Baltic J. Modern Computing, Vol. 12 (2024), No. 1, 110-124
https://doi.org/10.22364/bjmc.2024.12.1.07

UAV-based Traffic Intensity Analysis Framework: A Case Study on Pedestrian Crossings

Julius GUDAUSKAS, Gintarė PETKUTĖ, Kristupas TRAKŠELIS, Andrius KRIŠČIŪNAS

Faculty of Informatics, Kaunas University of Technology, Studentu 50, Kaunas, LT-51368, Lithuania

julgud@ktu.lt, ginpet7@ktu.lt, kritra1@ktu.lt, andrius.krisciunas@ktu.lt

ORCID 0009-0000-9966-1703, ORCID 0009-0001-7439-6170, ORCID 0009-0009-2795-8845, ORCID 0000-0002-3691-0423

Abstract. In this research, a novel framework is presented for traffic monitoring and analysis that leverages drone footage and georeferencing technology. The study focuses on the analysis of various pedestrian crossing configurations, including standard zebra crossings, crossings with islands, and crossings with speed bumps, with a particular emphasis on their impact on vehicle and pedestrian behaviour. High-resolution drone footage was captured along selected roads in Kaunas, Lithuania, enabling a comprehensive examination of the interactions between vehicles and pedestrians at these diverse crossings. The proposed framework incorporates advanced object detection models, such as Faster R-CNN, to effectively identify and analyse vehicular movements. The experimental validation of the framework is demonstrated through a specific case study involving pedestrian crossing analysis.

Keywords: vehicle detection, vehicle tracking, traffic flow analysis, convolutional neural network, UAV drone data, pedestrian crossing analysis

1. Introduction

In urban environments, pedestrian crossings play a crucial role in facilitating safe interaction between pedestrians and vehicular traffic. The effective design and placement of pedestrian crossings are essential for ensuring the safety of both pedestrians and drivers. However, despite their widespread use, there is a gap in the literature regarding studies that investigate the driver's behaviour when approaching pedestrian crossings of different types and various settings.

To understand the dynamics between pedestrian crossings and traffic intensity, it becomes necessary to acquire vast amounts of data. Traditionally, the acquisition of traffic data has relied on methods such as manual counters, vehicle fixed detector and probe vehicles (Brahimi et al., 2020). While these approaches have provided valuable insights, they come with inherent limitations. For instance, manual counting does not perform well when vehicles go at higher speed. Stationary detectors often offer restricted

spatial coverage and are susceptible to adverse weather conditions (Barmpounakis et al., 2016). Additionally, fixed camera devices face a problem of occlusion where vehicles are fully or particularly hidden behind other object like trees, road signs, another vehicle, etc. Such problems can be solved by installing more cameras however such approach drastically increases expenses (Muhammad et al., 2018). Meanwhile probe vehicles always require good signal coverage and road network for data acquisition (Serrone et al., 2023). The introduction of innovative technologies, notably drones, has transformed data collection methodologies, offering a more efficient and cost-effective means of gathering comprehensive traffic data. Drones enable an agile and expansive view of traffic patterns, overcoming the limitations posed by traditional methods (Coifman et al., 2006 and Serrone et al., 2023). This need for huge amounts of data required for traffic analysis forms the core of our study, motivating the utilization of drones to efficiently gather significant datasets without losing analysis accuracy or requiring costly equipment.

Table 1. Drone strengths and weaknesses

Strengths	Weaknesses
- Efficiency: Provides swift data collection across large areas. - Versatility: Can capture diverse traffic scenarios and environments. - Cost-Effectiveness: Offers relatively lower operational costs compared to traditional methods. - High Resolution Imaging: Captures detailed and high-resolution traffic data. - Non-Intrusive: Minimally disrupts traffic flow during data collection. - Real-Time Monitoring: Allows for real-time data collection and analysis.	- Regulatory Constraints: Drone application is conditioned by regulatory issues. - Weather Sensitivity: Susceptible to adverse weather conditions affecting flight operations. - Limited Flight Time: Restricted flight duration per battery charge.

This paper addresses the gap of studies when investigating drivers' behaviour by presenting a case study that analyses three different types of pedestrian crossings - standard zebra crossings, crossings with islands, crossings with slowing curbs, and the driver's behaviour while approaching such crossings. Our research aims to contribute valuable insights into the dynamic interactions between pedestrian crossings and vehicle traffic in urban environments. To achieve this, drone technology is being utilized to capture real-time data from selected busy roads in of city Kaunas, Lithuania. Drones were chosen as the primary data collection method due to above-mentioned strengths over traditional methods. The methodology involves the use of a small drone equipped with a camera to film selected busy roads. The drone footage will capture three types of pedestrian crossing.

The dataset generated for this research is analysed using computer vision techniques, specifically utilizing Faster R-CNN (Region-based Convolutional Neural Network) for object detection, allowing to detect and identify vehicles accurately. For vehicle tracking centroid tracking methods is implemented to monitor the movement of detected objects over time. This approach enables the calculation of each vehicle’s speed, and the observation of how pedestrian crossings affect vehicular behaviour. The analysis will extend to examining whether different type of pedestrian crossings impact vehicle speed and overall traffic intensity.

2. Related Work

The use of drones for traffic monitoring has been explored in several studies. A systematic review of drone-based traffic monitoring systems from a deep learning perspective was conducted by Bisio et al. (2022). This work focused on vehicle detection, tracking, and counting, which are fundamental building blocks towards founding solutions for traffic congestion, flow rate, and vehicle speed estimation. Another study by Butilă and Boboc (2022) examined the application of Unmanned Aerial Vehicles (UAVs) for civil engineering, especially those related to traffic monitoring. The study concluded that this field is still in its early stages and that progress in advanced image processing techniques and technologies used in the construction of UAVs will lead to an increase in the number of applications.

The use of Faster R-CNN for object detection in traffic analysis has also been explored. For instance, a study by Wang and Peng (2019) designed and implemented an object detection system using a Faster R-CNN method that shares full-image convolutional features with a detection network. Another study by Datta et al. (2020) used Faster R-CNN on their data to analyse the environment around the road and the environment around the car. As for vehicle tracking, a real-time wrong-way vehicle detection system was proposed, which uses the You Only Look Once (YOLO) algorithm to detect vehicles from the video frame and then tracks each vehicle in a specified region of interest using a centroid tracking algorithm (Rahman et al., 2020). Another study by Bakliwal et al. (2020) presented an algorithm for centroid-based tracking of moving objects.

3. Methodology

3.1. General workflow

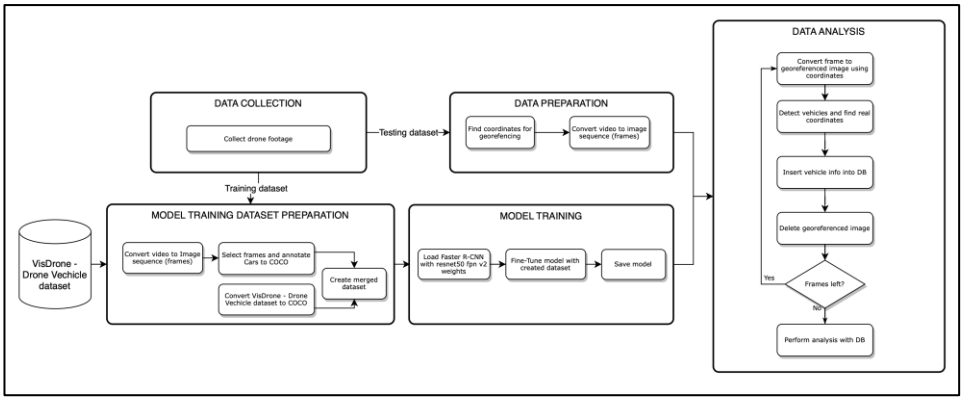

Figure 1. General workflow diagram

To conduct the analysis of traffic flow, real life data of city streets is first collected. The data is then used for prediction model training and testing. Open-source VisDrone vehicle dataset was used together with our collected data to finetune a CNN model. For testing and further analysis, data is georeferenced and split to frames. During the data

analysis stage, frames are converted to georeferenced images, vehicles are located, and a database is populated with vehicle locations at every time step.

3.2. Model training dataset preparation

The training dataset was constructed by combining manually created dataset from drone footage and open-source VisDrone – DroneVehicle (Sun et al., 2022) dataset. It provides a diverse set of images, incorporating a wide range of cars and scenes, enhancing the model's ability to learn and generalize across various scenarios.

A custom dataset consisting of a total of 188 images from various frames, has been manually annotated with the aid of the open-source tool, Label Studio. This annotation process establishes a robust dataset, providing the model with essential information for effective learning and generalization. The inclusion of diverse frames enhances the representativeness of the dataset, allowing the model to adapt to various scenarios. This systematic dataset preparation is crucial for fostering the accuracy and performance of the model during the training phase, enabling it to make informed predictions on new and unseen data.

To enhance the accuracy of the model, we incorporated an extra open-source dataset. The DroneVehicle dataset includes a total of 56,878 images captured by the drone, with almost an even split between RGB and infrared images. The dataset provides detailed annotations for multiple vehicle types. Model training dataset was extended with RGB images from the DroneVehicle dataset.

3.3. Prediction model selection

In the selection process of a machine learning model, consideration is given to two pivotal aspects: the model's efficacy in accurately discerning objects within a given dataset and its computational efficiency. In the landscape of object detection models based on CNNs, which can be broadly categorized into the two-stage and one-stage approaches, the YOLO (You Only Look Once) model stands out as a representative of the single-stage network. The model employs a basic CNN to directly forecast class probabilities and bounding boxes from the input image (Redmon et al., 2015). It is considered as one of the most common choices in production only because of its simple architectural design, low complexity, and easy implementation (Diwan et al., 2023).

In the statistical Analysis of various YOLO-Based deep learning models study (Sirisha et al., 2023) authors provide a detailed understanding of YOLO algorithm strengths and weaknesses. While YOLO is recognized for its speed and accuracy in object detection, it encounters challenges when it comes to detecting small objects. There are several ways to improve the small objects accuracy (Hu et al., 2023 and Ji et al. 2023) but it requires additional architecture changes that reduces the speed.

R-CNN is a region based two-stage convolutional neural network object detection algorithm proposed by Girshick (2015). This model comprises two main modules: the first module is a CNN known as the Region Proposal Network (RPN). Its primary role is to generate region proposals. Operating on a single image input, it produces bounding boxes and object confidence scores. Following this, a ROI pooling layer combines the extracted feature maps with the bounding box proposals, allowing a classifier to generate outputs for both the object class and an associated bounding box that encapsulates it (Avola et al., 2021).

In the Faster R-CNN and YOLO analysis research (Dong, 2023) the results presented by the authors indicate that YOLO exhibits greater speed, yet Faster R-CNN demonstrates a superior ability to accurately detect small objects. Furthermore, through various experiments in detecting different objects in orthographic images using DOTA (Xia et al., 2018) and HRSC2016 (Liu et al. 2016) datasets it is observed that region-based convolutional neural networks yield high precision results and demonstrates a high level of accuracy in detecting small objects.

Given these criteria, our deliberation led to the selection of Faster R-CNN, a convolutional neural network (CNN) architecture known for its commendable balance between robust object detection capabilities and computational efficiency. The implementation of the car detection model entailed the utilization of Faster R-CNN architecture integrated with a ResNet-50 Feature Pyramid Network (FPN) v2 backbone, leveraging pre-trained weights. This feature pyramid network is chosen because it is designed to provide a better performance and increased accuracy in detecting smaller objects. The improvements are shown in the experiments with a significant 9.7% mAP increase compared to regular Faster R-CNN (Rath, 2022). The implementation is visualized in **Figure 2**.

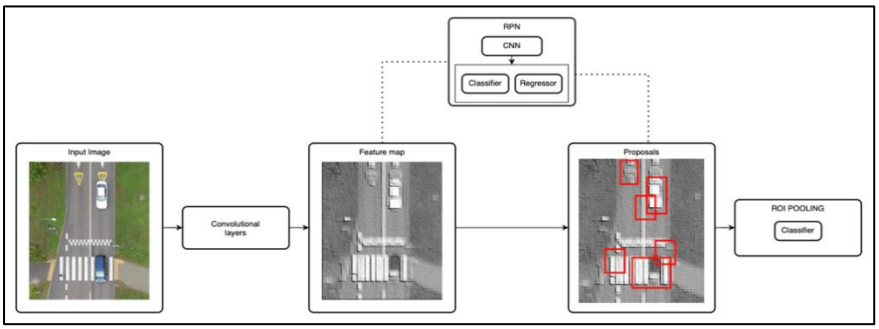

Figure 2. Faster R-CNN explanation

The model underwent fine-tuning process using a custom dataset created from drone footage images and the publicly available DroneVehicle dataset. Training ensued across 50 epochs, employing specific hyperparameters: a learning rate (lr) set at 0.001, a momentum (m) of 0.9, and a weight decay (wd) of 1e-6. The parameters generalized in **Table 2**.

Table 2. Parameters used in model training

Parameter	Value
FPN	ResNet-50 v2
Optimizer	Adam
Learning rate	0.001
Weight decay	1^{-6}
Batch size	8
Momentum	0.9

Epoch	50

3.4. Urban data collection with georeferencing

The data collection methodology for this research involves aerial filming using a DJI Mini Pro 3 drone positioned at an altitude of 80 meters above a selected urban street. The drone is strategically centred above a pedestrian crossing, with its camera oriented at a 90-degree angle to the street below. The chosen altitude provides an optimal perspective to observe and analyse vehicles and other elements within the urban environment, maintaining a 3840x2160 pixel resolution. The utilization of drone technology ensures a versatile and dynamic approach to data collection, enabling the acquisition of high-quality footage from an overhead vantage point that might be challenging to achieve through traditional ground-based methods.

For georeferencing, the methodology involves using a mapping tool to precisely establish the geographical coordinates of filmed frames. To georeference each frame, the four corner coordinates of the street are identified using the mapping tool. This approach leverages the detailed mapping data provided by the tool to accurately pinpoint the geographical location of specific points within the recorded footage. While drone GPS data is available for georeferencing purposes, findings indicate that utilizing this mapping tool yields superior accuracy. The reliance on the mapping tool enhances the precision of spatial referencing, allowing for a more detailed and reliable analysis of the vehicle movement captured in the video footage. This georeferencing methodology ensures the alignment of recorded frames with real-world geographical coordinates, facilitating a robust and geospatially accurate foundation for subsequent analyses and interpretations.

Figure 3**.** Coordinate selection for georeferencing using mapping tool

This research employs three types of urban crossings: regular crossings, crossings with islands, and crossings with speed bumps. Examples of specific locations used in this study are provided **Figure 4** and **Figure 5**.

Figure 4. Crossing in Location 1

Figure 5. Crossing in Location 3

3.5. Vehicle database

In the initial step, we extract the pixel coordinates of the located vehicles from the georeferenced images. Then, objects are organized into individual moving entities through the implementation of the centroid objects tracking technique. This technique entails taking an initial set of object detections, assigning unique IDs to each detection, and subsequently tracking these objects through the frames. The tracking process utilizes Euclidean distance calculations between the centres of detected bounding boxes to ensure accurate and continuous monitoring of object movements. The object is considered out of the frame when it's no longer detected for more than 2 frames.

The pixel values of these vehicles are then converted to real-world coordinates and inserted into a database. This straightforward process establishes a comprehensive dataset, providing essential spatial information for detailed analyses of vehicular dynamics within the urban environment under study. The resulting database serves as a data source for further analysis.

source_file	frame_id	timestamp	car_id	lat	long
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	0	54.90641401…	23.9452887142553
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	1	54.90577451…	23.945391568425
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	2	54.90613894…	23.9458624617387
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	3	54.90615607…	23.9458223764915
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	4	54.90618081…	23.9457786735017
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	5	54.90638527…	23.9453391065022
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	6	54.90609478…	23.9459566748771
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	7	54.90632203…	23.9454687724689
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	8	54.90627893…	23.94530351366
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	9	54.90611263…	23.9459170800663
DJI_0352.MP4	Frames-new/frame_1_1699365764128.jpg	16993657641…	10	54.90630721…	23.9455065485832

Figure 6. Database with vehicle info

3.6. Velocity calculation

Utilizing the pre-established database, which comprehensively captures the spatial coordinates of each vehicle at distinct timestamps, we have implemented a systematic approach for querying individual vehicles through SQL based on their unique identification (ID). The key metric of interest, vehicle velocity, is calculated by examining the changes in spatial positions between consecutive frames. This dynamic analysis of spatial changes enables us to derive the velocity (v) of each vehicle with precision. The velocity calculation is encapsulated by the following formula, where the vehicle's speed is determined by evaluating the spatial disparities over time:

$$v = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{\Delta t}$$

where:

- (x_1, y_1) and (x_2, y_2) are the real-world coordinates (latitude, longitude) of the vehicle in two consecutive frames.
- Δt is the time interval between the two frames.

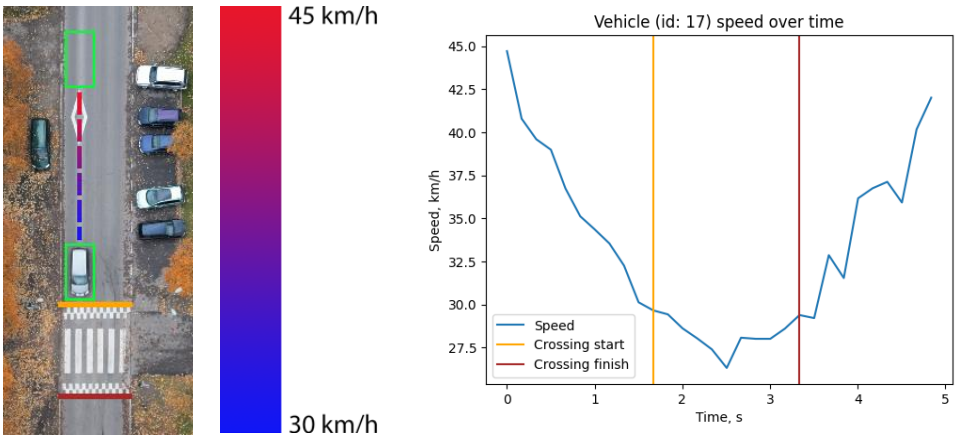

Figure 7. example procedure of velocity analysis. Vehicle movement is displayed in the (a) figure with its speed (b) before the crossing. Results are shown in the graph (c).

It is essential to note that real-world conditions can introduce errors in the position data. Factors such as drone position changes, vibrations, and the inherent limitations of the car detection algorithm may contribute to inaccuracies. High wind can lead to a greater deviation in velocity calculations from the actual speed. These sources of error should be considered when interpreting the velocity results. However, in this research, the data were gathered during periods of calm weather with minimal or no wind, and movement influence was not considered.

Results

3.7. Model performance results

In a comparative analysis between different FPN architectures (Figure **8**), ResNet50 v2 demonstrates a notable performance advantage over Mobile Net V3 Large and ResNet50 v1. The evaluation, based on total loss during validation, reveals that ResNet50 v2 outperforms its predecessors by approximately 13% compared to ResNet50 V1 and approximately 27% compared to Mobile Net. This improvement underscores the efficacy of the enhancements introduced in ResNet50 v2, which contribute to a more robust and accurate model, making it a preferable choice for detecting smaller objects more accurately.

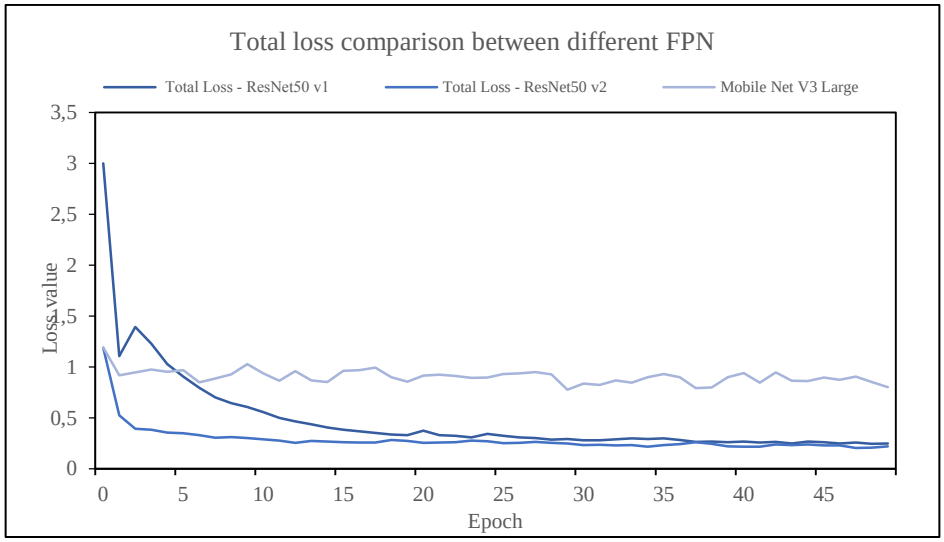

Figure 8. Total loss comparison between different FPN

The model was initially fine-tuned on the VisDrone dataset. However, to improve the model's performance on certain aspects, a custom annotated dataset was used for further enhance the model performance in certain situations - car detection on crosswalk, overlapping from trees etc. The accuracy difference when the model was trained with only VisDrone and using VisDrone and annotated dataset is displayed in Figure **9**.

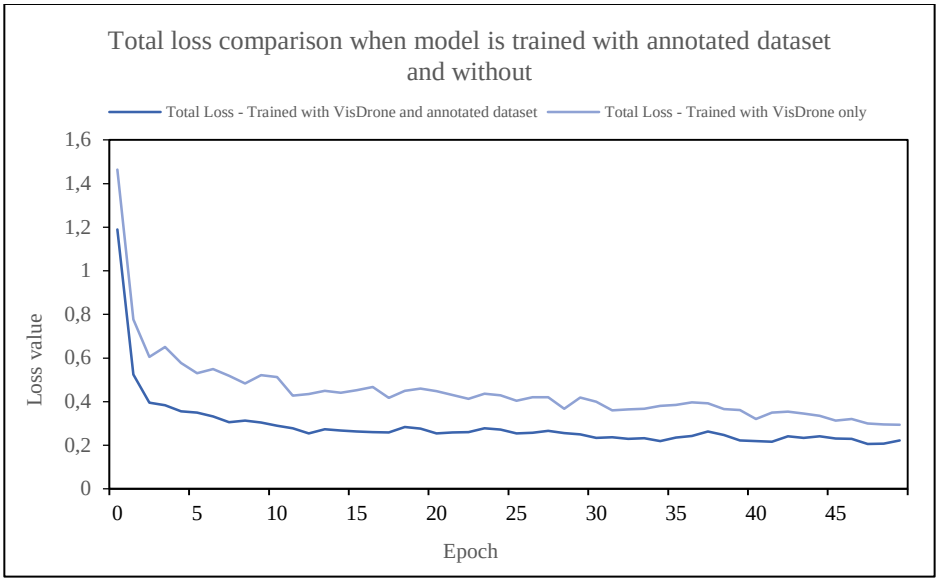

Figure 9. Total loss comparison when model trained with and without annotated dataset

During the training phase, the model performance was evaluated using different loss values:

- Total Loss: the overall error or discrepancy between predicted and true values, comprising all individual losses.
- Classifier Loss: measures the error in class predictions made by the model.
- Bounding Box Regression Loss: indicates the error in predicting the coordinates of bounding boxes around objects.
- RPN Box Loss: reflects the error in the region proposal network's bounding box predictions.
- Object Detection Loss: represents the error in detecting and classifying objects within the image.

In the training process over 50 epochs, the model has demonstrated commendable performance, as reflected in the consistently low and acceptable loss values across various components (**Figure 10**). The achieved Total Loss of 0.222 indicates robust learning and effective parameter adjustments. Notably, the Classifier Loss, Bounding Box Regression Loss, RPN Box Loss, and Object Detection Loss contribute collectively to the model's success. These results suggest that the model is successfully capturing intricate patterns in the data, showcasing its ability to predict class labels, bounding box coordinates, and handle region proposal intricacies. While these outcomes are promising, it is advisable to further assess the model's generalization by evaluating other available FNP architectures.

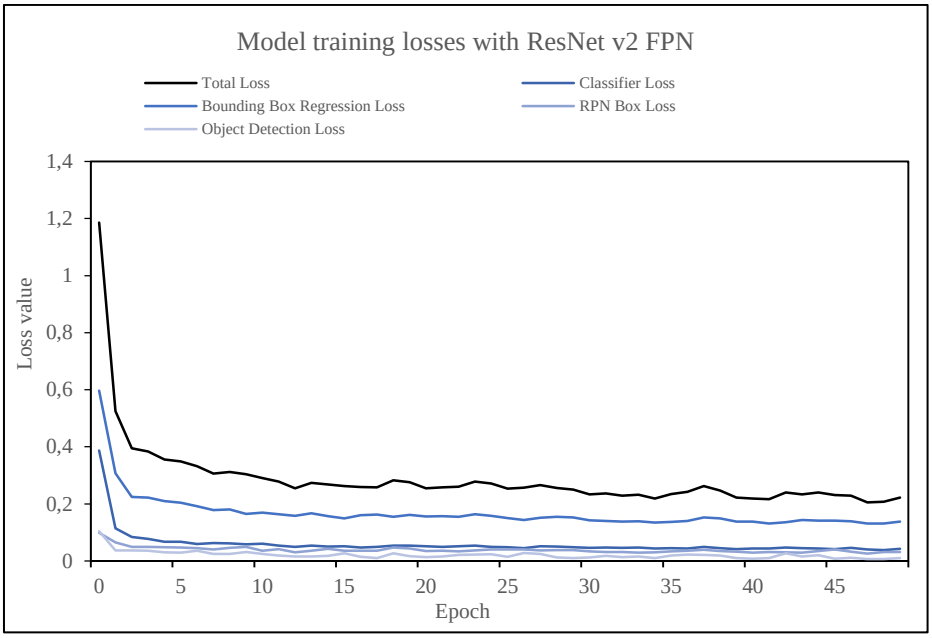

Figure 10. Prediction model losses over epochs

The pre-processing pipeline designed for training a car detection model encompasses a series of transformations. The augmentation sequence involves several operations, strategically chosen to enhance the diversity and robustness of the dataset. The introduction of a random-sized box crop, set with a probability of 0.4, plays a crucial role in maintaining the integrity of car bounding boxes during random cropping and resizing. This operation is essential for preserving accurate spatial information and ensuring that the model learns to detect cars across various scales. Additionally, Horizontal Flip and Vertical Flip operations, each with a 0.5 probability, are applied independently to introduce variations in viewpoint by flipping images horizontally and vertically. This variation aids the model in handling different perspectives, enhancing its ability to generalize effectively. The augmentation strategy also incorporates random brightness and contrast operations with probabilities of 0.2, contributing to the introduction of diverse lighting conditions and colour variations within the dataset. These transformations collectively aim to expose the model to a wide range of scenarios, improving its adaptability and robustness (Figure **11**).

Figure 11. Accuracy improvements after transformations are performed during the training

3.8. Data

Testing data was collected from various locations in Kaunas. This diverse sampling across the city aids in gaining a better understanding of vehicular behaviour in different locations. Average length of a video from a single location is about 8 minutes.

Table 3. Locations used for the testing

Location	Location image	Crossing type	Video length	Vehicle count
Location 1		With speed bump	10 min.	105
Location 2		With speed bump	10 min.	43
Location 3		With island	10 min.	73
Location 4		Regular	7min 24s.	55
Location 5		Regular	4min 29s.	29

3.9. Velocity analysis

The velocity calculation methodology developed earlier is employed, and the pre-established database is leveraged to perform velocity analysis for all the gathered locations. The analysis focuses on the key metric of interest, vehicle velocity, which is

derived by scrutinizing spatial position changes between consecutive frames. This approach is applied to the pedestrian crossing influence study, identifying specific crossing locations through Google Maps. This allows for the observation of fluctuations in vehicle speed over recorded periods. The automated analysis assesses speed variations both before and after the crossing, and the comprehensive results are presented in **Table 4**.

Table 4. Velocity analysis results

Location	Average vehicle speed whole distance	Average vehicle distance until crossing	Average vehicle distance after crossing	Speed difference until crossing
Location 1	31.87 km/h	33.9 km/h	32.7 km/h	-9.28 km/h
Location 2	26.10 km/h	27.8 km/h	27.05 km/h	-5.64 km/h
Location 3	43.06 km/h	42.36 km/h	44.37 km/h	-0.46 km/h
Location 4	32.66 km/h	33.05 km/h	33.87 km/h	-1.12km/h
Location 5	29.46 km/h	18.32 km/h	28.81 km/h	-7.3km/h

Velocity analysis for the whole distance was done for location 1, location 3 and location 4 crossings (Figure 12). It can be observed that the average speed of the cars decreases before the crossing with a bump and increases right after the crossing. Similar behaviour is evident with the regular crossings, where the analysed vehicles start to slow down once they enter the crossing. There is different behaviour caused by crossing with island. Vehicles usually increase velocity before the crossing and then slow down.

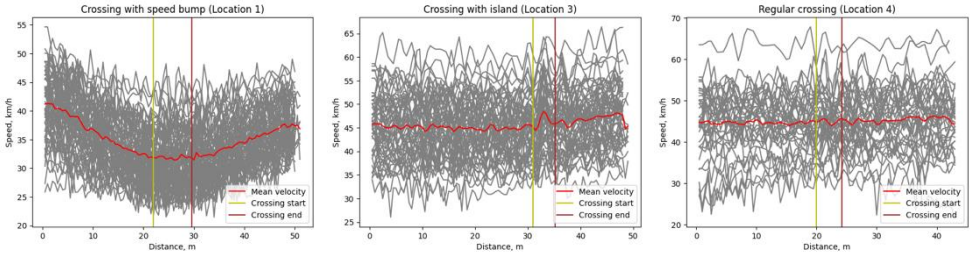

Figure 12. Speed distribution between different types of crossings

4. Discussion

In our proposed methodology, the use of drones to film urban traffic offers a promising approach to analyse vehicle traffic patterns. However, it is crucial to recognize a potential limitation affecting the accuracy of vehicle speeds derived from drone footage. The movement of the drone itself introduces a degree of variability that can impact velocity calculations. In addition to drone-related factors, real-world conditions should be considered also, including high wind, drone position changes, vibrations, as well as inherent limitations in the car detection algorithm. While data in this study were gathered during periods of calm weather with minimal or no wind, and movement influence was excluded from analysis, it is important to acknowledge these potential sources of error when interpreting the velocity results. Additionally, the reliance on manual coordinate selection via Google Maps, while offering a viable alternative for data collection, may

not always guarantee accuracy due to outdated satellite images. Despite this limitation, it stands out as a superior method when compared to the GPS system of the DJI Mini 3 drone. Also considering the potential impact of changing environmental conditions, it may be worthwhile to explore the effectiveness of augmenting the annotated dataset with additional images. It could also contribute to the stability of speed calculations in varying environmental conditions. For the further framework testing, more data could be gathered. It also has the potential to be tested in different scenarios.

5. Conclusions

The findings from this study have the potential to impact urban planning and traffic management strategies. Insights into how pedestrian crossings influence traffic intensity can inform the design and optimization of pedestrian infrastructure, leading to safer and more efficient urban transportation systems. Additionally, the study may contribute to the development of intelligent traffic management systems that could predict the intensity of the traffic in different environment settings.

Acknowledgements

The paper is developed based on Teaching Learning Sequence provided in the framework of the Erasmus+ programme's project STEAM-Active (Ref. number 2021-1-ES01-KA220-HED-000032107)

References

Avola, D., Cinque, L., Diko, A., Fagioli, A., Foresti, G. L., Mecca, A. (2021). MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images. *Remote Sensing, 13*(9).

Bakliwal, A., Puranik, A., Modi, A., Jain, A., Jaiswal, A., Godani, D., Jangd, P. (2020). Crowd Counter: an Application of Centroid Tracking Algorithm. *International Research Journal of Modernization in Engineering Technology and Science*, 1138-1141.

Barmpounakis, E. N., Vlahogianni, E. I., Golias, J. C. (2016). Extracting Kinematic Characteristics from Unmanned Aerial Vehicles. *Transportation Research Board 95th Annual Meeting*, (lpp. 16). Washington DC, United States.

Bisio, I., Garibotto, C., Haleem, H., Lavagetto, F., Sciarrone, A. (2022). A Systematic Review of Drone Based Road Traffic Monitoring System. *IEEE Access*, 101537-101555.

Brahimi, M., Karatzas, S., Theuriot, J., Christoforo, Z. (2020). Drones for Traffic Flow Analysis of Urban Roundabouts. *International Journal of Traffic and Transportation Engineering*, 9(3).

Butilă, E., Boboc, R. G. (2022). Urban Traffic Monitoring and Analysis Using Unmanned Aerial Vehicles (UAVs): A Systematic Literature Review. *Remote Sensing*, 620.

Coifman, B., McCord, M., Mishalani, R. G., Iswalt, M., Ji, Y. (2006). Roadway traffic monitoring from an unmanned aerial vehicle. *IEE Proc – Intell Transp Syst.*, 11-20.

Datta, A., Islam, M. T., Khatun, T., Hasan, B. M., Rahman, S. S., Mahfujur, R. M. (2020). Road Object Detection in Bangladesh using Faster R-CNN: A Deep Learning Approach. *IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, (lpp. 348-351). Bhubaneswar, India.

Diwan, T., Anirudh, G., Tembhurne, V. J. (2023). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications, 82*, 9243 - 9275.

Dong, W. (2023). Faster R-CNN and YOLOv3: a general analysis between popular object detection networks. *Journal of Physics: Conference Series, 2580*(1).

Hu, M., Li, Z., Yu, J., Wan, X., Tan, H., Lin, Z. (2023). Efficient-Lightweight YOLO: Improving Small Object Detection in YOLO for Aerial Images. *Sensors, 23*(14).

Ji, S.-J., Ling, Q.-H., Han, F. (2023). An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. *Computers and Electrical Engineering, 105*(108490).

Liu, Z., Wang, H., Weng, L., Yang, Y. (2016). Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE geoscience and remote sensing letters*, 1074-1078.

Muhammad, A. K., Wim, E., Tom, B., Yassine, R., Ansar-ul-Haque, Y., Davy, J., Geert, W. (2018). Unmanned Aerial Vehicle-based Traffic Analysis: A Case Study to Analyze Traffic Streams at Urban Roundabouts. *Procedia Computer Science*, 636-643.

Rahman, Z., Ami, A. M., Ullah, M. A. (2020). A Real-Time Wrong-Way Vehicle Detection Based on YOLO and Centroid Tracking. *Department of Electrical and Electronic Engineering Chittagong University of Engineering and Technology*, 5-7.

Rath, S. R. (2022). Object Detection using PyTorch Faster RCNN ResNet50 FPN V2. *Debugger Cafe*.

Redmon, J., Divvala, S. K., Girshick, R. B., Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, 779 - 788.

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, 28*.

Serrone, G. D., Cantisani, G., Peluso, P. (2023). Speed data collection methods: a review. *Transportation Research Procedia*, 512–519.

Sirisha, U., Phani Praveen, S., Srinivasu, P. N., Barsocchi, P., Bhoi, A. K. (2023). Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. *International Journal of Computational Intelligence Systems*, 16:126.

Sun, Y., Cao, B., Zhu, P., Hu, Q. (2022). Drone-based RGB-Infrared Cross-Modality Vehicle Detection via Uncertainty-Aware Learning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wang, C., Peng, Z. (2019). Design and Implementation of an Object Detection System Using Faster R-CNN. *International Conference on Robots & Intelligent System (ICRIS)*, (lpp. 204-206). Haikou, China.

Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J. (2018). DOTA: A large-scale dataset for object detection in aerial images. *IEEE conference on computer vision and pattern recognition*, 3974-3983.

Received January 23, 2024, revised March 4, 2024, accepted March 8, 2024