

Vehicle Detection from Overhead Satellite Images in Saudi Arabia with YOLOv12

Muhamad Syukron^{1,*}, & Radical Rakhman Wahid²

¹Department of Statistics, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo, Surabaya 60111, Indonesia

²Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Academic Belt Road, Dhahran 31261, Saudi Arabia

*Corresponding author: muhamad.syukron@its.ac.id

Abstract

One of the primary challenges in urban areas of the Kingdom of Saudi Arabia is managing traffic flow and implementing effective urban planning strategies in densely populated cities. Utilizing satellite imagery for vehicle counting presents an effective solution to enhance urban planning and economic monitoring. Based on a review of the literature, the lack of vehicle counting datasets using satellite imagery is a significant gap. To address this, we introduce the Saudi Vehicles Dataset, which comprises 550 images, each 500x500 pixels, with 5,609 manually annotated vehicles. To ensure the dataset's quality and generalization, we benchmarked it against two other satellite datasets: Synthetic Car and COWC. Benchmarking was performed using the latest version of the YOLO object detection algorithm, YOLOv12. The results indicate that our dataset yields comparable performance metrics to the other datasets. Specifically, when YOLOv12 was trained on the Saudi Vehicles Dataset, it achieved high scores for mAP50, recall, precision, and F1-score, with values of 0.977, 0.939, 0.945, and 0.942, respectively. While the best performance was observed on the Synthetic Car Dataset, achieving a perfect precision score of 1, the Saudi Vehicles Dataset showed superior performance in terms of recall, precision, and F1-score. The dataset is publicly available for use and can be accessed via this link: <https://github.com/Syukrondzeko/saudi-vehicle-dataset>.

Keywords: *deep learning; satellite image; Saudi Arabia; urban planning; vehicle detection; YOLOv12.*

Introduction

In today's world, managing traffic flow on the ground is a significant global concern, particularly in urban areas of the Kingdom of Saudi Arabia such as Riyadh, Jeddah, Haramain, and Dammam (Ministry of Municipal and Rural Affairs, 2019). Ensuring consistent monitoring of traffic presents a notable challenge. Historically, surveillance cameras and sensors have been the primary tools employed for this purpose. However, the impact of traffic congestion extends beyond transportation issues, influencing urban planning strategies in densely populated cities worldwide (Guyen et al., 2021). There is a need for large-scale traffic monitoring in which images from video cameras can be validated against overhead images without human intervention.

Leveraging satellite imagery for vehicle counting offers a promising solution for economic monitoring and enhances urban planning in cities. By analyzing vehicle density and movement in satellite images, city planners gain insights into traffic patterns, congestion, and transportation infrastructure usage (Maptionnaire, 2023). This data informs decisions on traffic optimization, infrastructure improvements, and public transportation development. Satellite-based vehicle detection provides a cost-effective and scalable approach for ongoing traffic monitoring, enabling continuous assessment and adaptation of urban planning strategies (Minetto et al., 2021).

To perform vehicle detection effectively, a well-structured dataset is essential for training machine learning or deep learning models. Previous studies, such as those by Hill (2021) and Stuparu et al. (2020), utilized aerial vehicle datasets such as VEDAI (Razakarivony & Jurie, 2016) and Cars Overhead with Context (COWC) (Mundhenk et al., 2016). These datasets primarily cover regions in the United States, Canada, Germany, and New Zealand. However, the absence of datasets from regions such as Asia, particularly Saudi Arabia, results in deep learning models that are less accurate and

less robust. This limitation arises because each country has distinct urban layouts, environmental conditions, and road structures, while the aforementioned datasets mainly represent Western regions. Furthermore, the dry terrain and intense sunlight in Arabian regions make vehicle detection even more challenging. Therefore, developing datasets from these geographical areas is necessary to improve model generalization and performance under such environmental conditions.

However, our research has identified only one available dataset for satellite images in Saudi Arabia, developed by Benjdira et al. (2019). This dataset consists of 128 images captured around the Prince Sultan University campus. Given its limited size and scope, dataset enrichment is necessary to develop a robust and generalizable model. Furthermore, the dataset is no longer publicly available, highlighting the need for an accessible dataset to enable researchers to develop deep learning models specifically tailored to satellite imagery of Saudi Arabia.

Furthermore, prior studies trained deep learning models on these datasets using architectures that are not state-of-the-art. For example, Hill (2021) and Stuparu et al. (2020) employed RetinaNet (Lin et al., 2018) for object detection. While accurate, RetinaNet is slower compared to other architectures specifically designed for object detection in video (Dupont, 2024), such as YOLO (Redmon et al., 2016). Benjdira et al. (2019), who previously developed a Saudi Arabia Dataset, used an older version of YOLO, namely YOLOv3 (Redmon & Farhadi, 2018). To conduct benchmarking, this study will use the state-of-the-art model, YOLOv12 (Tian et al., 2025), which was released in early 2025.

In summary, the contributions of this paper are as follows:

1. Creating a vehicle dataset from satellite images for detecting vehicles with geographical topology specific to Saudi Arabia.
2. Developing a labelling scheme that precisely labels all instances as “vehicle,” enabling deep learning models to detect vehicles more accurately while avoiding category imbalance.
3. Training YOLOv12 to benchmark the dataset and evaluating its performance for vehicle detection tasks to assess the effectiveness of this state-of-the-art model in the domain.

Materials and Methods

This section outlines the process of constructing the Saudi Vehicles Dataset, covering key steps such as data acquisition, cropping, and labeling. The dataset is then split into training, validation, and test sets to facilitate benchmarking using YOLOv12. The model's performance is evaluated using the Saudi Vehicles Dataset alongside two additional datasets, COWC (Mundhenk et al., 2016) and Synthetic Car (Orić et al., 2024). The evaluation employs precision, recall, F1-score, and mean Average Precision (mAP) as performance metrics. A detailed flowchart of the process is presented in Figure 1.

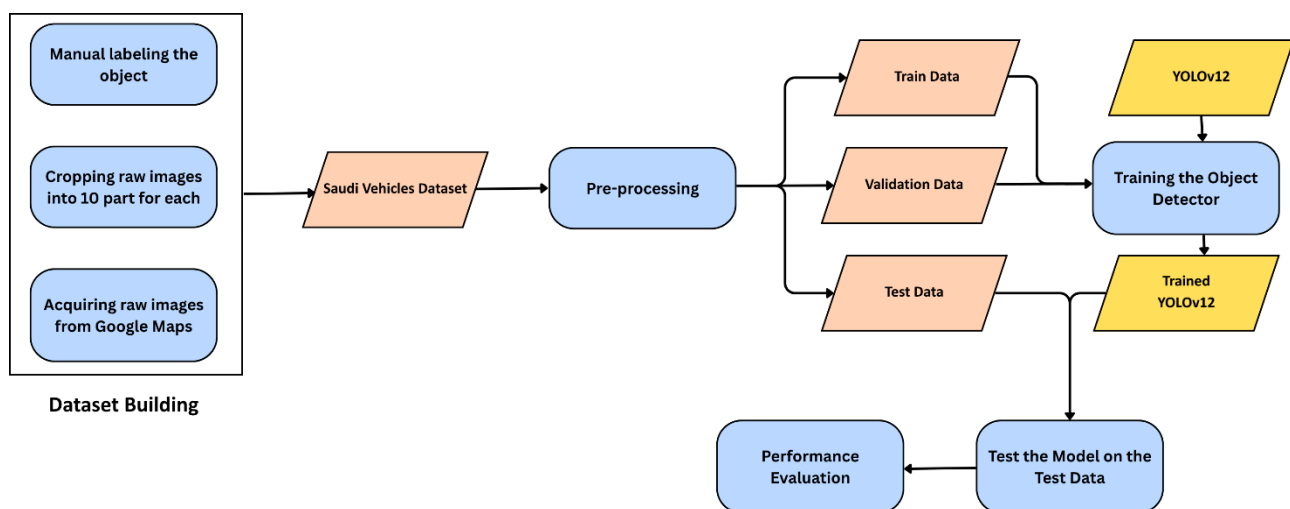


Figure 1 Experimental workflow from image acquisition to performance evaluation.

Saudi Vehicles Dataset

The satellite imagery used in this study for training the YOLO model was sourced from Google Maps, subject to Google's non-commercial Terms of Service for academic research. We utilized Google Maps satellite images because they are publicly available and constantly updated. The dataset comprises 550 unique images captured from various regions, including Riyadh, Jeddah, Khobar, Dammam, and Buraydah, collected between December 1 and December 15, 2024. The image dimensions are 500×500 pixels. To ensure the object detector's ability to generalize across object sizes, we included zoom levels of 10 m and 20 m (see Figure 2). Since the dataset contains vehicle data captured from satellite imagery over Saudi Arabia, we named it the Saudi Vehicles Dataset. For evaluation, we partitioned the dataset into training, validation, and testing splits with a ratio of 70%, 10%, and 20%, resulting in 350 images for training, 50 images for validation, and 100 images for testing.



Figure 2 Cropped sample images at 10 m (top) and 20 m (bottom) zoom levels.

The final step in building the dataset was manual labeling/annotation. The annotation process in this research was done using the Yolo_Label application (Kwon, 2023). The first step was to open the folder containing all the cropped images and load them into the application. Then, for each image, we manually draw a bounding box around each vehicle. After drawing the box, we assign it a class label, such as 'vehicle,' since it only detects vehicles. The application generates a text file for each image, containing the coordinates of the bounding boxes and the class label of the objects. These coordinates are normalized to fall between 0 and 1. While training for standard image classification only requires the image, image object detection also needs the coordinates of its objects in the ".txt" file. Through manual annotation, we have identified 5,609 vehicles across all the images.

YOLOv12 Object Detection Algorithm

Real-time object detection has garnered significant attention due to its low-latency characteristics, particularly in applications requiring rapid detection, such as vehicle monitoring on roads. One of the most powerful and widely used object detection models is YOLO. YOLOv12 represents the twelfth iteration in the You Only Look Once (YOLO) family of models. It surpasses all existing real-time object detectors in accuracy while maintaining competitive inference speed. For instance, YOLOv12 improves mean average precision (mAP) by 1.2% compared to its predecessor, YOLOv11 (Khanam & Hussain, 2024), without compromising inference speed (Tian et al., 2025). YOLOv12, like the previous YOLO version, has several types which are YOLO-N (Nano), YOLO-S (Small), YOLO-M (Medium), YOLO-L (Large), YOLO-X (Extra Large). In this experiment, we will use the Nano, Small, and Medium version due to the limitation of computing power.

In this experiment, we configured the maximum number of iterations to 200 and set the class number to 0, as the only object of interest is 'vehicle'. We set the batch size of training as 16. To reduce the risk of getting trapped in local minima and to stabilize training, we used a learning rate of 0.002. The majority of activation functions used are Sigmoid Linear Unit (SiLU), with some instances of linear activation. For the loss function, we utilized Complete Intersection over Union (CIoU).

Dataset Benchmarking

To evaluate the quality of our dataset, we perform benchmarking to compare its performance with other datasets. If the results are unsatisfactory, even though other datasets perform well, it may indicate issues with image quality or labeling. For benchmarking, we use well-known aerial vehicle image datasets, namely COWC and Synthetic Car. Given the limitations of the hardware accelerator, we adjust the data accordingly. Instead of using the entire dataset, we select a subset from each. For the Synthetic Car Dataset, we used 50 images provided by the authors, resizing them to a resolution of 500x500 pixels. For the COWC Dataset, we utilized 1,568 images with a resolution of 682x682 pixels.

Each dataset is divided into training (70%), validation (20%), and testing (10%) subsets. Training data is used to fit YOLOv12, while validation data were used to monitor training and prevent overfitting or underfitting. The test data is used to assess the model's ability to detect vehicles in previously unseen data. All experiments, from building the vehicle detection model to performance evaluation, are conducted on the Google Colab platform with Python version 3.10.2. Additionally, we use the NVIDIA T4 hardware accelerator provided by Google Colab. For vehicle detection, the YOLO model is implemented using the Ultralytics framework version 8.3.327 (Ultralytics YOLOv12: <https://github.com/ultralytics/ultralytics/blob/main/docs/en/models/yolo12.md>). For comparative analysis, the RetinaNet model is implemented using the Torchvision framework version 0.24.0+cu126 (Torchvision RetinaNet: <https://github.com/pytorch/vision/blob/main/torchvision/models/detection/retinanet.py>).

For benchmarking, we evaluate model performance using four metrics: precision, recall, F1-score, and mAP50. Precision measures the model's accuracy in making correct positive predictions, while recall assesses its ability to identify all relevant objects within an image. The F1-score provides a balanced measure of performance by considering both false positives and false negatives. mAP50, on the other hand, offers a comprehensive evaluation of the model's robustness across various detection scenarios, indicating that at least 50% of the bounding box area overlaps with the detected vehicle.

Results

Selection of YOLOv12 Variant

Table 1 presents the detection performance of the YOLOv12 Nano, Small, and Medium variants on the Saudi Vehicles Dataset. The Small and Medium models achieve higher values across all evaluation metrics than the Nano model. The Small model records precision, recall, and F1 score values of 0.927, while the Medium model achieves identical values. For mAP50, the Medium model attains 0.971, whereas the Small model reaches 0.970. Standard deviations remain low across all experiments, indicating consistent performance across repeated runs.

Table 1 Performance comparison of YOLOv12 variants on the Saudi Vehicles Dataset.

Metrics	YOLOv12 Nano	YOLOv12 Small	YOLOv12 Medium
Precision	0.895 (0.002)	0.927 (0.001)	0.927 (0.002)
Recall	0.926 (0.002)	0.927 (0.001)	0.927 (0.001)
F1 Score	0.910 (0.001)	0.927 (0.002)	0.927 (0.001)
mAP50	0.962 (0.003)	0.970 (0.002)	0.971 (0.002)

Table 2 summarizes the computational cost and model complexity of the evaluated models. Training time increases from the Nano model to the Medium model. Similarly, the number of parameters and FLOPs increase substantially with model size. The Medium model requires the longest training time and has the highest computational complexity, while the Nano model has the lowest computational requirements.

Table 2 Computational cost and model complexity of YOLOv12 variants.

Metrics	YOLOv12 Nano	YOLOv12 Small	YOLOv12 Medium
Training Time	28 - 30 minutes	35 - 38 minutes	45 - 50 minutes
Num of Parameters	2.6 millions	9.3 millions	20.2 millions
Size	6.5 Giga FLOPs	21.4 Giga FLOPs	67.5 Giga FLOPs

Comparison Between YOLOv12 and RetinaNet

To compare YOLOv12 with other modern object detection models that have been applied to aerial vehicle detection, experiments were conducted using RetinaNet on the Saudi Vehicles Dataset. Based on the results presented in Tables 1 and 2, the YOLOv12 Small variant was selected for comparison because it achieved performance comparable to the Medium model while requiring fewer computational resources.

Table 3 presents the performance comparison between YOLOv12 Small and RetinaNet. YOLOv12 Small achieves higher precision, F1 score, and mAP50 values than RetinaNet. Recall values remain similar between the two models. Standard deviations are low for both approaches, indicating stable performance across repeated experiments.

Table 3 Performance comparison between YOLOv12 Small and RetinaNet.

Metrics	YOLOv12 Small	RetinaNet
Precision	0.927 (0.001)	0.870 (0.003)
Recall	0.927 (0.001)	0.924 (0.001)
F1 Score	0.927 (0.002)	0.897 (0.002)
mAP50	0.970 (0.002)	0.935 (0.003)

Benchmark Evaluation Across Dataset

Figure 3 illustrates the validation loss and mAP50 curves obtained during training on the Saudi Vehicles, COWC, Synthetic Car, and VEDAI datasets. Validation loss decreases throughout training for all datasets, while mAP50 increases and gradually stabilizes. The Saudi Vehicles dataset achieves the highest mAP50, followed by the COWC and Synthetic Car datasets. The VEDAI dataset records the lowest mAP50 among the evaluated datasets.

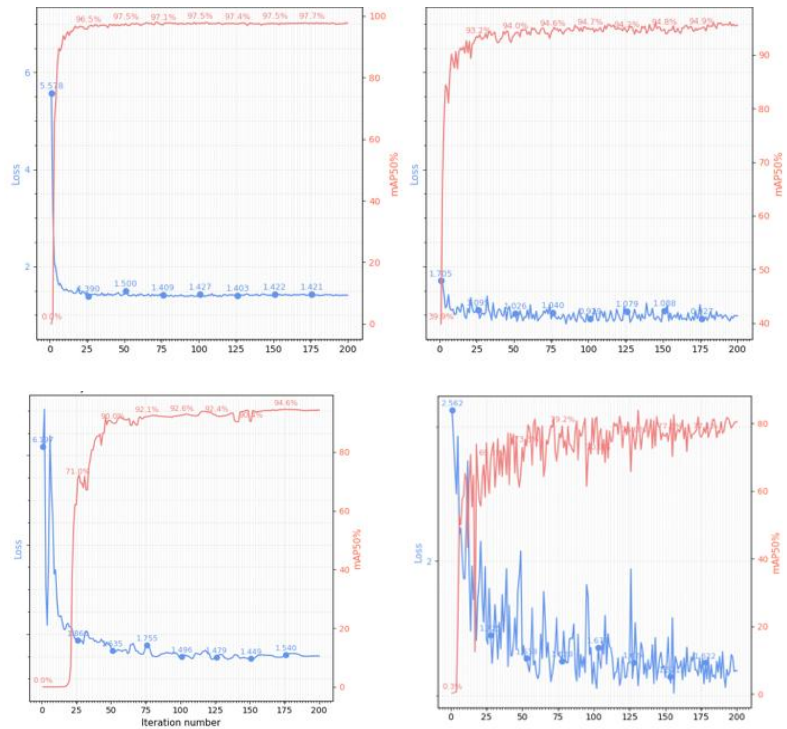


Figure 3 Validation loss and mAP50 for Saudi Vehicles (upper left), COWC (upper right), Synthetic Car (bottom left), and VEDAI Dataset (bottom right).

Table 4 presents the validation performance of YOLOv12 Small on the four datasets. The Saudi Vehicles dataset achieves the highest mAP50 value, while the VEDAI dataset records the lowest performance across all evaluation metrics. The COWC and Synthetic Car datasets demonstrate intermediate performance levels between these two datasets.

Table 4 YOLOv12 small performance on the validation dataset.

Metrics	Saudi Vehicles	COWC	Synthetic Car	VEDAI
Precision	0.945 (0.001)	0.916 (0.002)	1.000 (0.004)	0.710 (0.002)
Recall	0.939 (0.002)	0.913 (0.003)	0.961 (0.002)	0.855 (0.003)
F1 Score	0.942 (0.002)	0.914 (0.004)	0.943 (0.002)	0.776 (0.004)
mAP50	0.977 (0.002)	0.957 (0.002)	0.924 (0.001)	0.839 (0.002)

Table 5 reports the test performance of YOLOv12 Small. The Saudi Vehicles dataset maintains performance levels similar to those observed on the validation set. The COWC and Synthetic Car datasets exhibit lower performance on the test set than on the validation set. The largest decrease is observed in the recall score of the Synthetic Car dataset. The VEDAI dataset shows similar performance between validation and test evaluations.

Table 5 YOLOv12 Small performance on the test dataset.

Metrics	Saudi Vehicles	COWC	Synthetic Car	VEDAI
Precision	0.927 (0.001)	0.887 (0.003)	0.934 (0.005)	0.708 (0.002)
Recall	0.927 (0.001)	0.881 (0.002)	0.780 (0.004)	0.828 (0.001)
F1 Score	0.927 (0.002)	0.884 (0.004)	0.853 (0.005)	0.766 (0.001)
mAP50	0.970 (0.002)	0.922 (0.003)	0.880 (0.002)	0.834 (0.002)

Qualitative Detection Result

Figures 4 and 5 present sample detection results on the Saudi Vehicles Dataset at 20 m and 10 m zoom levels, respectively. Vehicles are successfully detected across different urban scenes and imaging scales. In both figures, most vehicles are correctly localized and classified. Some vehicles located near image boundaries remain undetected.

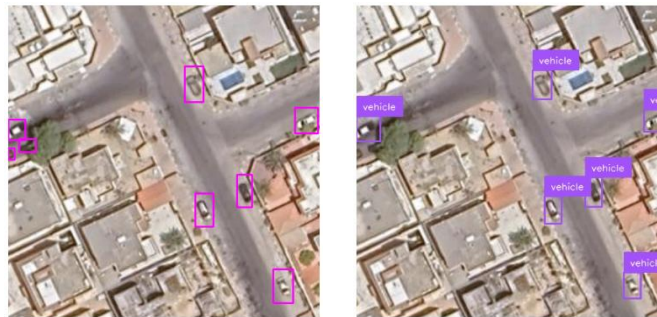


Figure 4 Detection results on the Saudi Vehicles Dataset at 20 m zoom: ground truth (left) and predictions (right).



Figure 5 Detection results on the Saudi Vehicles Dataset at 10 m zoom: ground truth (left) and predictions (right).

Figure 6 presents sample detection results on the benchmark datasets. YOLOv12 successfully identifies vehicles in the COWC, Synthetic Car, and VEDAI datasets. Detection results are observed across scenes with different levels of object density, background complexity, and vehicle distribution patterns.

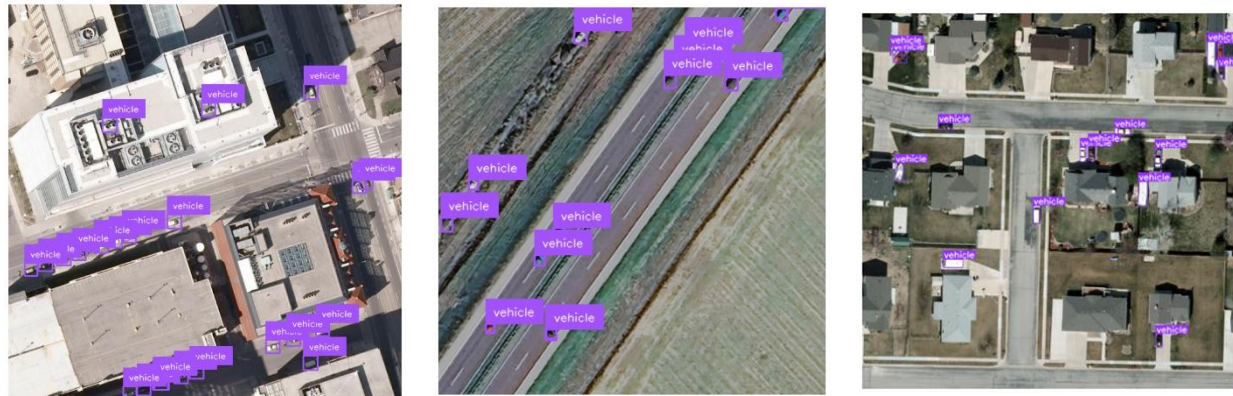


Figure 6 Sample detection results on benchmark datasets, including COWC (left), Synthetic Car (middle), and VEDAI Dataset (right).

Discussion

Model Performance and Generalization

The results demonstrate that increasing model complexity does not necessarily lead to substantial improvements in detection performance. Although YOLOv12 Medium contains considerably more parameters and computational operations than YOLOv12 Small, the performance differences between the two models are negligible across all evaluation metrics. This finding indicates that the representational capacity of the Small model is sufficient for learning discriminative vehicle features from the Saudi Vehicles Dataset. From a practical perspective, the trade off between accuracy and computational cost is an important consideration for real world deployment. The Medium model requires substantially longer training times and significantly higher computational resources, yet provides only marginal performance improvements. Consequently, YOLOv12 Small offers a more favorable balance between detection accuracy and computational efficiency, making it a suitable choice for subsequent experiments.

The comparison with RetinaNet further highlights the effectiveness of YOLOv12 for aerial vehicle detection. YOLOv12 Small achieves higher precision, F1 score, and mAP50 while maintaining a recall value comparable to RetinaNet. These results indicate that YOLOv12 is able to identify vehicles accurately while producing fewer false positive detections. The superior performance is consistent with the strengths of modern YOLO architectures, which integrate localization and classification within a unified detection framework and have demonstrated strong performance in remote sensing applications.

The training curves presented in Figure 3 show that all datasets converge successfully during training. The decreasing validation loss and increasing mAP50 values indicate that the model learns meaningful visual representations across different aerial imagery datasets. The relatively rapid stabilization of the performance curves may be attributed to the use of transfer learning, where YOLOv12 is initialized using weights pre trained on the COCO dataset (Lin et al., 2015) before being fine tuned for vehicle detection.

Differences in performance among the datasets reveal the influence of dataset characteristics on model behavior. The Saudi Vehicles Dataset consistently achieves the strongest performance across validation and test evaluations. Moreover, the similarity between validation and test results suggests strong generalization capability, indicating that the learned features remain effective when applied to previously unseen images. The Synthetic Car dataset exhibits excellent validation performance but experiences a noticeable decline on the test set, particularly in recall. This pattern suggests reduced generalization capability compared with the Saudi Vehicles Dataset. Because the vehicles are synthetically generated, the dataset contains more uniform visual characteristics than real world imagery. As a result, the model may learn dataset specific patterns that do not transfer equally well to unseen samples. In contrast, the VEDAI dataset achieves the lowest performance overall, which may be attributed to greater scene diversity, background

complexity, and variations in imaging conditions. These factors increase the difficulty of distinguishing vehicles from surrounding objects and have been recognized as common challenges in aerial object detection.

Overall, the benchmark comparison demonstrates that the Saudi Vehicles Dataset provides a suitable balance between visual diversity and data consistency. The strong performance achieved on both validation and test data indicates that the dataset is effective for training and evaluating deep learning based vehicle detection models.

Qualitative Analysis and Limitations

The qualitative results support the quantitative findings reported in the previous section. Figures 4 and 5 demonstrate that YOLOv12 can accurately detect vehicles across different zoom levels within the Saudi Vehicles Dataset. The model is also capable of distinguishing vehicles from surrounding urban structures, including rooftop objects that may exhibit visual characteristics similar to vehicles when viewed from above. This capability contributes to the high precision observed during quantitative evaluation.

The benchmark examples shown in Figure 6 further demonstrate the adaptability of YOLOv12 across datasets with different visual characteristics. The model successfully identifies vehicles in urban environments, highway scenes, and synthetic imagery. However, performance differences can be observed depending on object density and scene complexity. In densely populated regions, multiple adjacent vehicles occasionally appear as a single detection, indicating that closely spaced objects remain challenging for the detector.

Despite the strong overall performance, several limitations were identified during the experiments. One notable limitation is the reduced ability of YOLOv12 to detect vehicles located near image boundaries. Partially cropped vehicles contain incomplete shape and texture information, which reduces detection confidence and increases the likelihood of misclassification as background. This issue is visible in both Figures 4 and 5, where some truncated vehicles are not detected successfully.

Another limitation concerns scenes containing a large number of closely spaced vehicles. Although YOLOv12 generally performs well in such environments, neighboring vehicles may occasionally be merged into a single detection. These findings suggest that further improvements in small object representation and instance separation could enhance performance. Future work may investigate advanced data augmentation strategies, boundary aware detection techniques, and higher resolution feature representations to improve detection performance in challenging aerial imagery scenarios.

Conclusion

In this study, we successfully developed a new satellite overhead dataset containing vehicle images from the Kingdom of Saudi Arabia, named the Saudi Vehicles Dataset. Furthermore, we implemented YOLOv12 as an object detection algorithm to count vehicles in this dataset, achieving excellent results. The model obtained precision, recall, F1-score, and mAP50 values of 0.945, 0.939, 0.942, and 0.977, respectively, on the validation data. On the test data, it achieved 0.927, 0.927, 0.927, and 0.970 for the same metrics. To assess the generalizability of the model, we benchmarked YOLOv12 on additional public datasets using transfer learning to accelerate convergence. The results indicate that the model maintains strong performance and achieves comparable scores across different datasets.

With this Saudi Vehicles Dataset, the government can improve smart city planning. Accurate vehicle detection and counting from satellite imagery can support traffic flow analysis, urban planning, parking management, and transportation optimization. These applications contribute directly to data-driven decision-making in areas such as infrastructure development and congestion control. Moreover, integrating this dataset into urban analytics frameworks can assist in monitoring population mobility patterns and reducing environmental impacts through better traffic and emission management, aligning with the goals of Saudi Vision 2030 to develop sustainable and intelligent urban environments.

For future research, several directions can further enhance the usefulness of this work. First, high-resolution image processing and enhancement techniques can be applied to address the low-resolution in Google Maps imagery, improving vehicle detection accuracy. Second, the detected vehicle counts can be leveraged for environmental analysis, such as estimating emissions, assessing traffic congestion, and monitoring parking occupancy patterns. These

improvements can increase the practical impact of the dataset and support more comprehensive urban and environmental planning efforts.

Compliance with ethics guidelines

The authors declare they have no conflict of interest or financial conflicts to disclose.

This article contains no studies with human or animal subjects performed by the authors.

References

- Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A. & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3, *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, 1–6. <https://doi.org/10.1109/UVS.2019.8658300>
- Dupont, M. (2024). *Comparing YOLOv10 with Other Object Detection Models*. <https://www.labelvisor.com/comparing-yolov10-with-other-object-detection-models/>. Accessed 2 June 2026.
- Guven, C., Karaduman, O. & Avci, E. (2021). A Review on Urban Intelligent Traffic Management Problems: Sensors and Methods, *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, 1–6. <https://doi.org/10.1109/IISEC54230.2021.9672412>
- Hill, C. (2021). Automatic Detection of Vehicles in Satellite Images for Economic Monitoring [University of South Florida]. <https://digitalcommons.usf.edu/etd/8792/>. Accessed 2 June 2026.
- Khanam, R., & Hussain, M. (2024). YOLOv11: An Overview of the Key Architectural Enhancements. arXiv preprint arXiv:2410.17725. <https://arxiv.org/abs/2410.17725>.
- Kwon, Y. (2023). YOLO-Label. https://github.com/developer0hye/Yolo_Label. Accessed 2 June 2026.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). *Focal Loss for Dense Object Detection*. <https://arxiv.org/abs/1708.02002>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). *Microsoft COCO: Common Objects in Context*.
- Maptionnaire. (2023). *GIS in Urban Planning: Application, Tools & Examples*. <https://www.maptionnaire.com/blog/gis-in-urban-planning-benefits-application-examples>
- Minetto, R., Segundo, M. P., Rotich, G. & Sarkar, S. (2021). Measuring Human and Economic Activity from Satellite Imagery to Support City-Scale Decision-Making During COVID-19 Pandemic, *IEEE Transactions on Big Data*, 7(1), 56–68. <https://doi.org/10.1109/TBDATA.2020.3032839>
- Ministry of Municipal and Rural Affairs. (2019). *Saudi Cities Report 2018 – Executive Summary*. https://saudiarabia.un.org/sites/default/files/2020-01/SSCR2018-ENG_0.pdf
- Mundhenk, T. N., Konjevod, G., Sakla, W. A., & Boakye, K. (2016). A large contextual dataset for classification, detection and counting of cars with deep learning. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016 (Lecture Notes in Computer Science, Vol. 9907)*. Springer, Cham. https://doi.org/10.1007/978-3-319-46487-9_48
- Orić, M., Galić, V., & Novoselnik, F. (2024). Synthetic car dataset for vehicle detection: Integrating aerial and satellite imagery. *Data in Brief*, 53, 110105. <https://doi.org/10.1016/j.dib.2024.110105>
- Razakarivony, S. & Jurie, F. (2016). Vehicle detection in aerial imagery : A small target detection benchmark. *Journal of Visual Communication and Image Representation*, 34, 187–203. <https://doi.org/10.1016/j.jvcir.2015.11.002>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779–788). IEEE. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767. <https://arxiv.org/abs/1804.02767>. Accessed 16 September 2026.
- Stuparu, D.-G., Ciobanu, R.-I. & Dobre, C. (2020). Vehicle Detection in Overhead Satellite Images Using a One-Stage Object Detection Model. *Sensors*, 20(22), 6485. <https://doi.org/10.3390/s20226485>
- Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-centric real-time object detectors. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 87151–87175. <https://doi.org/10.52202/085713-2627>