

Abbreviation	Meaning
COCO	Common Objects in Context
FPN	Feature Pyramid Network
SGD	Stochastic Gradient Decent
CNN	Convolutional Neural Network
VOC	Visual Object Classes
SPP	Spatial Pyramid Pooling
CSP	Cross Stage Partial
ODD	Object Detection Dataset
SSD	Single Shot Detector

The rest of the paper is organised as follows: Section II presents the literature on existing datasets; the proposed methodology is given in Section III; Section IV provides extensive results and experiments; the detailed results and comparison are given in Section V, while Section VI includes the conclusion and future work.

2. Related Work

In terms of autonomous driving, deep learning technology has achieved significant advancements (Muzammul & Li, 2025). Traffic safety is greatly impacted by vehicle detection, which is why it has received substantial attention from both academia and industry. Autonomous driving is a critical task as it requires high precision and real-time response to its surrounding environment. For this purpose, high-quality data must be collected from various scenarios to achieve promising results for the perception module.

Various deep-learning-based vehicle identification algorithms have been proposed in recent studies. These deep learning methods can be categorised into two categories: region-based and regression-based methods (Berwo et al., 2023). The former methods help to generate region candidate boxes and then extract and classify these boxes using deep learning techniques (Chen, Qiao, & Li, 2020). Currently, the main region-based methods are R-CNN (Yeh, Lin, & Chang, 2021) and Fast R-CNN (Girshick et al., 2014). However, these methods are computationally complex for real-time object detection. In comparison, the latter techniques consider object detection as a single feed-forward neural network problem. Regression-based methods are computationally fast and demonstrate high accuracy in object detection, including Darknet-based You Only Look Once (YOLO) (Ren et al., 2015), YOLOv2 (Redmon et al., 2016), YOLOv3 (Zhao & Li, 2020), and single shot detector (SSD) algorithm.

The above-mentioned techniques require high-quality data to give satisfactory results. Dataset are essential to train systems to perform their required role. For this purpose, numerous vehicle datasets have been proposed for self-driving research. One of the most well-known datasets for self-driving technology is collected by Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI). The image-based KITTI uses RGB and greyscale stereo vision cameras to cover various driving circumstances. This benchmark dataset includes over 7,400 training images and over 7,500 test images. This dataset only consists of cars, bicycles, and pedestrians among the classes of on-road objects. Another dataset introducedin 2005 and expanded in 2012 named PASCAL Visual Object Classes (VOC) constitutes 20 categories of objects. Researchers have extensively used this dataset as a benchmark to evaluate object detection, semantic segmentation, and classification-based problems. This dataset consists of three sets of images, namely, training, validation, and testing images, and comprises approximately 10,000 images.

In contrast, the Nepalese dataset (Bhatti et al., 2021) was collected as part of a final-year project. They captured 30 videos (each of length 4 minutes) of different traffic scenarios, and the images of interest were cropped from the frame. This dataset contains a total of 4,800 images consisting of two and four-wheeler vehicles. However, the researchers did not consider collecting data on pedestrians on the road.

Moreover, to evaluate the model developed by an image processing group, a new dataset was collected from video sequences using a camera attached to the vehicle's front end (Li et al., 2022). The dataset is named Grupo de Tratamiento de Imágenes (GTI) and comprises 3,425 rear-end views of on-road vehicles. In comparison, GTI contains 3,900 images of the road without any vehicle on it. However, the dataset contains only the rear-end view of four-wheeled vehicles and does not exhibit a variety of vehicles that are required for efficient training of the perception module.

In 2015, the Beijing Institute of Technology (BIT) collected a vehicle model recognition dataset (Dong et al., 2014). The BIT dataset consists of 9,850 images of four-wheeled vehicles. Two cameras captured images at various times, along with the positions of the vehicles. They collected

images under varying illumination conditions, viewpoints, scales, and vehicle surface colours. It has been observed that in the captured images, the bottom part of the vehicle is not recorded; images may be due to the size of the vehicle or capturing delay. They considered six vehicle classes: buses, microbuses, minivans, sedans, SUVs, and trucks, while ignoring pedestrians and other types of on-road vehicles.

The datasets discussed by Everingham et al. (2010), Geiger et al. (2013), Bhatti et al. (2021), and Dong et al. (2014) do not consider all three main classes of vehicles, -two-wheelers, three-wheelers, and four-wheelers altogether, or the images of pedestrians have not been taken into account. Besides, the amount of data collected is insufficient to train the deep learning models to prove their applicability in this susceptible application. Moreover,these datasets capture only the front or rear-end view of the vehicle's position, which may affect the overall efficiency of the proposed deep-learning model. Hence, despite the diversity of existing datasets in the literature, none of them provides the dataset of Asia's local vehicles and pedestrians.

3. Methodology of the Proposed Dataset

Currently, there is no public dataset available for object detection that encompasses Asia's local vehicles and pedestrians. Therefore, in this article,we present Asia's only stereo-vision dataset of different types of vehicles and pedestrians for use in autonomous driving research. Stereo vision helps to perceive objects based on the principle of the human eye, and it computes the depth based on the binocular disparity between the images of an object acquired from the left and suitable cameras attached to the front dashboard of our vehicle. The features of the images acquired at a time 't' are matched by examining the relative positions of objects in the visual scene. Our proposed methodology is illustrated in Figure 1.

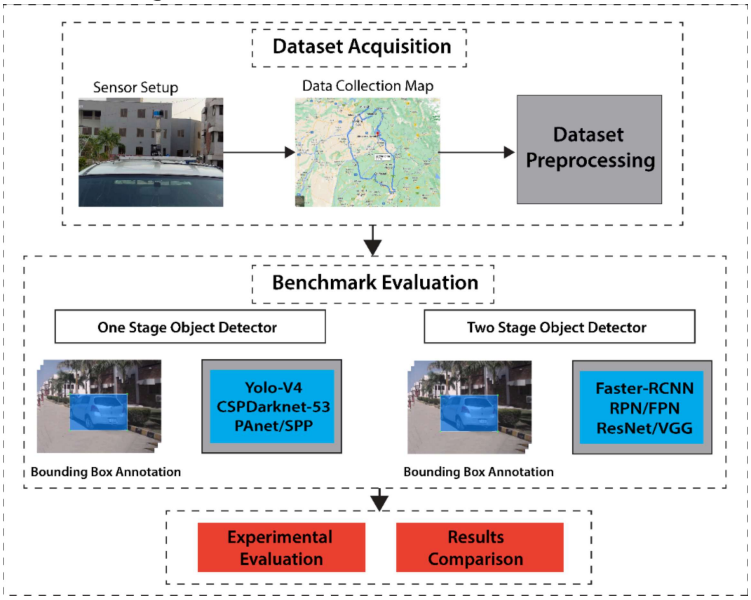

Figure 1. Proposed methodology

3.1. Data Acquisition

For the collection of robust and high-quality data, this paper utilised high-resolution cameras equipped on our vehicle while covering 360-degree views of the environment. To cover the front view, three 12-megapixel, 2K resolution cameras were equipped on a flat base fitted to the vehicle dashboard (as shown in Figure 2). These cameras were configured to capture both symmetrical stereo and monocular frames while minimising vibration.

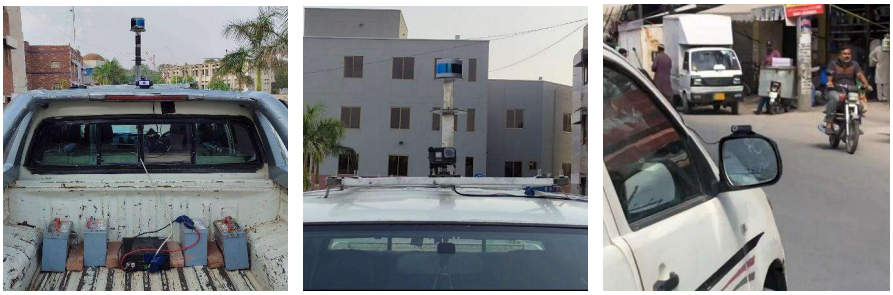

Figure 2. A glimpse of the camera and power backup setup on the data collection vehicle

In many Asian countries, most vehicles are Right-Hand Drive (RHD) compared to the European standard, which is Left Hand Drive (LHD). Therefore, a 5-megapixel HD- resolution camera has been placed on the driver-side mirror to capture environmental perception while driving an RHD vehicle. Moreover, a 5-megapixel HD+ resolution camera was placed on the rear end of the vehicle to collect the data from trailing vehicles. To ensure an uninterrupted power source for the cameras and processing machine, a power backup of four 12-Volt batteries and a UPS plus inverter was installed to provide power to the equipment mentioned above.

3.2. Road Map for Video Collection

In the literature, it can be seen that existing datasets only cover urban road traffic scenarios with limited diversity, including congested traffic scenarios and different light conditions. Moreover, three-wheeler vehicles, extensively used in Asian countries such as Pakistan, India, Bangladesh. To overcome these limitations in a state-of-the-art dataset, on-road visual traffic data of more than 1,100 KM covering almost 80 cities of Asia (Pakistan) has been collected as shown in Figure 3. During data acquisition, our primary focus was on the following vehicle types including two-wheelers (motorbikes), three-wheelers (rickshaws), and four-plus wheelers (cars, buses, and trucks). Moreover, the dataset covers rural, urban, hilly, and national highways, unpaved roads, and congested traffic scenarios. These data were collected from different types of cities, including big cities, metro cities, medium and small towns. Samples from extracted image frames of additional road traffic are depicted in Figure 4.

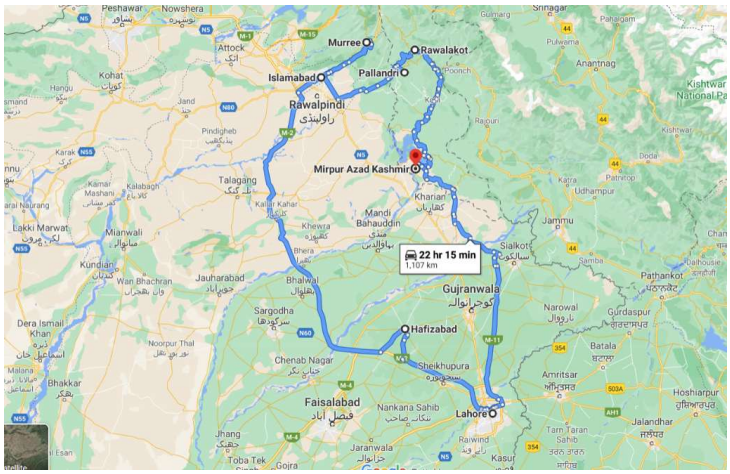

Figure 3. Google road map of 1100-KM captured in the proposed dataset, including 80+ cities of three provinces of Pakistan

3.3. Data Preprocessing and Selection

After dataset collection, image frames were extracted from the collected videos using widely used tools (as shown in Figure 4). While extracting frames from the video sequences, we carefully selected every third-frame from the video (decimate by 3) to avoid any repetition of frames in the dataset.

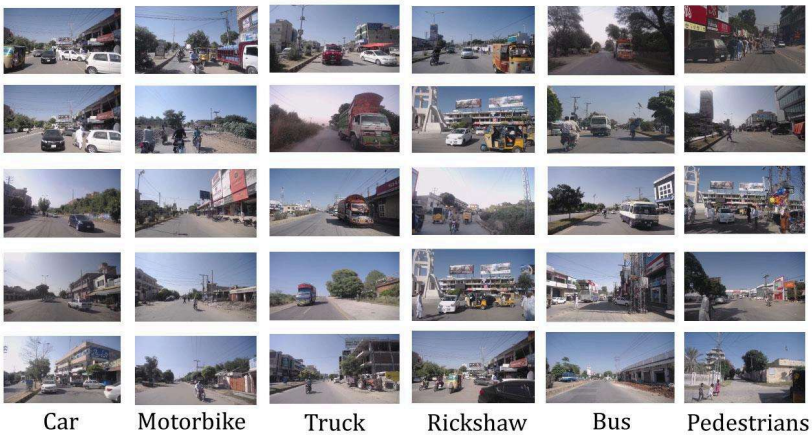

Figure 4. Example frames from the dataset demonstrating diverse classes of vehicles

Moreover, images with no objects were identified and removed from the final dataset. These extracted frames were resized to the standard High Definition (HD) resolution of 1280 x 720 pixels, with 24-bit colour depth with an RGB colour scheme. The extracted frames exhibited variations in brightness and blur due to lighting and camera errors. Therefore, data preprocessing involved removing noisy, blurry, and corrupted frames extracted from the video sequences. For this purpose, multiple data preprocessing techniques were applied, including contrast enhancement, smoothing, histogram equalisation, to improve image quality and correct brightness issues.

3.4. Data Annotation

Our CARL-ODD dataset comprises 20,000 images, which required manual annotation by trained human resources. For this purpose, a team of volunteer undergraduate students was selected and trained by experts in the Control, Automotive, and Robotics Lab (CARL) to achieve maximum performance during annotation. They were trained for one week before deploying for dataset annotation. These preprocessed images were labelled using an advanced image annotation tool named Label Image. These images are labelled in multiple formats, including YOLO and PASCAL VOC, to support single-stage and multistage object detectors. By drawing bounding boxes around the key elements in the scene, we manually annotated and classified six categories of road users.

Bounding box annotation samples are shown in Figure 5. Our proposed CARL-ODD dataset contains approximately 55,000 labels in total, as shown in Figure 6.

Figure 5. Samples of 2D annotation on the proposed dataset

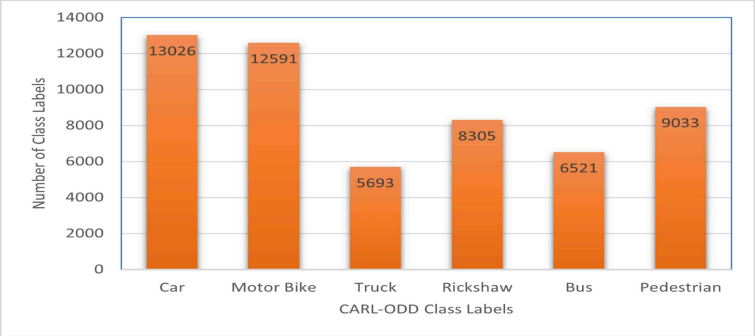

Figure 6. Total number of Labels for each class in our proposed dataset

3.5. Object Detection

One of the robust fields of computer vision is 2D objection detection, which helps locate different objects of interest in a given image or video. Object detection is considered more advanced than classification because it not only classifies objects , but also locates the position of multiple objects in an image or video sequence. Object detectors usually predict the bounding box for each object, followed by the classification score for the predicted object. Nevertheless, detectors can also predict multiple bounding boxes for each object,which are later filtered using techniques such as non-maximum suppression. Object detectors are usually categorised into two types: one-stage and two-stage. These object detectors differ from each other in their detection approach. To examine the proposed dataset, both one-stage and two-stage detectors were employed to prove its diversity.

3.6. One-Stage Object Detector

In recent years, one-stage detectors have progressed rapidly with sufficient *upgrades* in their algorithms. Due to their high inference speed, one-stage detectors are ideal for real-time applications such as perception modules of self-driving vehicles, indoor, and outdoor robot navigation, etc. In this work, we have deployed the famous and latest YOLOv4object detector to evaluate our proposed benchmark dataset. Compared to its predecessors, YOLOv4 has better inference speed and baseline accuracy.

Moreover, it is the fastest in terms of frame per second and has higher mean average precision as compared to other available object detection algorithms. To extract the different features required to detect a specific object from the images, CSP-Darknet-53 has been deployed as a backbone network for the YOLOv4, which is formed by concatenating CSPnet with the Darknet-53 network. CSP-Darknet-53 achieves higher object detection accuracy than state-of-the-art feature extractors such as ResNet, which serves as the backbone in two-stage detectors.

This fifty-three-layer ResNet design comprises multiple convolutional layers, each followed by batch normalisation and leaky ReLU layers. These layers are organised into blocks, with multiple residual blocks at the lower levels of the network to mitigate the gradient vanishing problem in deep neural networks. The pooling layer and the fully connected layer are excluded from the backbone, and the stride of the convolutional kernels is set to 2 to prevent overfitting by achieving network down--sampling. The proposed architecture is shown in Figure 7.

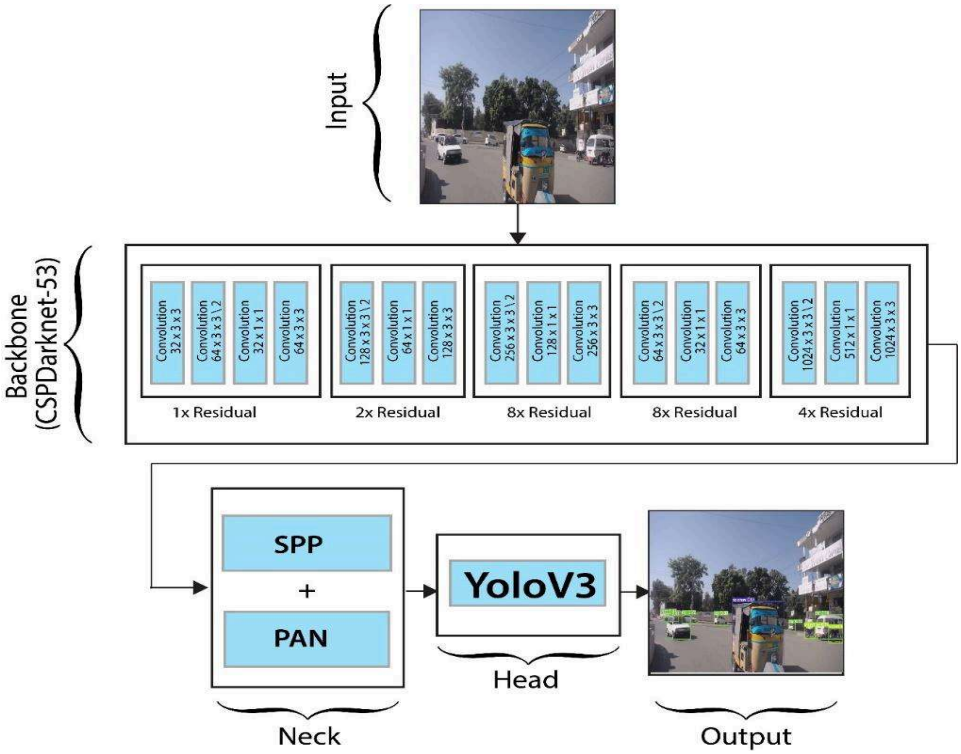

Figure 7. Proposed architecture for YOLO-based one-stage object detector

In addition to the backbone, SPP was integrated as a neck attached to the CSPDarknet-53 architecture, where the last pooling layer was replaced with a spatial pyramid pooling layer. The SPP block can generate fixed-size outputs even when given variable-sized inputs. Moreover, the SPP block can also produce similar length pooling features by taking different dimensions of the same image as input to the block. One of the main advantages of integrating the SPP block with the YOLO backbone is that it does not alter the overall network architecture, as it simply replaces the original pooling layer with the spatial pyramid pooling layer.

In this article, PANet was deployed instead of FPN, which was used in the neck of the previous YOLO version, YOLOv3. When an image is forward propagated through the neural network, the complexity of the feature vector increases while the spatial resolution decreases, resulting in lower detection and recognition accuracy. However, FPN in YOLOv3 tried to reduce this limitation by utilising a top-down approach while extracting and combining the features required for accurately locating objects in the image. However, it still lacks efficiency as image size increases, because spatial information must be propagated through hundreds of layers, which increases the time complexity. While in PANET, a supplemental enhanced bottom-up path is taken with the top-down path, which aims to reduce the information flow from the lower to the high-level layers. It helps intensify the feature pyramid and precisely localises the features present at low levels to intensify the entire feature level. Moreover, an adaptive feature pool was added in PANet to connect the feature pyramid with all feature extraction layers, enabling the sharing of valuable information via shortcut paths.

3.7. Two-Stage Object Detector

Two-stage object detectors are beneficial when high accuracy is a crucial factor compared to the inference speed, for example, in bioinformatics applications and text recognition. Therefore, to demonstrate the diversity of our proposed dataset, this article employed a widely recognised two-stage object detector called Faster Region-Based Convolutional Neural Network (FasterR-CNN). This network architecture consists of three neural networks: the first is the Feature Extraction Network, followed by Region Proposal Network, and Detection Network. The proposed architecture is shown in Figure 8.

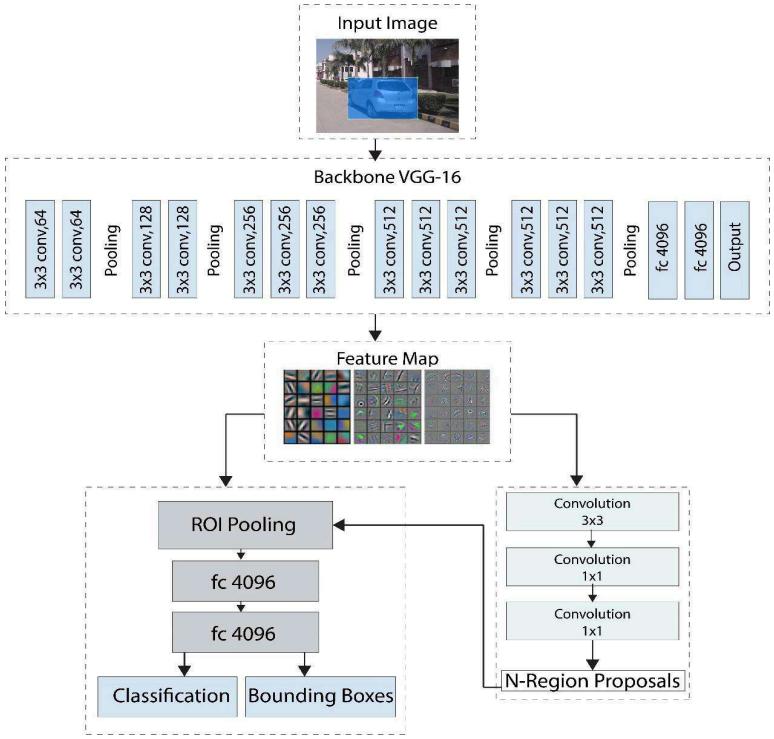

Figure 8. Proposed architecture for Faster R-CNN based two-stage object detector

For feature extraction, a 16-layer pre-trained VGG-16 network was utilised as the backbone of our network architecture , with some of its top layers. VGG-16 is a widely used convolutional neural network due to its more straightforward but powerful network architecture that is suitable for extracting generic features, which helps in identifying multiple objects. This network was employed to extract the good features from the image while maintaining the original shape and pattern.

After extracting multiple features, the Region Proposal Network (RPN), comprising three convolutional layers, was used to generate regions of interest (bounding boxes) with a high probability of containing an object. Subsequently, these features and bounding boxes were provided as input to the detection network, which consists of fully connected layers that detect and classify the objects in the input image.

4. Experiments and Results

Different state-of-the-art deep learning models including one-stage and two-stage detectors are applied to our proposed dataset to evaluate the performance of the benchmark .These models were trained on a powerful deep learning machine rigged with an Intel Core i9-9900K processor 32 GB DDR5 RAM, and NVIDIA RTX 2080Ti 11GB GPU. Multiple Python-based deep learning frameworks were employed for training these models based on the nature of the training.

4.1. One-Stage Detector

In this work, the YOLOv4 object detector was employed to evaluate our proposed benchmark dataset. A PyTorch-based setup was employed, with model training performed on GPU. Images were preprocessed by resizing to 608 × 608 pixels, to achieve training efficiency with less resource overhead. The dataset was split in the ratio of 70:20:10, with 70% of the data used for training, 20% for testing, and 10% for validation. For training, this study employed transfer learning, where YOLOv4 was initialised with pre-trained COCO weights to acquire low-level features. This can expedite the training process compared with training from scratch. While selecting the training options, we fixed the batch size to 5 per our GPU memory. An adaptive learning rate was employed to maximise performance in terms of speed and accuracy. We trained our network for 100 epochs, with approximately 4,000 iterations per epoch, given the dataset size

(20,000 images) and the chosen batch size of 5. This training process took around 18 hours to complete. The result performance achieved after training are depicted in Figure 9.

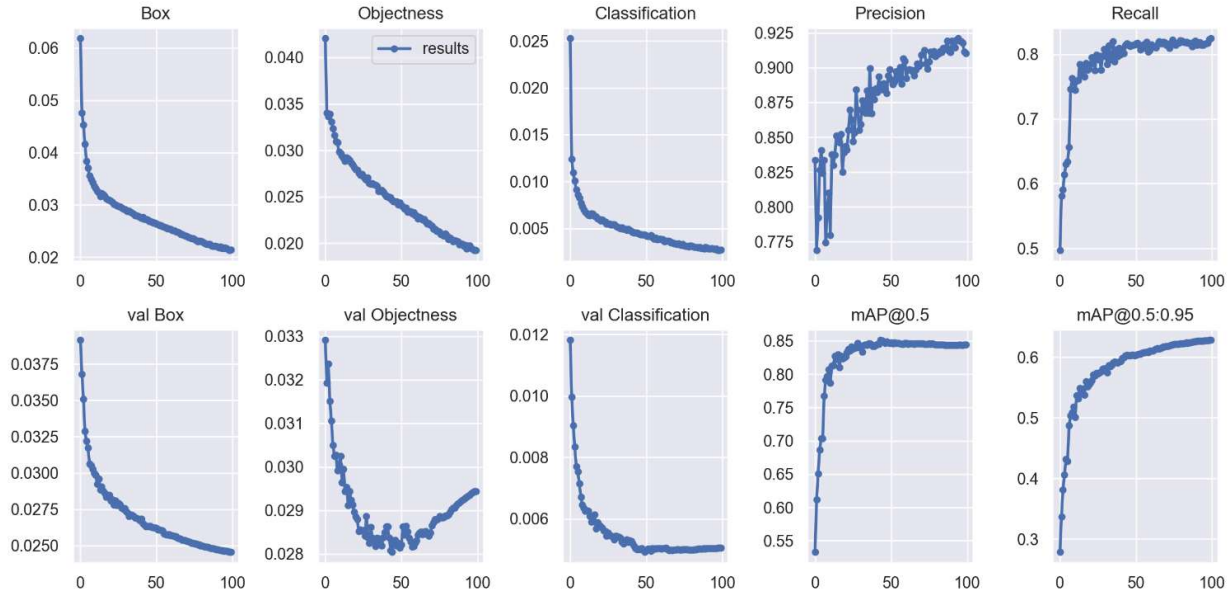

Figure 9. Training results of One Stage Detector on our proposed dataset

The performance graph as shown in Figure 10 demonstrates that the one-stage detector has shown decent performance on our dataset. This network attained an average precision of 0.91 and recall of 0.84, which shows the robustness and diversity of our proposed data collection. n the benchmark data, the deployed single-stage object detector achieved a mean average precision of 84.40% (Figure 8), demonstrating satisfactory performance for use in machine learning applications where real-time detection is critical. Some of the predicted samples of this network are also shown in Figure 11.

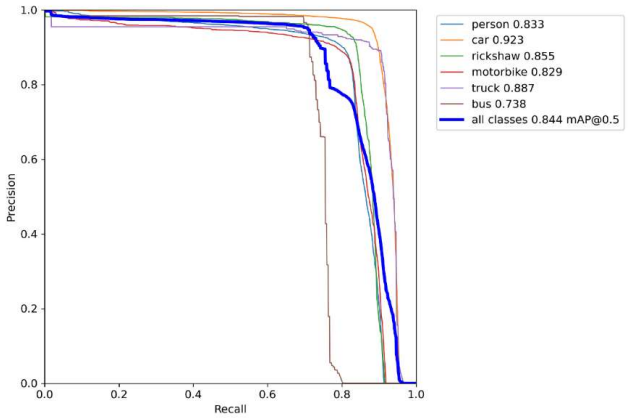

Figure 10. Training Performance (Precision and Recall) of the one-stage object detector

Figure 11. Object detection results of the single-stage detector on the proposed data collection

4.2. Two-Stage Detector

To demonstrate the diversity and applicability of the proposed dataset, experiments were extended to two-stage object detectors. For this purpose, a pre-trained Faster R-CNN was employed to evaluate the benchmark dataset. As mentioned above, for the base network of the two-stage detector, this study employed the VGG-16 model, pre-trained on the ImageNet dataset (Bansal et al., 2021) and further fine-tuned on our proposed benchmark dataset. During training, the weights of the batch normalisation layers in the pre-trained model were frozen to accelerate training and prevent overfitting. The detailed process of 2D object detection using a two-stage custom Faster R-CNN is illustrated in Algorithm 1.

4.3. Pseudocode for Training

Algorithm 1: Algorithm for 2D Object Detection (Two Stage Module using Custom FasterRCNN)

```
START
INPUT:       NIM, bb-annotation(label-location)
OUTPUT:      Rectangular Coordinates for surrounding detected objects, NFstrVGG16,
NFstrResnet18, NFstrMobilenetV2, AvgP_vgg16, AvgP_resnet18, AvgP_mobilenetv2
                  NIM: Number of input Images containing surrounding objects
                  bb-annotation: Rectangular bounding boxes of objects in images
                  Rectangular Coordinates for surrounding detected objects: Objects'
location
                  NFstrVGG16: VGGnet-based custom faster R-CNN model
                  NFstrResnet18: Resnet18-based custom faster R-CNN model
                  NFstrMobilenetV2: MobilenetV2 based custom faster rcnn model
                  PreObjLoc: function to Predict the rectangular coordinates for the
Object Location
// Estimation of anchor boxes
Þ ⟵   AnchorboxEstimation (NIM, bb-annotation)
input image size ⟵ [400 400], [500 500], [600 600]
NFstrVGG16       ⟵ ConstructCustomVggnetFasterRCNN (inputImageSize, Þ)
NFstrResnet18    ⟵ ConstructCustomResnet18FasterRCNN (inputImageSize, Þ)
NFstrMobilenetV2 ⟵ ConstructCustomMobilenetV2FasterRCNN (inputImageSize, Þ)
[ ϑ-trn, ϑ -tst] ⟵ splitting NIM into train and test set
// Training Module for 2D Object Detection
For each input training image n: ϑ-trn
            a) ϙn  ⟵  Feature Extraction VGG-16
            b) ϗn  ⟵  Feature Extraction Resnet18
            c) ɯn  ⟵  Feature Extraction MobilenetV2
End For
Train Model NFstrVGG16 for features ϙn
Train Model NFstrResnet18 for features ϗn
Train Model NFstrMobilenetV2 for features ɯn
₹_vgg16    ⟵  PreObjLoc (ϙn)
₹_res18    ⟵  PreObjLoc (ϗn)
₹_mobilev2 ⟵  PreObjLoc (ɯn)
Avgp_vgg16    ⟵  Evaluate_AvgP(NFstrVGG16, ₹_vgg16)
Avgp_resnet18 ⟵  Evaluate_AvgP(NFstrResnet18, ₹_res18)
Avgp_mobilenetv2 ⟵  Evaluate_AvgP(NFstrMobilenetV2, ₹_mobilev2)
For each testing image N: ϑ-tst
            a) ₾N  ⟵ Features extraction using selected trained model ɨ
            b) [boundingbox_coordinates, class_label] ⟵  Predict (₾N)
            c) Display input image with boundingbox_coordinates, class_label
            d)  ⊙⟵ [ ⊙ boundingbox_coordinates]
End For
Avgp_ɨ ⟵  Evaluate model ɨ using ⊙
FINISH
```

In this article, the region proposal network (RPN) was trained on a mini-batch before the classifier. The parameters of the Region Proposal Network and the base network were updated once, while the classifier was trained and updated using the positive and negative proposals generated by the Region Proposal Network. Consequently, the parameters were updated once whereas the base convolutional layers were updated twice. The Region Proposal Network and VGG-16based classifier share these base convolutional layers. Compared to the SGD optimiser used in the original Faster R-CNN, this study employed the Adam algorithm to improve the loss function. An adaptive learning rate was also employed, as mentioned in one stage detector, to get maximum performance in terms of speed and accuracy. Multiple backbone networks were implemented to check the performance of Faster R-CNN with different feature models. Training was performed on an

NVIDIA RTX 2080 Ti (11 GB) using Keras and TensorFlow backend. The dataset split ratio was the same as the previous one used for one stage detector, while the image is resized to multiple dimensions, including 400 x 400, 500 x 500, and 600 x 600. The image batch size for training was eight per GPU memory, and the max epochs was set to 100. The training process took around 48 hours to complete the maximum epochs. The proposed network architecture took near to the linear time complexity of $O(n)$.

This article used precision and recall performance metrics to evaluate the performance of the two-stage detector on our dataset. The mathematical formulae for these metrics are presented in Equations (1) and (2).

$$Precision = TP/TP + FP \quad (1)$$

$$Recall = TP/TP + FN \quad (2)$$

Here, TP (True Positive) refers to correctly predicted objects, FP (False Positive) refers to incorrectly predicted objects, and FN (False Negative) refers to objects that the model failed to predict. The performance of FasterR-CNN on the proposed dataset is shown in Table 2.

Table 2. Performance results of the two-stage detector on our proposed dataset

Resolution	Backbone	Performance Metrics	
		Average Precision	Recall
400 x 400	VGG-16	0.9126	0.8674
	Resnet-18	0.9041	0.8611
	Mobilenet-v2	0.9097	0.8638
500 x 500	VGG-16	0.9322	0.8880
	Resnet-18	0.9347	0.8894
	Mobilenet-v2	0.9286	0.8713
600 x 600	**VGG-16**	**0.9394**	**0.8910**
	Resnet-18	0.9321	0.8790
	Mobilenet-v2	0.9297	0.8761

It can be seen from the above table that Faster R-CNN performed exceptionally on our proposed dataset. Input image resolution, as well as backbone networks, has a significant effect on the accuracy of the model. The best results (i.e., 0.93 average precision and 0.89 recall) were achieved by the VGG-16 backbone network with a 600 x 600 pixel resolution image. These performance metrics demonstrated that two-stage object detectors can perform exceptionally well on the proposed dataset.

5. Comparison

When comparing the proposed dataset with state-of-the-art public datasets, the comparison was categorised in terms of vehicle classes and traffic scenarios. In Table 3, the proposed dataset was compared in different vehicle classes, including two-wheelers, three-wheelers, four-wheelers, and on-road pedestrians. Most of the existing datasets omit three-wheeler vehicles, which are highly prevalent in Asian countries, especially Pakistan, India, and Bangladesh. Similarly, in Table 4, the proposed data repository was compared to traffic scenarios, including urban, rural, dense, and motorway conditions.. Moreover, existing data collections were captured using low-resolution cameras, which can reduce performance and lead to overfitting when training deep neural networks.

However, this issue was addressed in the proposed data collection, where high-quality cameras were employed during capture.

Table 3. Comparison of the proposed data collection with existing datasets in terms of vehicle classes

Datasets	Dataset Size	Vehicle Class			
		Two-wheeler	Three-wheeler	Four-wheeler	Pedestrian
KITTI (Geiger et al., 2013)	14,000	-	-	✓	✓
Nepalese Vehicles (Bhatti et al., 2021)	4800	✓	-	✓	-
GTI (Li et al., 2022)	7325	-	-	✓	-
BIT (Dong et al., 2014)	9850	-	-	✓	-
UA-DETRAC (Wen et al., 2020)	140,000	✓	-	✓	✓
Proposed dataset	20000	✓	✓	✓	✓

Furthermore, most of the existing data collections are relatively small, regarding the total number of images and labels unsuitable for real-world deployment. However, some datasets have more images and annotations than ours. Nevertheless, these datasets lacked diverse traffic scenarios as they only contained data from urban traffic areas. Compared with existing datasets, the proposed data collection was more diverse in vehicle classes and traffic scenarios, which can significantly enhance environmental perception in varied traffic conditions (Table 4).

Table 4. Comparison of the proposed data collection with existing datasets based on traffic scenarios

Datasets	Resolution	Annotations	Traffic Scenarios				Total No of Labels
			A	B	C	D	
KITTI (Geiger et al., 2013)	0.5 MP	14,000	✓	-	✓	-	40,000
UA-DETRAC (Wen et al., 2020)	18 MP	8250	✓	-	✓	-	-
nuScenes (Caesar et al., 2020)	1.4 MP	90,000	✓	-	✓	-	-
TME Motorway Dataset (Caraffi et al., 2012)	1.0 MP	30,000	-	-	-	✓	-
Udacity (Ramanishka et al., 2018)	2.0 MP	15,000	✓	-	✓	-	-
Pandaset (Xiao et al., 2021)	2.1 MP	48,000	✓	-	✓	-	-
Caltech (Bansal et al., 2021)	0.5 MP	800	✓	-	-	-	-
Proposed dataset	12 MP	20000	✓	✓	✓	✓	55,000

6. Conclusion and Future Directions

To improve the ability of self-driving cars to perceive their environment, this paper proposed a vision benchmark dataset for on-road vehicle detection and recognition. This data repository was collected from more than 80 cities in Pakistan with diverse traffic scenarios. Different vehicle classes, including two-wheelers, three-wheelers, and four-wheelers, were incorporated into the proposed dataset, making it one of the most diverse of the Asian vehicles. To evaluate the proposed benchmark suite, different state-of-the-art CNN models, including one-stage and two-stage detectors, were employed and evaluated. These models demonstrated outstanding performance, which shows their effectiveness for the environmental perception of self-driving vehicles. Although

our ensemble has certain limitations, it does not include diverse weather conditions such as fog, rain, and snow.

Moreover, training our deep learning algorithms from scratch requires more data compared to using transfer learning. In the future, we aim to expand the dataset with diverse weather conditions including fog, rain, and snow, so that autonomous cars can navigate effectively on dynamic temperature and road conditions. Moreover, we will train our deep learning model from scratch rather than using transfer learning to improve the accuracy of these models on Asian-based traffic patterns.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

Amine, M. M., Fradi, H., Sahbani, A., & Amara, N. E. B. (2022). Unsupervised thermal-to-visible domain adaptation method for pedestrian detection. *Pattern Recognition Letters, 153,* 222–231. https://doi.org/10.1016/j.patrec.2021.11.024

Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2021). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing, 12*(1), 1–12. https://doi.org/10.1007/s12652-021-03488-z

Berwo, M. A., Khan, A., Fang, Y., Fahim, H., Javaid, S., Mahmood, J., Abideen, Z. U., & M. S., S. (2023). Deep learning techniques for vehicle detection and classification from images/videos: A survey. *Sensors, 23*(10), 4832. http://dx.doi.org/10.3390/s23104832

Bhatti, U. A., Yu, Z., Chanussot, J., Zeeshan, Z., Yuan, L., & Luo, W. (2021). Local similarity-based spatial–spectral fusion hyperspectral image classification with deep CNN and Gabor filtering. *IEEE Transactions on Geoscience and Remote Sensing, 60*(1), 1–15. https://doi.org/10.1109/TGRS.2021.3090410

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., & Xu, Q. (2020). nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11621–11631. IEEE. https://doi.org/10.1109/CVPR42600.2020.01164

Caraffi, C., Vojíř, T., Trefný, J., Šochman, J., & Matas, J. (2012). A system for real-time detection and tracking of vehicles from a single car-mounted camera. In *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 975–982. http://dx.doi.org/10.1109/ITSC.2012.6338748

Chen, W., Qiao, Y., & Li, Y. (2020). Inception-SSD: An improved single shot detector for vehicle detection. *Journal of Ambient Intelligence and Humanized Computing, 11*(3), 1–7. https://doi.org/10.1007/s12652-020-02085-w

Dong, Z., Pei, M., He, Y., Liu, T., Dong, Y., & Jia, Y. (2014). Vehicle type classification using a semi-supervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems, 16*(4), 2247–2256. https://doi.org/10.1109/ICPR.2014.39

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338. http://dx.doi.org/10.1007/s11263-009-0275-4

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research, 32*(11), 1231–1237. http://dx.doi.org/10.1177/0278364913491297

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. IEEE. https://doi.org/10.48550/arXiv.1311.2524

Han, Y., Wang, F., Wang, W., Li, X., & Zhang, J. (2024). YOLO-SG: Small traffic signs detection method in complex scene. *The Journal of Supercomputing, 80*(2), 2025-2046. https://doi.org/10.1007/s11227-023-05547-y

Li, T., Li, J., Liu, J., Huang, M., Chen, Y.-W., & Bhatti, U. A. (2022). Robust watermarking algorithm for medical images based on log-polar transform. *EURASIP Journal on Wireless Communications and Networking, 2022*(1), 1–11. https://doi.org/10.1186/s13638-022-02106-6

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), Computer vision – ECCV 2014. ECCV 2014. *Lecture notes in computer science, 8693*, 740–755. Springer. https://doi.org/10.1007/978-3-319-10602-1_48

Lin, C.-L., Fan, K.-C., Lai, C.-R., Cheng, H.-Y., Chen, T.-P., & Hung, C.-M. (2022). Applying a deep learning neural network to gait-based pedestrian automatic detection and recognition. *Applied Sciences, 12*(9), 4326. https://doi.org/10.3390/app12094326

Mohammad, J., Szalay, Z., & Török, Á. (2021). Evaluation of non-classical decision-making methods in self-driving cars: Pedestrian detection testing on cluster of images with different luminance conditions. *Energies, 14*(21), 7172. http://dx.doi.org/10.3390/en14217172

Muzammul, M., & Li, X. (2025). Comprehensive review of deep learning-based tiny object detection: challenges, strategies, and future directions. Knowledge and Information Systems, 1-89. http://dx.doi.org/10.1007/s10115-025-02375-9

Ramanishka, V., Chen, Y., Misu, T., & Saenko, K. (2018). Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7699–7707. http://dx.doi.org/10.1109/CVPR.2018.00803

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. IEEE. https://doi.org/10.48550/arXiv.1506.02640

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)*, 91–99. MIT Press. https://doi.org/10.48550/arXiv.1506.01497

Saxena, S., Dey, S., Shah, M., & Gupta, S. (2024). Traffic sign detection in unconstrained environment using improved YOLOv4. *Expert Systems with Applications, 238*, 121836. http://dx.doi.org/10.1016/j.eswa.2023.121836

Vishwakarma, P. K., & Jain, N. (2024). Design and augmentation of a deep learning based vehicle detection model for low light intensity conditions. *SN Computer Science, 5*(5), 605. http://dx.doi.org/10.1007/s42979-024-02944-9

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.-C., Qi, H., Lim, J., Yang, M.-H., & Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding, 193,* 102907. https://doi.org/10.1016/j.cviu.2020.102907

World Health Organization. (2018). *Global status report on road safety 2018*. World Health Organization. https://www.who.int/publications/i/item/9789241565684

Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., & Jiao, J. (2021). PandaSet: Advanced sensor suite dataset for autonomous driving. In *Proceedings of the IEEE International Intelligent Transportation Systems Conference (ITSC)*, 3095–3101. IEEE. https://doi.org/10.1109/ITSC48978.2021.9565009

Yeh, T., Lin, H., & Chang, C. (2021). Traffic light and arrow signal recognition based on a unified network. *Applied Sciences, 11*(17), 8066. https://doi.org/10.3390/app11178066

Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2018). Towards reaching human performance in pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 973–986. https://doi.org/10.1109/TPAMI.2017.2700460

Zhao, L., & Li, S. (2020). Object detection algorithm based on improved YOLOv3. *Electronics, 9*(3), 537. http://dx.doi.org/10.3390/electronics9030537

Zhou, S., Yang, H., Lashkov, I., Chen, C., Xu, H., Zhang, G., & Yang, Y. (2025). Deep learning-based vehicle detection and tracking from roadside LiDAR data through robust affinity fusion. *Expert Systems with Applications, 279*, 127338. http://dx.doi.org/10.1016/j.eswa.2025.127338