VEHICLE TRACKING AND SPEED ESTIMATION FROM UNMANNED AERIAL VEHICLES USING SEGMENTATION-INITIALISED TRACKERS

S.M. Tilon[1], F. Nex[1]

[1]Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands - (s.m.tilon, f.nex)@utwente.nl

KEY WORDS: infrastructure monitoring, edge computation, vehicle tracking, segmentation, lightweight

ABSTRACT:

We propose an effective vehicle tracker and speed estimation method from Unmanned Aerial Vehicles (UAVs) videos that can be deployed on UAV-embedded edge devices. Our tracker uses segmentation-derived vehicle regions to initialise a MOSSE tracker. This enables road operators to make multipurpose use of segmentation outputs while still being able to track the vehicles across frames. The vehicle speed is estimated using flight parameters derived from the UAV's flight computer and the vehicle displacement across frames. We trained CABiNet on the UAVid urban segmentation benchmark dataset and finetuned it on a dataset collected at our study site. A mean Intersection over Union (mIoU) of 0.73 was obtained for the vehicle class. Our segmentation-initialised MOSSE tracker was evaluated on the VisDrone Multi-Object Tracking (MOT) benchmark dataset and compared against traditional methods that utilise object regions for tracker initialisation. Our approach yielded a Multi-Object Tracking Precision (MOTP) of 0.872 compared to 0.830 when using YOLOv4. Our vehicle speed estimations approach was evaluated using a privately collected ground truth vehicle speed dataset. Our approach yielded a Root Mean Square Error (RMSE) between 3.42 and 16.12 km/hr across different flight configurations. Finally, our approach was deployed on an NVIDIA Jetson Xavier NX edge device and could be executed at 8 Frames Per Second (FPS). The results indicate that our approach is a simple yet fast alternative to traditional tracking methods while producing multipurpose segmentation information.

1. INTRODUCTION

Vehicle speed estimation is essential in traffic management operations to monitor traffic flow and enforce speed limits. Traditional methods for speed estimation use fixed sensors such as inductive loops and magnetic systems (Gheorghiu et al., 2021), range sensors such as radar (Czyżewski et al., 2019) and lasers (Zhang et al., 2020) or video systems (Fernández Llorca et al., 2021). Using video data from traffic cameras is a popular way to estimate vehicle speed, as it can also be used for other traffic management tasks, such as recognising vehicle types or license plates. However, traffic cameras are limited to observing a relatively small area at a fixed location and often require expensive equipment installation. Unmanned Aerial Vehicles (UAVs) offer a low-cost solution to survey large areas. The utilisation of UAVs has become more prevalent in recent years due to technological advancements and the scientific and industrial testing of UAVs in various applications. This increased usage has made UAVs an essential tool for road operators to monitor their infrastructures and gain a deeper understanding of their surroundings (Nex et al., 2022). With the increased capability of UAVs in manoeuvrability and flight-time, the improved performance of deep learning algorithms and the nature of applications in which UAVs are being deployed, there is a trend in industry and research to reach holistic and real-time UAV-based monitoring by using (UAV-embedded) edge devices that are fit for deploying deep learning models and effective communication to a ground control station (Tilon et al., 2022).

Extensive research has been conducted towards vehicle tracking and speed estimation from UAV systems (Balamuralidhar et al., 2021; Biswas et al., 2019; Hossain & Lee, 2019; Li et al., 2019; Shan et al., 2021). Most methods use deep learning-derived object detections to initialise their trackers. While vehicle object detections are helpful when vehicle tracking is the single objective, segmentations are more advantageous for road operators because they can provide multiple insights, such as deriving the shape and size of assets. In addition, they enable operators to accomplish various tasks simultaneously, such as asset inventory and traffic monitoring.

Nowadays, operators require a mix of information products to acquire multiple insights or achieve multiple tasks. Segmentation annotations are needed to gain information about the shape or size of infrastructure assets, such as road surface damages. Object detections are required for vehicle speed estimations or asset identification. These information products can be gained by running several single-tasks in parallel or using multi-task methodologies, which is computationally expensive and, therefore, unfeasible to acquire using computationally constrained UAV-embedded edge devices (Chang et al., 2021). There is a need to reduce the required information products to achieve several tasks. As such, deriving vehicle speeds from segmentation products rather than object detections is a logical first step.

Therefore, this paper aimed to estimate vehicle speed from segmentation information. We present a simple but effective vehicle speed estimation and tracking approach that can operate in real-time on a UAV-embedded edge device while the tracker is initialised from segmented vehicle regions instead of bounding box detections.

2. RELATED WORK

2.1 Vehicle Tracking

Vehicle speed estimation is achieved by detection-and-tracking (Fernández Llorca et al., 2021). First, vehicles are detected across frames, after which a multi-object tracking (MOT) algorithm is applied to the detections. How vehicles are detected varies and depends on the type of tracker used. Feature-based methods track

features such as key points or optical flow, while others utilise regions or bounding boxes of the complete vehicle or vehicle parts such as license plates across frames (Fernández Llorca et al., 2021). Region-based trackers are more popular because they showed to better account for large image motions. Within the region-based tracker category, various trackers exist. Popular tracking algorithms that make use of filters are, for example, Simple Online Real-time Tracking (SORT), which uses Kalman filters, Minimum Output of Sum of Squared Error (MOSSE) that uses correlation filters or CSRT that uses a Discriminative Correlation Filter with Channel and Spatial Reliability (Bewley et al., 2016; Bolme et al., 2010; Lukezic et al., 2018). Other methods make use of deep learning-derived metrics. DeepSORT combines SORT with a deep learning association metric (Wojke et al., 2017). Generic Object Tracking Using Regression Network (GOTURN) learns the appearance and motion of vehicles in an offline manner, whereas other trackers learn the appearance of vehicles during runtime (Held et al., 2016; Wojke et al., 2017). As said earlier, trackers must be initialised with a region to start the tracking process. In most studies, these initialisations are derived from deep-learning vehicle object detectors. While this is a reasonable approach for scenarios where tracking is the single objective, this is not valid in a scenario where multiple objectives exist. Therefore, this paper proposes to use vehicle segmentations to initialise the vehicle trackers.

UAV-based vehicle tracking is more challenging than tracking from fixed camera systems due to the constantly changing background, the UAV speed, illumination variances or other extrinsic parameters, which are static for fixed camera systems. Vehicle tracking from UAVs is researched by many. Li et al. (2019) proposed a homography-based motion tracking method to compensate for dynamic background changes. Biswas et al. (2019) used a Faster R-Convolution Neural Net (CNN) to detect vehicles and CSRT to track them. Shan et al. (2021) used YOLOv3 to detect objects and DeepSORT. Balamuralidhar et al. (2021) proposed an efficient multi-task vehicle detection framework that could be embedded on an edge device in combination with the MOSSE tracker that could run in parallel on the Central Processing Unit (CPU) without putting strain on Graphic Processing Unit (GPU) resources that were required for the deep learning process. Hossain and Lee (2019) compared various GPU-embedded devices for onboard object detection and tracking using DeepSORT. They concluded that the distance to the object is the main factor influencing the performance of their designed system. Vehicle speed is estimated using a function variation based on prior information on vehicle size and in-flight metrics such as the Ground Sampling Distance (GSD) and time between frames and travelled distance. The most common issue in these studies is the difficulty in testing and validating the proposed method without a proper benchmark dataset. To overcome this, some have created their own validation dataset using real-world vehicles (Shan et al., 2021) or proxy objects such as bicycles (Balamuralidhar et al., 2021), or tested their approach in a simulated environment (Li et al., 2019).

2.2 On-the-edge Multi-Objective Deep Learning

UAV-embedded edge devices can process data instantly once they arrive or transmit the data to a remote workstation, where they are processed in real-time. Obtaining multiple information objects on an edge device in real-time is challenging for two reasons: the computational constraints of edge devices and the traditional single-objective way deep learning neural nets are traditionally constructed.

Edge devices are constrained in memory, and computational ability, which affects how the device can be utilised. Therefore, neural nets must be constrained in size, reflected in the number of parameters residing in the net. This often diminished the performance of the net. In addition, most CNNs are constructed in a single objective manner, where scene segmentation or object detection is the single main aim.

3. METHODOLOGY

3.1 Overview

Our approach is as follows. A segmentation CNN deployed on the UAV edge device delineates the road scene in the categories “Road”, “Vehicle” and “Background”. A tracker is initialised using the delineated vehicle regions. The vehicle is tracked across frames, and the vehicle speed is estimated using the vehicle's displacement between frames and flight parameters derived from the UAV board computer. More details are provided in the following sections.

3.2 Scene Segmentation

The segmentation CNN CABiNet is used to segment the UAV video into the categories “Road”, “Background” and “Vehicle” (Kumaar et al., 2021). It is a lightweight segmentation CNN and can efficiently segment a scene with high speed and low computational overhead while providing competitive performances on edge devices (Kumaar et al., 2021).

It contains a context branch that efficiently details global and local context and a shallow spatial branch that captures spatial information rapidly and effectively. It reached a competitive mean Intersection over Union (mIoU) of 75.9% on the CityScapes dataset on an NVIDIA Xavier NX with 8 Frames Per Second (FPS). The model’s performance is described in mIoU.

3.3 Vehicle Tracking

The MOSSE tracker is used in this study because it was shown to detect object locations at high computational speeds while functioning on CPU rather than GPU at comparative performance to deep learning-derived trackers such as DeepSORT or GOTURN (Balamuralidhar et al., 2021). This allows the GPU resources on the edge device to be used by CABiNet, speeding up the overall approach while keeping competitive performance. The vehicle annotations obtained using CABiNet are used to initialise the tracking algorithm. The contours and enclosing bounding box are extracted from the detected vehicle regions. To account for localisation drift and the appearance and disappearance of new vehicles in the frame, the tracker was reinitialised every 15th frame using new vehicle regions derived from CABiNet. A lower refresh rate resulted in more segmentation inference operations, lowering the overall execution speed, while a higher rate failed to track all vehicles. A refresh rate of 15 was observed to result in most of the vehicles being tracked from time of appearance until time of disappearance while achieving a reasonably fast execution time.

The tracking performance was analysed using the Multi-Object Tracking Precision (MOTP) metric (Milan et al., 2016). See Equation 1.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (1)$$

where c_t = Number of matches in frame t
$d_{t,i}$ = the bounding box overlap between the target i and the ground truth in frame t.

This metric is a notion of localisation accuracy and calculates the average overlap between the predicted object and the ground truth.

3.4 Vehicle Speed Estimation

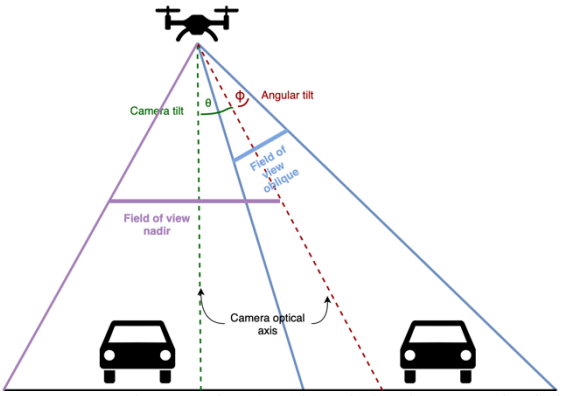

Figure 1. Diagram showing the relation between the field of view and the camera tilt during data acquisition.

Figure 1 is a schematic overview showing the difference between the field of view when acquiring a UAV video in a nadir or an oblique viewing angle. Using the camera tilt (θ) and angular tilt (Φ), the GSD can be calculated as follows:

$$GSD = \frac{H \times S_w}{F} \times \frac{1}{\cos(\theta + \Phi)'} \quad (2)$$

Where H = the flight height in meters
S_w = the camera's sensor pixel width
F = the camera's focal length
θ = the camera tilt in relation to the optical axis
Φ = the angular position of the pixels in the image in relation to the optical axis.

Figure 2 is a schematic overview that shows how vehicle and UAV displacement is calculated between frames. Using these parameters, vehicle speed (V) is calculated as follows (Balamuralidhar et al., 2021):

$$V = \frac{|\vec{D}_{UAV} - (\vec{D}_{vehicle} \times GSD)|}{f} \quad (3)$$

where $\vec{D}_{UAV}$ = is the UAV displacement in meters
$\vec{D}_{vehicle}$= the displacement of the vehicle in pixels
GSD = the ground sampling distance
f = time interval

The UAV displacement is derived from the UAV inflight speed log in real time. The vehicle's pixel displacement is derived by subtracting the object centres obtained from the tracking results. The time interval is derived from the FPS in which the video is recorded. The assumption is that the vehicle and the UAV travel in the same direction, parallel to the road.

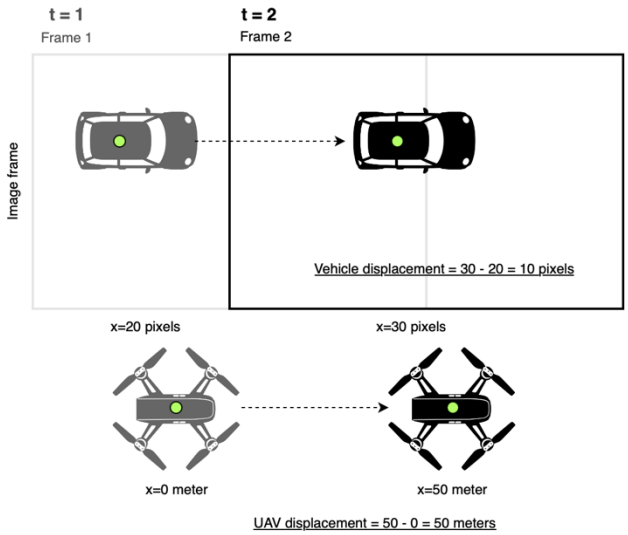

Figure 2. Schematic overview of how displacement is calculated.

3.5 Data

CABiNet is first trained using the UAVid dataset (Lyu et al., 2020). This dataset consists of 30 video sequences of realistic urban scenes and over 300 high-resolution images annotated with the eight classes "Buildings", "Road", "Static Car", "Moving Car", "Tree", "Low Vegetation", "Humans" and "Background" (Figure 3). Then CABiNet is finetuned on a UAV dataset collected on the University of Twente campus at a flying height of 50 meters in nadir. The dataset consists of 242 images with a resolution of 5472×3648 that are manually annotated with the classes "Background", "Vehicle" and "Road" (Figure 4).
The performance of our tracking algorithm is evaluated using the VisDrone dataset (Du et al., 2019). It is a vehicle detection and tracking benchmark dataset containing over 300 video sequences of urban scenes annotated with ten classes, of which the class "car" was used for testing purposes (Figure 5). The inflight parameters, such as UAV speed and flight height, are unknown for the VisDrone dataset. Therefore, reasonable static values were manually set based on the perceived flight direction, height, and camera angle. The FPS of the VisDrone dataset was 30 FPS.
A second dataset with vehicle speed ground truths was collected on the campus of the University of Twente. A vehicle was driven back and forth across a road while recording the per-second speed using a bike computer, the Wahoo Elemnt Bolt V2. It calculates speed from the Global Position System (GPS) derived displacement information and has an average speed error of 3.53% (Siddiqui et al., 2021). The vehicle aimed to drive at different average target speeds of 15, 30 and 50 km/hr. In the meantime, a Mavic Enterprise drone was flown at 50 m height above the road in 4 different configurations. It was positioned stationary and in motion, moving in the same direction as the vehicle, with the camera in nadir or oblique perspective at a 30-degree angle. The acquired videos are in 4K Ultra HD (3840×2160) resolution at 30 FPS. In total, 12 videos were collected, resulting in approximately 11 minutes of footage. An overview is given in Table 1 and an example video frame is shown in Figure 6.

Figure 3. Example of the UAVID dataset (Lyu et al., 2020).

Figure 4. Example of the campus dataset.

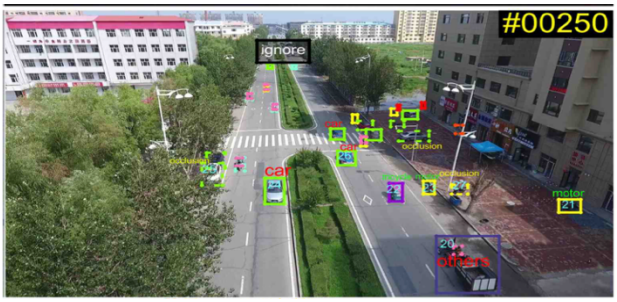

Figure 5. VisDrone Multi-Object Tracking annotations (Zhu et al., 2018).

Table 1. Configurations for ground truth data collection.

	Stationary	In motion	Stationary	In motion
Vehicle speed	**Nadir**	**Nadir**	**Oblique**	**Oblique**
15 km/hr	Flight 1	Flight 4	Flight 7	Flight 10
30 km/hr	Flight 2	Flight 5	Flight 8	Flight 11
50 km/hr	Flight 3	Flight 6	Flight 9	Flight 12

Figure 6. Example frame from Flight 5. The caption shows the vehicle speed, time and frame nr.

3.6 Training, deployment and testing

Our approach is applied and tested threefold to judge the performance of the segmentation, vehicle tracking and vehicle speed estimation method. First, CABiNet was tested by training it on the UAVid dataset. The hyperparameters for training CABiNet were found using grid-search. It was trained on an NVIDIA 2080ti for 100 epochs with a learning rate of 0.005, a gradient decay of 0.005 and a CrossEntropy optimiser. Then, the model was finetuned on the campus dataset for 20 epochs.
Second, the tracking algorithm was initialised by segmenting the VisDrone video frames using the CABiNet model that was trained on the UAVid dataset because this dataset represented the VisDrone scenes more closely. The tracking algorithm was evaluated using only the vehicle trajectory ground truth values. To assess the effect of tracker initialisation using segmented vehicle regions, we evaluated the performance of our tracker when the trackers were evaluated using object detections obtained using YOLOv4 or when the tracker was initialised using the ground truth values within the VisDrone dataset.
Finally, the vehicle speed approach was tested by segmenting the campus vehicle videos using the CABiNet model that was finetuned on the campus dataset by tracking the vehicle across frames and calculating the vehicle speed according to the approach explained in section 3.4.
Finally, the complete approach, including scene segmentation, vehicle tracking and speed estimation, was deployed on an NVIDIA Jetson Xavier NX. The trained CABiNet model was optimised for deployment on the edge device using NVIDIA Tensor RT optimisation library. This prunes weights and parameters not required for inference, resulting in a lightweight model that runs more efficiently on NVIDIA edge devices. The real-time performance of our approach was evaluated in terms of FPS.

4. RESULTS

4.1 Scene Segmentation

CABiNet reaches an average mIoU of 0.598 for the UAVid dataset (Table 2). These results align with the mIoU of 0.635 reported in the code repository of the original authors of CABiNet. The class “human” is the most significant influencer on the obtained average mIoU. Its low score can be explained by the scale in which persons are perceived compared to other objects in the dataset. Notably, “moving vehicles” are better detected than “static vehicles”. This could be explained by differences in headings, locations, illumination and shadows in which static cars are situated. In contrast, moving vehicles are less influenced by shadows from nearby objects while their background consists largely homogenous of the road class.

Table 2. Performance of CABiNet on the UAVid dataset.

Class	**mIoU**
Building	0.881
Road	0.705
Vehicle (static)	0.501
Vehicle (moving)	0.605
Tree	0.717
Low Vegetation	0.639
Human	0.160
Background	0.573
Average	**0.598**

After finetuning, CABiNet reaches an average mIoU of 0.869 on the campus dataset (Table 3). The “Vehicle” class reaches a mIoU of 0.733. Example output is shown in Figure 7. Although

the shape of vehicles is not always precisely delineated, the model performs sufficiently enough not to miss vehicle instances.

Table 3. Performance of CABiNet when finetuned on the campus dataset.

Class	**mIoU**
Background	0.983
Road	0.890
Vehicle	0.733
Average	**0.869**

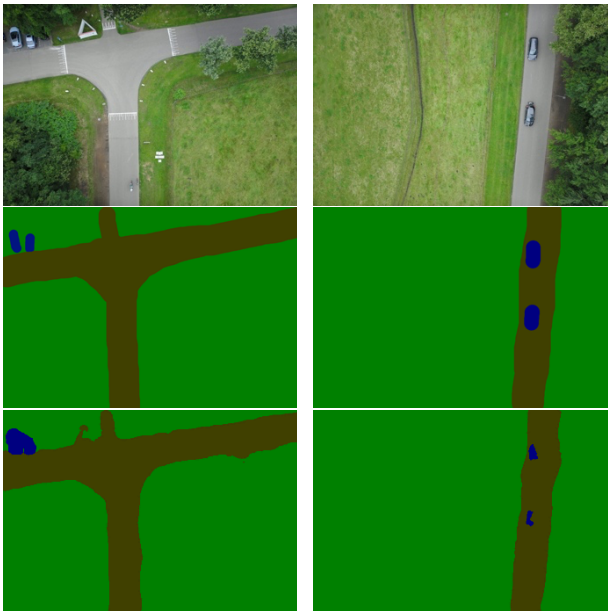

Figure 7. Original image (top), ground truth (middle), and segmentation predictions (bottom) for the campus dataset.

4.2 Vehicle Tracking

The segmentation initialised MOSSE tracker reaches 0.872 MOTP at 8 FPS when deployed on the NVIDIA Xavier NX (Table 4). We compare this value against a traditional approach where the tracker is initialised from object detections obtained using a YOLOv4 model. YOLOv4 is trained on the bounding boxes within the VisDrone dataset, reaching a mean Average Precision (mAP) at 50% of 0.856 (Bochkovskiy et al., 2020; Wang et al., 2022). When using YOLOv4 obtained object detections for tracker initialisation, a MOTP of 0.830 is reached at 9 FPS. When the MOSSE tracker is initialised with the regions obtained from the ground truth files, a MOTP of 0.842 is obtained at 9 FPS. These results show that our proposed method reaches a higher MOTP at comparable speeds compared to traditional object detection initialised MOSSE tracker while providing valuable semantic information that can be further utilised in other road operator tasks.

Table 4. Multi-Object Tracking Precision (MOTP) and FPS for our proposed method compared to object detection and ground truth initialised MOSSE tracker.

	MOTP	FPS
CABiNet (segmentation)	0.872	8
YOLOv4 (object detection)	0.830	9
Ground truth	0.842	9

Figure 8 shows a screenshot of the tracking results on the VisDrone dataset. The segmentation output is overlaid on the video frame. The tracked vehicles are annotated with green boxes, while the ground truth is annotated with black boxes. Parked vehicles are not tracked, which can be explained by false negatives in the segmentation step. Vehicles that just have entered the frame, like the vehicle at the bottom centre, are also not tracked, which can be explained by the tracker re-initialisation rate of 15 frames.

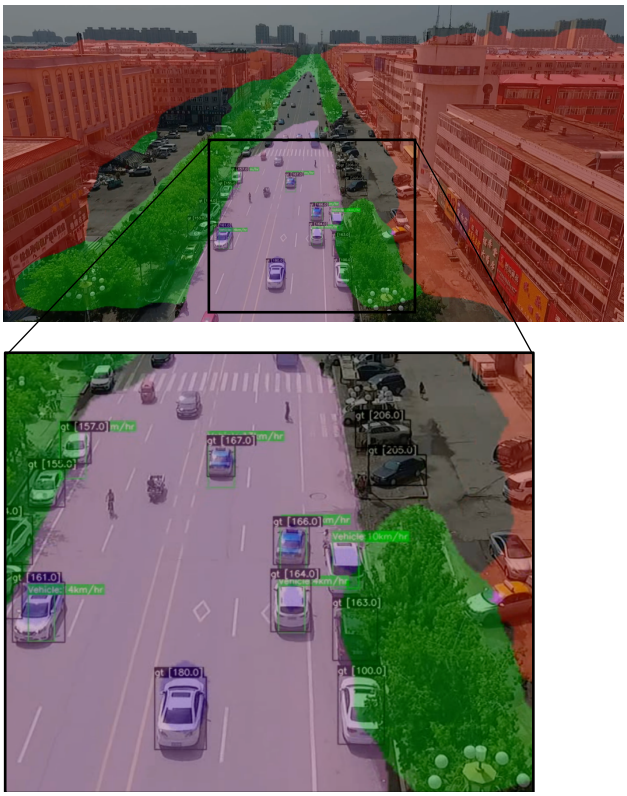

Figure 8. Tracked vehicles (green boxes) versus the ground truth (gt; black boxes) in the VisDrone dataset.

4.3 Vehicle Speed Estimation

Table 5 shows the average ground truth speed, average predicted speed, and the Root Mean Square Error (RMSE) obtained for the campus dataset acquired in different flight configurations. For the flight configuration where the UAV was stationary with an oblique viewing angle and the vehicle was driving at 50 km/hr, our segmentation model failed to recognise the vehicle, and no tracking and speed measurements could be provided. For the other flight configurations, the RMSE is lower when the vehicle is driving at lower speeds. A possible explanation is that the vehicle appears in more frames, leading to a better ability to track the vehicle across frames and estimate the vehicle displacement. In addition, a lower RMSE is obtained when the UAV is stationary compared to when the UAV is in motion. Less UAV motion could reduce the accumulation of potential erroneous or drifting GPS values obtained from the UAV board computer, leading to erroneous UAV displacement values. Since the segmentation model was finetuned on data in the nadir viewing angle, it was expected that our approach would have difficulties segmenting and tracking vehicles in oblique mode. Missing vehicle segmentation was the main reason for missed tracking and speed estimations in the stationary and oblique flight configuration (Flight 9). However, when vehicle tracking works,

the RMSE for speed estimation varies equally across the two viewing angles.

Table 5. Average ground truth and predicted speed, and the Root Mean Square Error (RMSE) in km/hr for the different flight configurations.

	Parameter	Stationary Nadir	Motion Nadir	Stationary Oblique	Motion Oblique
15 km/hr	Avg. speed gt.	14.68	16.56	16.58	18.11
	Avg. speed pred.	17.97	11.37	26.57	8.29
	RMSE	4.80	6.03	3.42	10.19
30 km/hr	Avg. speed gt.	28.55	27.09	29.27	28.9
	Avg. speed pred.	22.83	17.18	35.98	13.92
	RMSE	6.43	6.48	5.15	11.46
50 km/hr	Avg. speed gt.	42.09	39.46	43.11	39.66
	Avg. speed pred.	44.50	21.48	NaN	17.2
	RMSE	5.83	16.12	NaN	7.04

Figure 9 depicts speed estimations, overlaid on screenshots of the videos recorded during Flight 5 and Flight 12.

Figure 9. Speed estimations and ground truths overlaid on Flight 5 (top) and Flight 12 (bottom) video frames.

5. DISCUSSION

We proposed a simple and efficient method to detect vehicle speeds from segmentation-initialised trackers. The disadvantage of using segmentations and maximum enclosed bounding boxes instead of directly using bounding boxes retrieved by object detection methods is that segmentations are prone to be inaccurate at the edges of the objects due to illumination differences or overlapping objects. Therefore, the minimum enclosing bounding box rarely represents the entirety of the object (Fernández Llorca et al., 2021), nor does the centre of the box, used to calculate the vehicle displacement, represent the actual centre of the car. However, erroneous box unalignment is also a problem when using object detectors. In fact, the results in Table 4 favour our approach. It shows that our method can more effectively delineate regions relevant to tracker performance than when initialising the tracker with YOLOv4 detected regions. Still, with a mIoU of 0.733 for the moving vehicle class, there is room for improvement in delineating vehicles more precisely. Future work will focus on improving these segmentation results. Due to the many parked vehicles in the VisDrone dataset, our tracking approach could not be fairly evaluated using other tracking accuracy performance metrics that indicate the number of missed targets, identity switches, or ghost trajectories because our tracker was initialised for moving vehicles only. Future experiments will consider removing the static vehicle objects from the VisDrone ground truth file to include these performance metrics.

Finally, Table 5 shows that the flight configurations influence speed estimations. Further studies will investigate the influence of GPS accuracies on UAV displacement estimation. This study did not consider vehicles that travel in the opposite direction of the UAV. Equation 3 could be adapted to this case by replacing the minus sign with an addition sign. Future studies will implement this by developing an additional function that detects the travel direction of individual vehicles in relation to the UAV such that Equation 3 can be appropriately adjusted per car in real time.

In this study, and other studies conducted towards UAV-based vehicle speed detections, the lack of a unified UAV-based vehicle segmentation, tracking and speed benchmark dataset is the main limitation and, as was done in this study, requires a patchwork of datasets to develop and evaluate the different steps in the segmentation, tracking and speed detection process. As a result, the performance of the complete pipeline for a single study site could not be investigated. With the maturation of simulation software, such as AirSim, the usage of synthetic datasets will become a probable alternative.

6. CONCLUSION

We proposed a simple tracker initialisation method from segmentations instead of traditional object detection methods that could function onboard an edge device. Our method performs on par with trackers initialised with regions derived from object detectors. The main advantage of our approach is that the single multipurpose segmentation layer reduces the number of information products needed to carry out multiple tasks simultaneously.

REFERENCES

Balamuralidhar, N., Tilon, S., Nex, F., 2021. MultEYE: Monitoring System for Real-Time Vehicle Detection, Tracking and Speed Estimation from UAV Imagery on Edge-Computing Platforms. *Remote Sensing*, 13(4), 573. https://doi.org/10.3390/rs13040573.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B., 2016. Simple online and realtime tracking. *IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA. https://doi.org/ 10.1109/ICIP.2016.7533003.

Biswas, D., Su, H., Wang, C., & Stevanovic, A., 2019. Speed Estimation of Multiple Moving Objects from a Moving UAV

Platform. *ISPRS International Journal of Geo-Information*, 8(6). https://doi.org/10.3390/ijgi8060259.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

Bolme, D. S., Beveridge, J. R., Draper, B. A., Lui, Y. M., 2010.. Visual Object Tracking using Adaptive Correlation Filters. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA. https://doi.org/ 10.1109/CVPR.2010.5539960.

Chang, X., Pan, H., Sun, W., Gao, H., 2021. YolTrack: Multitask Learning Based Real-Time Multiobject Tracking and Segmentation for Autonomous Vehicles. *IEEE Trans Neural Netw Learn Syst,* 32(12), 5323-5333. https://doi.org/10.1109/TNNLS.2021.3056383

Czyżewski, A., Kotus, J., Szwoch, G., 2019. Estimating Traffic Intensity Employing Passive Acoustic Radar and Enhanced Microwave Doppler Radar Sensor. *Remote Sensing*, 12(1). https://doi.org/10.3390/rs12010110

Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., others., 2019. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. *Proceedings of the IEEE/CVF international conference on computer vision workshops*

Fernández Llorca, D., Hernández Martínez, A., García Daza, I., 2021. Vision-based vehicle speed estimation: A survey. *Intelligent Transport Systems*, 15(8), 987-1005. https://doi.org/10.1049/itr2.12079.

Gheorghiu, R. A., Iordache, V., Stan, V. A., 2021. Urban traffic detectors – comparison between inductive loop and magnetic sensors. *13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Pitesti, Romania. https://doi.org/ 10.1109/ECAI52376.2021.9515014.

Held, D., Thrun, S., Savarese, S., 2016. Learning to Track at 100 FPS with Deep Regression Networks. *Computer Vision–ECCV 2016: 14th European Conference*, Amsterdam, The Netherlands. https://doi.org/ 10.1007/978-3-319-46448-0_45.

Hossain, S., & Lee, D. J., 2019. Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices. *Sensors*, 19(15), 3371. https://doi.org/10.3390/s19153371.

Kumaar, S., Lyu, Y., Nex, F., Yang, M. Y., 2021. CABiNet: Efficient Context Aggregation Network for Low-Latency Semantic Segmentation. *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China. https://doi.org/ 10.1109/ICRA48506.2021.9560977.

Li, J., Chen, S., Zhang, F., Li, E., Yang, T., Lu, Z., 2019. An Adaptive Framework for Multi-Vehicle Ground Speed Estimation in Airborne Videos. *Remote Sensing*, 11(10), 1241. https://doi.org/10.3390/rs11101241.

Lukezic, A., Vojir, T., Cehovin Zajc, L., Matas, J., Kristan, M., 2018. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126(7), 671-688. https://doi.org/ 10.1007/s11263-017-1061-3.

Lyu, Y., Vosselman, G., Xia, G.-S., Yilmaz, A., Yang, M. Y., 2020. UAVid: A semantic segmentation dataset for UAV imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165, 108-119. https://doi.org/10.1016/j.isprsjprs.2020.05.009.

Milan, A., Leal-Taixe, L., Reid, I., Roth, S., Schindler, K., 2016. MOT16: A Benchmark for Multi-Object Tracking. arXiv preprint arXiv:1603.00831v2.

Nex, F., Armenakis, C., Cramer, M., Cucci, D. A., Gerke, M., Honkavaara, E., Kukko, A., Persello, C., Skaloud, J., 2022. UAV in the advent of the twenties: Where we stand and what is next. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184, 215-242. https://doi.org/10.1016/j.isprsjprs.2021.12.006.

Shan, D., Lei, T., Yin, X., Luo, Q., Gong, L., 2021. Extracting Key Traffic Parameters from UAV Video with On-Board Vehicle Data Validation. *Sensors*, 21(16). https://doi.org/10.3390/s21165620.

Siddiqui, O., DiBiase, S., Hoang, R., Nguyen, B., Khan, O., Famiglietti, N., 2021. Evaluating the Accuracy and Reliability of Bicycle GPS Devices. *SAE Int. J. Adv. & Curr. Prac. in Mobility*, 3(6), 3093-3114. https://doi.org/10.4271/2021-01-0882.

Tilon, S., Nex, F., Vosselman, G., Sevilla de la Llave, I., Kerle, N., 2022. Towards Improved Unmanned Aerial Vehicle Edge Intelligence: A Road Infrastructure Monitoring Case Study. *Remote Sensing*, 14(16), 4008. https://doi.org/10.3390/rs14164008.

Wang, C. Y., Bochkovskiy, A., Liao, H. Y. M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. *IEEE International Conference on Image Processing (ICIP)*, Beijing, China. https://doi.org/ 10.1109/ICIP.2017.8296962.

Zhang, J., Xiao, W., Coifman, B., Mills, J. P., 2020. Vehicle Tracking and Speed Estimation From Roadside Lidar. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 5597-5608. https://doi.org/10.1109/jstars.2020.3024921.

Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q., 2018. Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437.