Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Fine-Tuning YOLOv8 for Top-View Vehicle Detection: Dataset Preparation, Training, and Performance Analysis for Traffic Surveillance

Muhammad Zain Ul Abdin[1, *]

[1]Department of Computer Science, NCBA&E, Multan, 60000, Pakistan
[*]Corresponding Author: Muhammad Zain Ul Abdin. Email: zainulabdin1472@gmail.com

Received: 29 November 2025; Revised: 14 January 2026; Accepted: 07 February 2026; Published: 16 March 2026
AID: 005-01-000061

Abstract: Traffic surveillance systems require a very high level of accuracy and real-time vehicle detection so it can be used to enable intelligent transportation and city management applications. The study proposes a detailed study on how to fine-tune the YOLOv8 object detector to top-view aerial vehicle detector in traffic monitoring sites. The model was trained using a domain-specific dataset of 626 annotated aerial images of vehicles, which with the use of transfer learning delay the optimization of detection precision and recall. The training was done with a batch size of 32, the resolution of the input normalized to 640×640 pixels, and a 100-epoch run in its full Tesla P100 GA. The model is good at finding different types of vehicles in different traffic situations since it has a high mean Average Precision (mAP50) of 97.5% and a strong mAP50-95 of 74.2%, as well as precision and recall rates over 91%. The ability to draw conclusions within seconds (about 4 milliseconds per picture) renders the technique even more applicable to the live traffic surveillance systems. Integration with Weights & Biases made it possible to keep an eye on all aspects of the training process, which made sure that convergence was stable and the model could be used in other situations. The results show how important it is to use specialized datasets and fine-tune them to improve systems for detecting aerial vehicles for smart traffic management. The main argument of the present study is that a domain-specific highly accurate aerial vehicle detector can be obtained using a rather small dataset with the use of efficient fine-tuning of YOLOv8. Such research provides a computationally low and efficient model compared to the literature available, which is constituted by huge data sets, or a complex architecture that could be utilized in real time and in traffic monitoring systems.

Keywords: YOLOv8; Traffic Surveillance; Vehicle Detection; Top-View Vehicle Detection;

1. Introduction

One of the most significant computer vision problems is object detection. It is an automatic process of detecting and identifying objects in videos or images. This is a significant task due to a variety of applications in the real-world, in particular in the field of smart traffic tracking where the capability to identify vehicles correctly in real time is crucial to the traffic flow analysis systems, traffic jams regulation, and accident prevention. Finding a compromise between efficiency and accuracy regarding computation

https://doi.org/10.66108/mna.v5i01.88

was difficult without new deep learning architectures, which have revolutionized object detection in every aspect [1].

YOLO family of architectures have offered a paradigm shift due to their ability to obtain detection through only one forward pass; hence, capable of speeds that can achieve real-time processing. In 2023, Ultralytics released YOLOv8, a follow-up of this feature; it features an anchor-free detection head, a superior backbone, Darknet53, alongside novel loss algorithms that produce improved localization and classification [2]. These adjustments render YOLOv8 particularly suitable to tough crowds such as identifying airborne automotive, wherein slight variations in scale, obscuring, and lighting circumstances pose unique issues.

Even though much progress has been achieved in detecting objects, current literature on detecting aerial vehicles uses large-scale or is general-purpose object detectors with limited data (domain-specific) challenges properly considered. Specifically, the literature has not concentrated its research on the optimization of lightweight and efficient models such as YOLOv8 on small top-view datasets of traffic and at the same time ensuring real-time application. This gap indicates the necessity of systematic research on the preparation of datasets, fine-tuning methods, and performance measurement in terms of a specific situation with aerial traffic surveillance.

According to several recent researches, aerial vehicles detection models tend to rely on big-scale benchmarks and may not more accurately represent domain-specific conditions when data about them is scarce. Besides, problems (size of small objects, occlusion, different viewpoints with top-view imagery) cannot be tackled sufficiently unless task-specific fine-tuning is taken into account. Such deficiencies of the current literature also explain why the creation and assessment of efficient solutions adapted to small and specialized aerial datasets is required.

To fill the defined gap in research, the following objectives are followed in this study:

- Top-view aerial vehicle detection: Domain-specific dataset
- To obtain an annotated top-view aerial vehicle dataset, including top-view aerial vehicle data and preprocessing.
- To use transfer learning to optimize the YOLOv8 model to achieve a better detection performance in aerial images.
- To compare the model with the help of traditional object detection performance measures like precision, recall and mAP.
- To examine the trade-off between precision of detection and real time inference performance of traffic surveillance applications.

This study looks into how to fine-tune the YOLOv8 model for vehicle detection from aerial top-view photographs and how well it works. To obtain such views, increasingly, the current systems of traffic monitoring use drone cameras or permanently mounted cameras on towers. It means that the detectors should be able to detect various visual signs in contrast to traditional ground level images [3]. We explain our data curating, annotating and preparing process in detail and give a detailed explanation of our fine-tuning strategy, the choice of hyperparameters, and our performance metric. The final stages of the study are analysis of results and conclusions about implications of deployment on the real-life traffic monitoring scenario.

2. Related Work

Traditionally, object detection methods have been developed based on region-based convolutional neural networks (R-CNN) that suggested candidate regions and did classification, to more combined ones such as SSD and YOLO that combined localization and classification tasks into a single streamlined system. R-CNN and faster versions made important foundations, but were either too computationally intensive to be practiced in the real time [4]. SSD brought in a lot of speed by quantifying the output space to the number of bounding boxes and scores in a fixed set which boosted speed with a small tradeoff on accuracy [5].

Started with the first entry in the YOLO series with the initial YOLO in 2016 changed the paradigm of looking at detection as an end-to-end regression task where impressive inference times can be achieved [6]. Earlier implementations, namely YOLOv2-YOLOv7, were sequential upgrades in accuracies, offering multi-scale prediction, and network architecture refinements [7]. YOLOv8 expands these innovations with anchor boxes removed and an anchor-free system, which makes prediction simpler and allows pixel-wise estimates of bounding boxes, akin to segmentation models, thus offering better performance in sparse object scenes [2].

In the application of aerial vehicle detection, recent studies have highlighted the issues surrounding the inability to identify vehicles in top-down imaging because of their tiny sizes, different orientations, and the complexity of their backgrounds [8, 9]. Deep learning models in the form of Faster R-CNN, SSD, and YOLO variants have been used in the studies to overcome such challenges with a strong focus on domain-specific dataset curation and augmentation techniques [10]. Transfer learning has become a general procedure to take advantage of large-scale trained weights, so that training is less time-consuming, and generalizing to smaller specialized datasets is more effective [11].

Our study fits the studies within this category because we have used the state-of-the-art YOLOv8 architecture on a well-crafted vehicle dataset (top-view) and tested its ability to meet the demands of the task, assessing its results with a series of experimental findings and performance indicators.

3. Methodology

Since it is necessary to define a clear relationship with the study goals, a general methodology was built to be the structured pipeline based on the experience of transfer learning in 4 major steps, i.e., (1) data preparation, and annotation of top-view vehicle images, (2) tuning and setting up of the YOLOv8 model, (3) performing an actual training of the model with the optimal hyperparameters, and (4) evaluating These phases are closely linked with the research aims themselves, and as such, an estimation of the model performance in terms of the aerial traffic surveillance can be performed in a systematic fashion.

Below is the proposed methodology diagram.

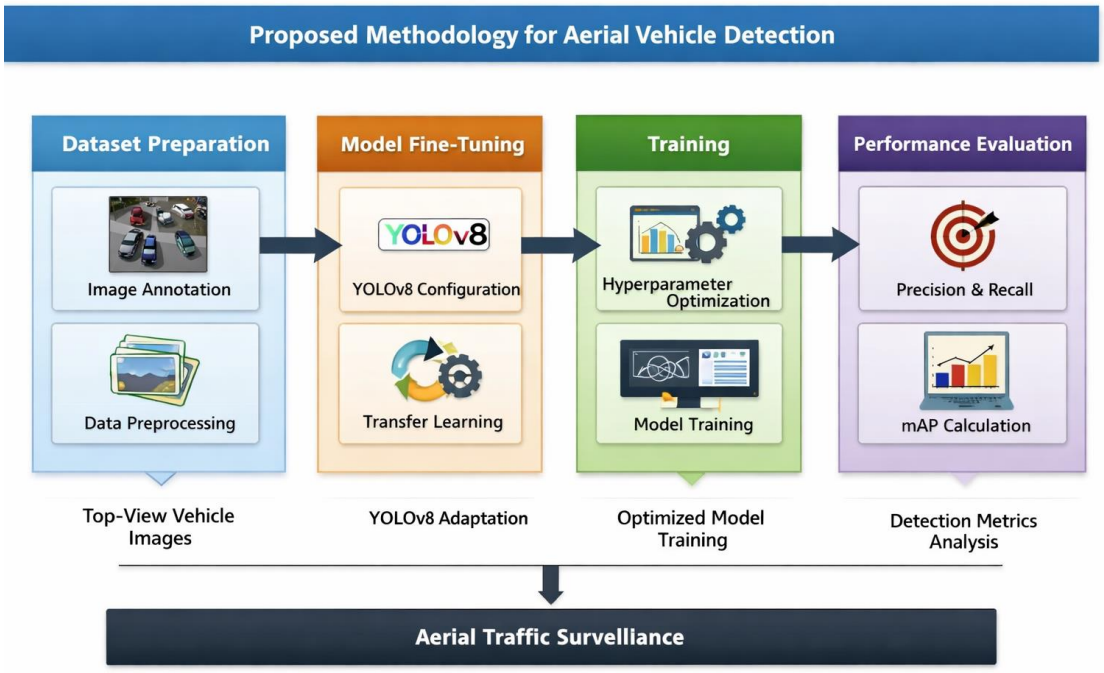

Figure 1: Proposed Methodology Diagram for Aerial Vehicle Detection

3.1. Object Detection and YOLOv8 Framework

The problem of object detection is to determine all the occurrences of defined object classes in an image with bounding boxes and probability of the class. YOLOv8 improves upon previous YOLO versions with the implementation of a Darknet53 backbone network, which provides a better ability to extract features through more convolutional layers and more feature maps at a variety of scales. The anchor-free detection head of the model estimates bounding boxes on a pixel-level level of granularity, enhancing both the localization accuracy and object detection of different shapes and sizes. Moreover, YOLOv8 comes with a new loss function, the combination of localization and classification with distribution focal loss terms that increase convergence and generalization power of the model [2].

Table 1: YOLOv8 Model Comparison Table

Model	Size (pixels)	mAPval 50–95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

3.2. YOLOv8 Model Variants and Trade-offs

The YOLOv8 model comes in five sizes nano (n), small (s), medium (m), large (l), and xlarge (x). The empirical analysis reveals the steady tendency: bigger models tend to have high mean Average Precision (mAP) because of more parameters and due to better feature representations, yet, at the cost of low inference times and higher computational demands [12]. As an illustration, the nano size has about 1.9 million parameters with 3.1 GFLOPs, which can provide fast inference but with low accuracy, whereas the xlarge size has more than 50 million parameters and 144 GFLOPs, which is more suitable to use where the accuracy is more important than the speed. In cases where real-time performance is important as in traffic surveillance, the medium or large model usually offers a trade-off between speed and precision.

All YOLOv8 models have a universal input resolution of 640 640 pixels, meaning the input pipeline is standardized and thus model comparisons can be made. This is a well thought size that will maximize accuracy of the detection without overloading the computation level.

3.3. Evaluation Metrics

To stringently test the performance of detection, we can take some well-defined metrics used in the object detection community:

- **Intersection over Union (IoU):** The ratio between the intersection area of the predicted bounding boxes and the ground truth can be used to measure as the overlap. IoU of 0.5 (50%), 0.75 or other values are used to establish whether a detection is counted as a true positive, which affects the calculation of precision and recall [13].
- **Mean Average Precision (mAP):** The mean of scores of precision calculated at different levels of recall, summed between all classes as well as at different levels of IoU. mAP50 is defined as mAP at 50% IoU and mAP50-95 is the average mAP across IoU thresholds between 95 and 50 percent in 5-percent steps, giving a higher quality evaluation of detection [14].
- **Precision:** The proportion of correct positive identifications/finds out of all positive findings, which measure how accurately the model identifies the right objects.
- **Recall:** Fraction of examples of true positive detection of objects compared to the number of examples of all actual objects, or the capability of the model to find all the cases of relevance.

All these metrics will give an effective framework to assess both accuracy and completeness of the detection system.

4. Data Preparation

4.1. Dataset Overview

The dataset has been carefully selected to include those vehicles which are in aerospace positions, in top-down views, a situation prevalent with a drone camera or a high-mount camera in traffic monitoring. It is made up of 626 high-quality images, the vast majority of which is provided by PeXels, with a variety of vehicles, such as cars, trucks, and buses, taken during different light and weather conditions. The flexibility enhances the generalization process of the model to the actual traffic monitoring environment.

4.2. Data Annotation and Format

All the images in the data set were annotated manually by the use of the Roboflow annotation tool. The labels are in the usual YOLOv8 format: text files with normalized bounding box coordinates in the format class x_center y_center width height, with all coordinates normalized to 0-1 based on dimensions of the image. Bootstrapping: Photos with no vehicles were not annotated as it would cause label noise.

4.3. Data Split and Augmentation

To guarantee a sound model evaluation and avoiding overfitting, the dataset was split in 536 training and 90 validation images, with the ratio of 85:15. The training sub-sample was synthetic augmented by geometric transformations, colour jittering, and random crops to artificially expand the variation of the data set and approximates the differences that occur in the real world such as occlusions, shadows and different viewpoints. None of the data augmentation of validation occurred to not bias the model performance measure.

4.4. Dataset Configuration

The directory structure of the dataset was carefully designed to enable easy training of the model:

- The 536 images of the training and the 536 label files are also stored in the train directory, in subfolders of images and labels.
- The valid directory: The valid directory is in which there are 90 validation images and label files to them.
- data.yaml configuration file contains parameters: dataset paths, number of classes (1 on Vehicle), and the name of classes. This file is important to provide the training pipeline of YOLOv8.
- All images were brought to a constant 640 x 640 pixels, as this is the input size requirement of YOLOv8, to normalize the training process and have the highest possible inference rate.

5. Model Fine-tuning

5.1. Transfer Learning Approach

Using transfer learning, we loaded our YOLOv8 model with weights having gone through the COCO training with a wide variety of 80 object classes. Though this gives a solid background knowledge of the general object characteristics, this general training may lead to worse performance of a particular task such as vehicle detection in aerial perspectives. Tuning our domain-specific dataset allows the model to modify its parameters, based on visual properties specific to our aerial vehicles (e.g. smaller scale, different orientations, and complicated backgrounds), increasing detection accuracy and recall, [11].

5.2. Training Protocol

We used a batch size of 32 and an input image size of 640×640 pixels. The 100 epochs of training were carried out using a Tesla P100 on 16GB of VRAM, striking a balance between comprehensive learning process and viable resource requirements: The learning rate was initialized to 0.0001 and gradually reduced to 0.00001 in successive epochs with the help of a scheduler to refine weight updates in subsequent epochs. A dropout of 0.1 was used to help prevent overfitting by randomly shooting neurons during training to help the model to create stronger features. Early stopping was implemented with a patient threshold of 50 epochs to stop training in case of no increase in the validation metrics, which saved on the use of computational resource.

The overall training time (0.21 hours or about 12.5 minutes) indicated the efficiency of the transfer learning and the computational feasibility of working with the fine-tuning of the YOLOv8 in real-life conditions.

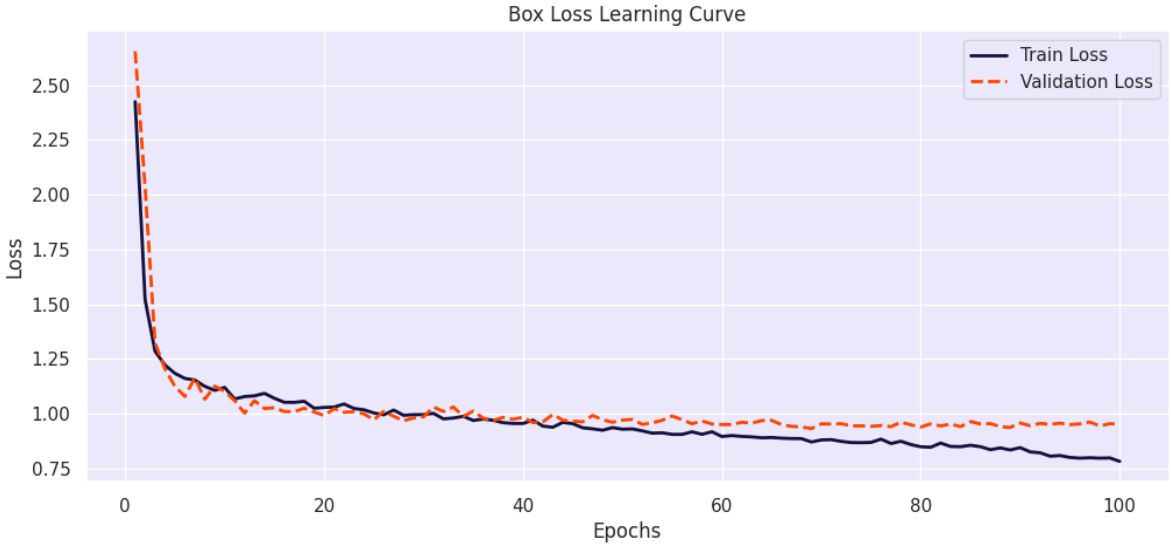

Figure 2: Box Loss Learning Curve

6. Results and Analysis

6.1. Quantitative Performance

Fine-tuned YOLOv8 was tested on the validation set of 90 images with 937 vehicle instances. Performance metrics indicate:

- Precision: 91.6, a high precision of the model, showing that the model is very accurate and does not have many false positives.
- Recall: 93.8%, which means that it detects most of vehicles in the images.
- As illustrated in the table above, mAP50: 97.5% indicates that it can be detected almost perfectly at a single 50th IoU.
- mAP50-95: 74.2%, showing continuous detection accuracy at higher levels of IoU, which demands fine localization of bounding box.

Table 2: Quantitative Results

Metric	Value
Precision	91.6%
Recall	93.8%
mAP50	97.5%
mAP50-95	74.2%

The outcomes highlighted in table 2 above emphasize the importance of fine-tuning YOLOv8 using domain-specific data in the event of aerial vehicles detection activities.

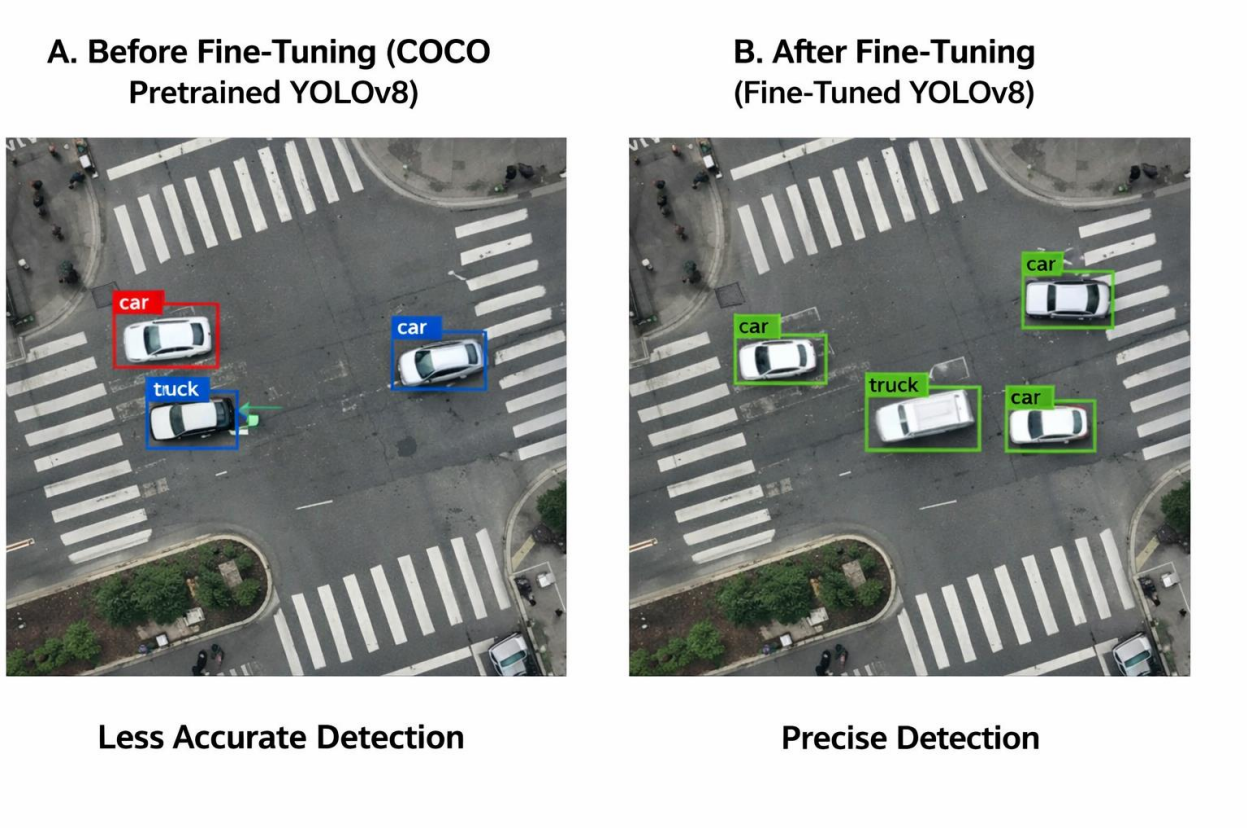

Figure 3: Qualitative comparison of vehicle detection results before and after fine-tuning YOLOv8

By the results depicted in above figure 3, it could be clearly concluded that fine-tuned model has better capability of classification as compared to pre-trained model.

6.2. Loss Convergence and Overfitting

The training and validation loss curves of bounding box regression, classification and distribution focal loss all had a happy smooth reduction pattern with negligible difference between model training and validation losses. This behavior can be interpreted as the convergence of the models when operating in a steady manner, and points out on the fact that the model fine-tuned shows no overfitting, and thus supports the generalization capability of the fine-tuned model to previously unknown data.

6.3. Inference Efficiency

The Tesla P100 GPU was able to execute the model at a rate of about 4 milliseconds per image, which validated the appropriateness of the model in real-time traffic surveillance systems, where it is necessary to process video streams at high speed.

6.4. Experiment Monitoring with Weights & Biases

The existing integration with the Weights and biases gave a real-time visualization and logging of important training parameters such as learning rates, losses, and accuracy metrics. This continuous observation enabled the identification of training abnormalities early and informed the hyperparameter tuning to enable effective and reproducible model creation.

Figure 4: Vehicle detection probability-based prediction

7. Discussion

In comparison to the baseline COCO-trained YOLOv8 model, it was found that upon fine-tuning, the vehicle detection accuracy had improved significantly. This domain-specialized training enabled the model to more effectively identify harder-to-detect vehicle details and also customize to the high-view angle, minimizing false alarm rates and missed detections.

Nevertheless, the weak mAP50-95 measure observed suggests that there is still room to improve. It can be improved by adding more vehicle types to the dataset, using multi-scale training schemes, or considering ensemble methods, to achieve higher accuracy at high levels of IoU.

Further comparison with the studies that are available further underlines the effectiveness of the proposed approach. Previous findings have revealed that even the state of the vehicles detecting algorithm such as faster R-CNN, SSD and the older versions of the YOLO cannot achieve the highest level of accuracy and real-time performance in a top view with smaller object sizes. In opposition, the fine-tuned YOLOv8 model applied in this study allows high accuracy and recall at high-inference rate.

However, unlike some other more recent publications, which provide bigger samples or modify the architecture to further boost the performance, the current work thinks about the ways of optimizing a typical YOLOv8 model with just a relatively small domain-specific dataset. Though this is computationally

efficient and very practical implementation, it may limit the ability to directly compare the performance with large-scale benchmark schemes. This trade-off indicates the need to trade datasets scale, intricacy of the models, and the restrictions of application of the system to actual traffic monitoring.

8. Limitations

In spite of the promising performance achieved, there are several limitations that should be admitted. To start with, dataset adopted in this work is quite limited (626 images) and this fact might not allow the model to be applicable to more heterogeneous real-world situations including various geographic areas, traffic volumes, and weather conditions. Second, there is one single class (Vehicle) in the dataset, which limits the ability of the model to identify various types of vehicles at a fine level of detail. Also, the images have been downsized to a constant size of 640x640 pixels, which can cause fine details to be lost, especially of small or distant cars in an aerial image. The model was as well tested on a validation set based on the same data distribution and therefore might perform differently on entirely unknown datasets. Lastly, even though the creation of a high-performance, it is possible to perform real-time inference on a resource-constrained edge device, which might necessitate additional optimization and model compression methods.

9. CONCLUSION AND FUTURE WORK

This work shows that fine-tuning the YOLOv8 model on a carefully chosen dataset of aerial vehicles greatly improves its ability to recognize things in traffic surveillance situations. The model had good precision, recall, and mean Average Precision scores, and it could be used in real time since it could make inferences quickly.

The primary contribution of this article is the thorough presentation of the adaptation of YOLOv8 to top-view vehicle detection by a small yet well-selected dataset. Unlike other previous works which rely on large or very complex models, the present study emphasizes on a balanced solution which will provide high-detection-accuracy as well as be able to operate in real time. Additionally, the study provides a certain amount of practical learning about the preparation of data sets, fine-tuning techniques, and the performance evaluation that will be selected to adapt to the specifics of the aerial traffic monitoring scenario.

In practice, the proposed plan will be implemented successfully in real traffic management systems, including smart cameras monitoring the city, classifying traffic flows and identifying traffic incidents with the assistance of flying platforms, i.e., drones or overhead cameras. Its ability to work very accurately in detection and real time properties makes it suitable in the implementation of smart transportation systems. Future research directions would include diversifying the environments of interest, extending the resilience of the model to challenging environments such as occlusion and low visibility as well as extrapolating the model to operate on edge devices. Additionally, the system can also be expanded with vehicle tracking as well as multi-class classification in order to assist the system in the future in the context on complex traffic management.

Future investigations will involve the inclusion of additional types of vehicles and environmental factors into the data used and the model will become more edge device adaptable by incorporation of time-related data to trace the vehicles on video streams. Further studies of the multi-task learning models incorporating detection and segmentation could contribute to stabilizing things in case of adverse conditions.

Funding Statement: This work has been done without any dedicated funding support.

Conflicts of Interest: No conflicts of interest.

Data Availability: The dataset comprises 626 aerial vehicle images collected mainly from Pexels and is available upon request.

References

[1] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149. 10.1109/TPAMI.2016.2577031

[2] Ramos, L. T., & Sappa, A. D. (2025). A decade of you only look once (yolo) for object detection: A review. IEEE Access, 13, 192747-192794. 10.1109/ACCESS.2025.3630988

[3] Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE, 105(10), 1865-1883. 10.1109/JPROC.2017.2675998

[4] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587). 10.1109/CVPR.2014.81

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

[6] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271). https://doi.org/10.48550/arXiv.1612.08242

[7] 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 (pp. 779-788). https://doi.org/10.48550/arXiv.1506.02640

[8] Wu, Y., Guan, X., Zhao, B., Ni, L., & Huang, M. (2023). Vehicle detection based on adaptive multimodal feature fusion and cross-modal vehicle index using RGB-T images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 8166-8177. 10.1109/JSTARS.2023.3294624

[9] Dikbayir, H. S., & BÜLBÜL, H. İ. (2020, December). Deep learning based vehicle detection from aerial images. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 956-960). IEEE. 10.1109/ICMLA51294.2020.00155

[10] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29. https://doi.org/10.48550/arXiv.1605.06409

[11] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359. 10.1109/TKDE.2009.191

[12] Widayani, A., Putra, A. M., Maghriebi, A. R., Adi, D. Z. C., & Ridho, M. H. F. (2024). Review of application YOLOv8 in medical imaging. Physics Letters, 5(1). 10.20473/iapl.v5i1.57001

[13] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136. https://doi.org/10.1007/s11263-014-0733-5

[14] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-10602-1_48