2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI) | 979-8-3503-6213-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/RTSI61910.2024.10761327

Leveraging AI Techniques to Understand and Simulate Driving Behaviours

Angelo Coppola
Department of Civil, Architectural and Environmental Engineering
University of Naples "Federico II"
Naples, Italy
angelo.coppola@unina.it

Amir Reza Safari
Department of Civil, Architectural and Environmental Engineering
University of Naples "Federico II"
Naples, Italy
a.safari@studenti.unina.it

***Abstract*—Understanding and correctly analysing driving behaviours is pivotal for the advancement of autonomous vehicles and the impact assessment of intelligent transportation systems. Indeed, driving behaviour models derived from vehicle trajectories can be employed in several applications, including autonomous driving, traffic management and safety analysis. In this context, artificial intelligence techniques can be used to track vehicle trajectories and have a solid ground truth for analysing and calibrating driving behaviour. Common deep learning architectures can extract vehicle trajectories from camera-based traffic condition sensors. The proposed paper applies AI-enabled tracking of vehicle trajectories and shows how the obtained data can be adopted to identify different families of driving behaviour patterns, which are useful for estimating car-following models and improving the simulation of Intelligent Transportation Systems solutions. The results show that trajectories revealed by artificial intelligence tools, under different traffic conditions, enable the reliable and accurate validation of different modelling approaches.**

***Index Terms*—CCAM, Digital Twin, Simulation, Testing, Validation**

I. INTRODUCTION

Understanding and modelling driving behaviour is crucial for the successful implementation of Cooperative, Connected and Automated Mobility (CCAM) solutions, as well as and for their simulation in environments where the innovation must be tested and evaluated [1], [2]. Firstly, it enhances road safety by identifying and predicting risky driving patterns, enabling the development of advanced driver-assistance systems (ADAS) and autonomous vehicles that can more effectively react to real-world scenarios [3]. Secondly, it facilitates the creation of personalized driving experiences, adapting in-vehicle systems to individual driver preferences and habits. Finally, it supports urban planning and traffic management by providing insights into traffic flow and congestion, affecting infrastructure decision-making and reducing the overall travel time.

Since driving behaviour has a direct impact on fuel consumption and emissions [4], CCAM systems can optimize energy usage and reduce environmental impact by understanding driving patterns such as acceleration, deceleration, and speed fluctuations. Moreover, an automated recognition of driving behaviour along with easy identification of the resulting modelling parameters can help in dealing with the dispersion over cultures, geography, and individual preferences of driving styles. Thus, understanding these variations is essential to design CCAM systems that are adaptable and effective across different contexts.

In order to correctly analyse and reproduce driving behaviour, possibly taking into account the dispersion of the behaviour among the population of drivers, a massive amount of data is needed [5]. One of the possibilities for data gathering is to equip fleets of vehicles with GNSS devices and to keep them in platoons, observing the reciprocal cinematic of the vehicles. Another possibility is to equip a vehicle with appropriate sensors like radars, lidars, or cameras to keep track not only of the kinematics of the vehicle but also of the relative positioning and relative speed with respect to the surrounding vehicles and the traversed road. While this approach offers a more detailed analysis, it falls short in terms of equipping costs per vehicle.

A different approach is to make traffic analysis using other devices which are external from the traffic stream and thus non-solidly attached to them, such as Unmanned Aerial Vehicles (UAV) [6]–[8].

The aim of this paper is to explore the application of artificial intelligence techniques for tracking vehicle trajectories and utilizing this data to enhance the understanding and modelling of driving behaviours. By leveraging deep learning architectures to extract vehicle trajectories from camera-based traffic sensors, the study aims to identify various driving behaviour patterns. These patterns are crucial for developing accurate car-following models and improving the simulation of Intelligent Transportation Systems (ITS). The research demonstrates how AI-enabled trajectory tracking, under varying traffic conditions, provides a robust ground truth for analysing and calibrating driving behaviours, thereby enabling reliable and precise validation of different modelling approaches. The ultimate goal is to advance the development of autonomous vehicles and optimize traffic management and safety analysis

---

Authors are listed in alphabetical order. This research was supported by the Italian Ministry of Research (Ministero dell'Università e della Ricerca - MUR) within the framework of the project DIGIT-CCAM (grant E67G22000010005).

979-8-3503-6213-8/24/$31.00 ©2024 IEEE

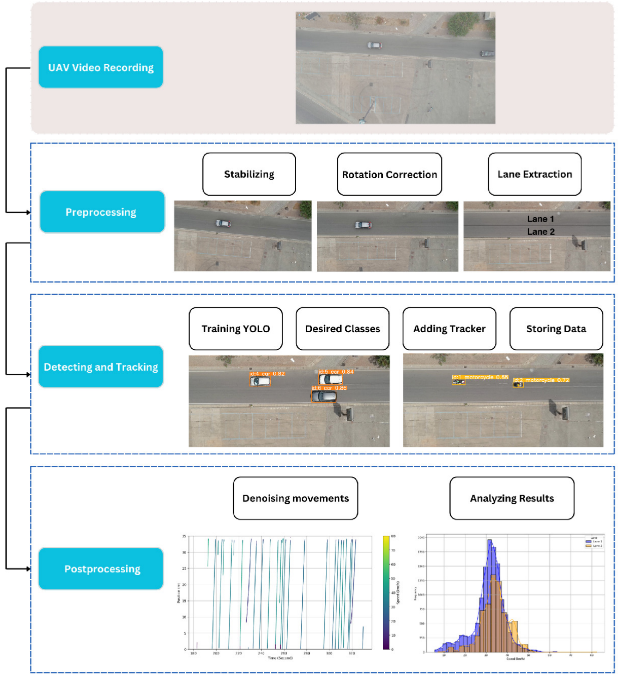


Fig. 1: Flowchart of the employed framework pipeline to estimate drivers behaviours using UAV captured footage

within ITS solutions.

II. METHODOLOGY

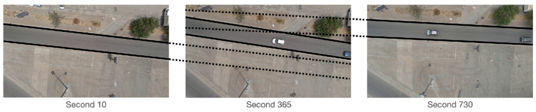
In order to present the process of understanding driving behaviours hastily, we decided to propose a framework consisting of an operative pipeline in order to leverage the available AI techniques. More in detail, the latter consists of four steps:

1) Recording video footage of the road portion under test. This can be done using roadside surveillance cameras or, as in our case, through a UAV.
2) Apply a pre-processing filter, consisting of rotation correction and video stabilization.
3) Proceed with vehicle detection using a deep-learning approach, specifically through a pre-trained neural network such as the YOLO (You Only Look Once) algorithm; we decided for version 8 of the latter combined with a tracking algorithm to trace detected vehicles' trajectory along the video frames [9].
4) Post-processing step consisting of a denoising algorithm to produce a smoother trajectory figure
5) Data analysis, including vehicle trajectories, counting, and speed estimation

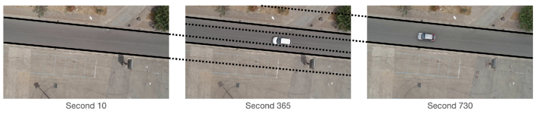
Figure 1 visualizes the framework pipeline just described.

A. *Video Recording*

The first step involves recording road video footage using a remote-controlled UAV equipped with a video camera. In our case, we used the EVO II Dual 640T RTK with a visible light camera [10]. The video recorded has a duration of 12 minutes and 13 seconds (732.73 seconds) with a resolution of 3840 x 2160 pixels and a frame rate of 30 frames per second, resulting in a total of 21,960 frames. The camera has a focal length of 25.6 mm, and the UAV was flown at a height of 60 meters, covering an area of 58 meters of the road.



(a) *Before stabilization and rotation correction*



(b) *After stabilization*

(c) *After stabilization and rotation correction*

Fig. 2: Stabilization and rotation correction

B. *Pre-processing*

Road extraction involves limiting the region of interest (ROI) to decrease the false positive error of detecting vehicles outside the ROI. This is achieved by drawing a polygon on the video to detect vehicles within that defined area. However, the recorded video is considerably unstable due to wind and UAV motor vibration. Therefore, to mitigate this issue, video stabilization is necessary [11] as a preliminary step to ROI extraction. This step was conducted through the DaVinci Resolve Software [12]. Furthermore, if the road is not perpendicular to the video frame, a rotation correction step is also mandatory, as we can appreciate in Figure 2). It is important to note that both stabilization and rotation correction may lead to footage reduction. Indeed, as a result of it, the road extension on the video footage is decreased from 58 meters to about 37 meters after the preprocessing step. However, only after these two corrections, it is then possible to draw the ROI.

C. *Vehicle Recognition Methods*

As for vehicle detection, we employed the YOLO version 8 deep learning algorithm. YOLO is a popular object detection and image segmentation model developed by Joseph Redmon and Ali Farhadi at the University of Washington and released for the first time in 2015. It quickly gained popularity both for its high speed and accuracy. YOLOv8 is the latest version of YOLO by Ultralytics and supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. Different variants of YOLOv8 exist, with different size and complexity of the model. Larger models are more accurate, but the detection is also slower, while smaller network are favoured when dealing with real-time detections, even though the accuracy decreases. Detect, Segment and Pose models are pretrained on the COCO dataset (capable of detecting 80 different classes), while Classify models are pretrained on the ImageNet dataset.

During the detection phase, we observed limitations when applying YOLO to UAV-based videos, particularly when detecting vehicles from a vertical perspective. Consequently, we undertook additional training on the algorithm, annotating vehicles to facilitate recognition from a vertical viewpoint and obtain corresponding weights for future detections. With the newly trained algorithm, we effectively managed to detect the desired classes of vehicles in UAV videos. Three different classes are of our interest: car, motorcycle, and truck, each of which has different Passenger Car Units (PCU), which is a metric used to describe road capacity depending on vehicles' length and width. With the detection step sorted out, multi-object tracking (MOT) of the detected vehicles was now possible, allowing us to assign specific IDs to each vehicle and subsequently store related data. While previous literature offers various tracking algorithms, newer versions of YOLO enable simultaneous detection and tracking of vehicles. Two tracking algorithms, Bot-SORT [13] and ByteTrack [14] are employed, with the latter demonstrating superior performance by minimizing noise in data.

D. Postprocessing

Once the position of each vehicle at every frame is obtained, it is possible to reconstruct their trajectories over time. During this operative step, two main drawbacks were encountered:

1) The first problem is Bounding Box Inaccuracy (Figure 3), which is caused by the fact that while a vehicle enters or exits the video frame its bounding box size is not complete. This issue heavily compromises trajectory analysis and plotting, as the drawing is not in the correct position during these frames (Figure 4). To mitigate this issue, we removed the x-coordinates within 2 meters from each side of the frame, ensuring that only the accurate positions are considered for analysis when bounding boxes are complete.
2) The second problem is related to data noise. While analyzing the results, we observed significant noise in the trajectory data due to fluctuations in the size of bounding boxes over time. To overcome this issue, we employed a wavelet denoising algorithm [15] with tailored configurations and parameters, as we can appreciate in figures 5a and 5b). This approach effectively reduces noise, resulting in much smoother trajectory lines for enhanced analysis and interpretation. Figure 6 illustrates the position of a sample vehicle within the video frame and comparison between raw and denoised coordinate data. In our trajectory denoising model, we experimented with various parameter configurations based on commonly utilized basis families, including Daubechies, Symlet, and Coiflet. Specifically, we employed the Coiflet1 basis family with a hard mode setting.

Fig. 3: Exemplary condition leading to Bounding Box Inaccuracy

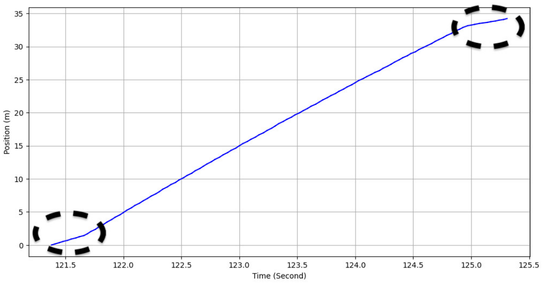


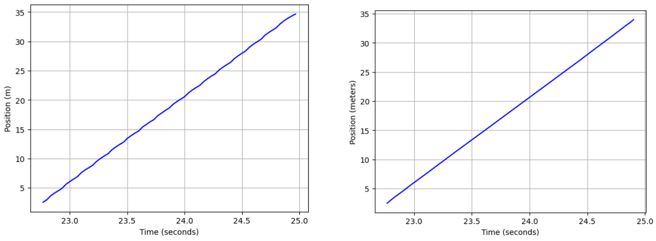
Fig. 4: Trajectory inaccuracy due to Bounding Box Inaccuracy

III. DATA ANALYSIS

This section discusses the outcome of the applied methodology on the recoded video.

The first result of the analysis is the vehicle counting and trajectories. In addition, we were able to obtain data regarding the associated lane with each detected vehicle, vehicle counts per lane, vehicle speeds, and the density and flow of each lane. We also calculated the spacing and gap between consecutive vehicles. First, we divided the road segment into two lanes, as illustrated in Figure 7. Then, we calculated the number of vehicles for each lane, and reconstructed their trajectories over time. The results in Figure 8 indicate that most of the vehicles travel along Lane 1. This difference in lane utilization can be associated with drivers preferences and specific road geometry features.

The distribution of vehicles within the lanes was analysed using their bounding boxes' centre positions along Y coordinate axis in meters. The distribution of bounding boxes' centre along Y coordinate indicates how vehicles are spread across the width of the road. The presence of peaks in the histogram suggests preferred positions within lanes, likely corresponding to lane centres. When lane boundaries are marked, it becomes evident that vehicles tend to stay within specific regions, reinforcing the notion of lane discipline. Figure 9 shows the distribution of vehicles along the whole road carriage (both lanes). Most of the vehicles travel in Lane 1 (2.7 – 6.2 meters), but their position is not centred in the lane but shifted towards the centre of the carriage. Figure 10 shows the distributions of vehicles per each lane. The figure clearly shows how most of the vehicles travel near the middle lane (dividing the two lanes). The two distributions for the two lanes confirm that



(a) Raw trajectory (b) Denoised trajectory

Fig. 5: Trajectory comparison in x-time diagram of a sampled vehicle of both raw (a) and denoised (b) trajectories

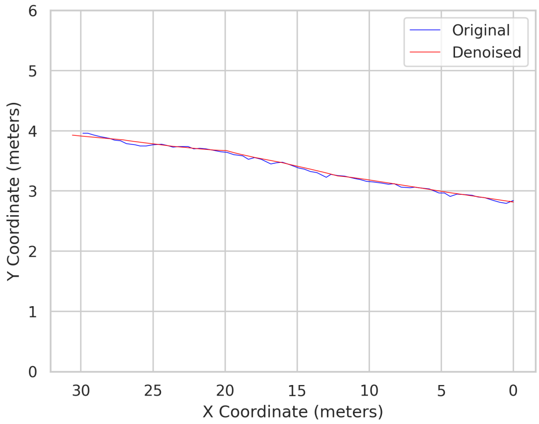


Fig. 6: Raw and Denoised Trajectory comparison in x-y diagram.

vehicles, on average, do not follow the middle part of a lane, but their position is shifted towards the middle of the whole carriage. This trend shows that drivers tend to occupy, in the lateral direction, as much space as possible; accordingly, the road capacity could be affected by this behaviour since the road segment is not exploited at its best, i.e., in some cases drivers travel as in a one-lane road segment. Then, we examined the speed distribution for each lane by means of histograms. The analysis in Figure 11 showed distinct speed distribution profiles for each lane, highlighting the variations in driving behaviour across lanes. The outcome proves that, on average, the vehicles travelling along Lane 2 travel at a higher speed.

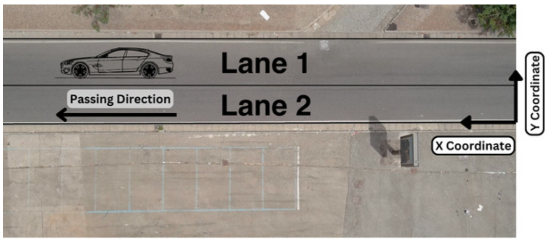
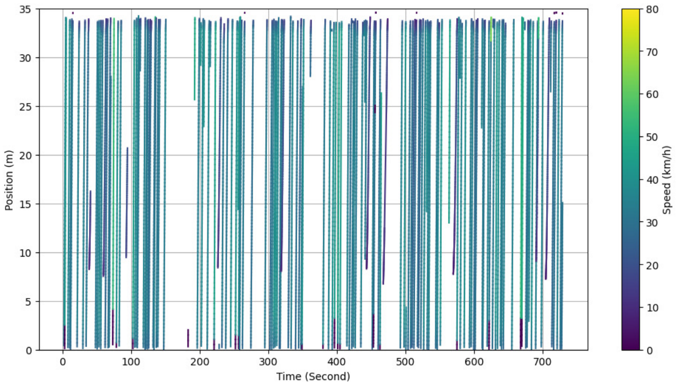
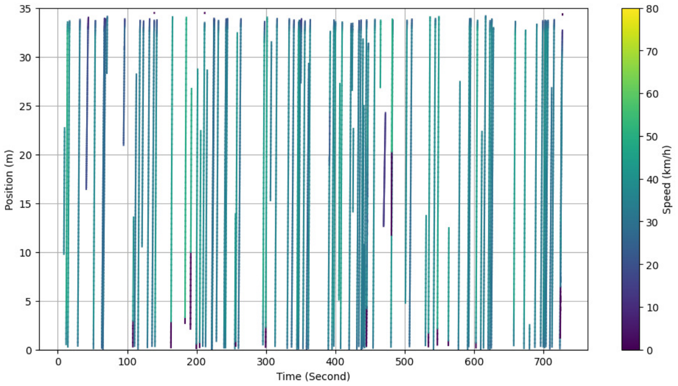


Fig. 7: Road section division into two lanes.



(a) Lane 1 trajectories



(b) Lane 2 trajectories

Fig. 8: Trajectory of Vehicles over time. (a) Lane 1; (b) Lane 2.

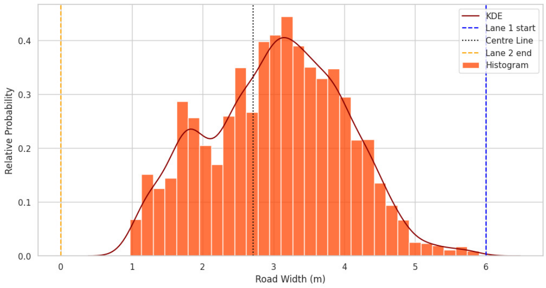


Fig. 9: Distribution of vehicles across the whole road carriage

IV. DISCUSSION

Vehicle trajectories are detailed records of the path that vehicles take over time, typically including information such as position, speed, and acceleration at various points along their route. These trajectories can be utilized in several ways, particularly for developing behavioural models and calibrating traffic simulators.

Regarding behavioural models, these latter describe how drivers and vehicles interact with each other and with the road environment. Vehicle trajectory data can be used to:

- develop models that replicate realistic driving action by analysing trajectories, in particular observing patterns in

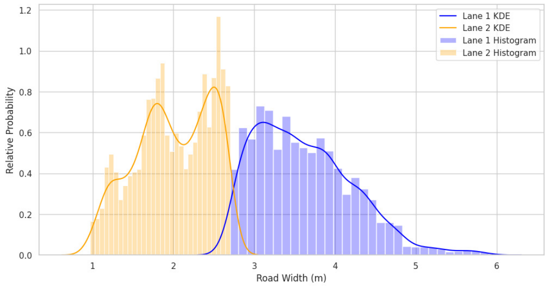


Fig. 10: Distribution of vehicles across each lane.

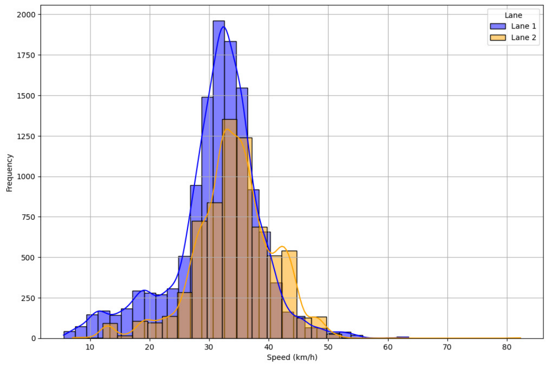


Fig. 11: Speed distribution over the two lanes.

lane-changing, acceleration, deceleration, and speed.

- develop predictive models that can be used in autonomous vehicle systems to anticipate and react to other vehicles' movements.
- identify near-misses or patterns leading up to collisions, which helps in developing safety measures or ADASs.

Regarding the calibration of traffic simulators, this involves adjusting the parameters of the simulation model so that its output matches real-world observations. Vehicle trajectory data enhances this calibration process in several ways:

- trajectories provide fine-grained data on vehicle movements, allowing for precise calibration of models regarding acceleration, deceleration, lane-changing, and gap-acceptance behaviours.
- by comparing simulated trajectories with real-world data, the accuracy of the simulator can be validated more comprehensively than with aggregate data like vehicle counts and speeds.
- Trajectories offer microscopic details that help in refining the models to represent realistic interactions between individual vehicles, which is essential for accurate traffic flow simulation and prediction.

V. CONCLUSIONS

Driving behaviour reconstruction and analysis holds the potential to improve the performance of drivers, road and traffic safety, fuel management, and designing more efficient intelligent transportation systems and ADASs. Hence, this is a crucial task for the correct development of CCAM services, especially under mixed traffic flow conditions.

To this end, this work proposes a methodology to identify different families of driving behaviour patterns by means of AI-enabled tracking of vehicle trajectories. The trajectories of vehicles were extracted by road video recorded with a remote-controlled UAV equipped with a video camera. A pre-processing filter, consisting of rotation correction and video stabilization, was applied, and then the vehicle detection task was done using a deep-learning approach through YOLO v.8. The output underwent a post-processing step consisting of a denoising algorithm to produce a smoother trajectory. Finally, the data analysis, including vehicle trajectories, counting, and speed estimation have been carried out.

The results showed that trajectories revealed by artificial intelligence tools, under different traffic conditions, enable the reliable and accurate validation of different modelling approaches.

Possible exemptions are: use of different input dataset (videos); use and comparison of the performance of different neural network model; use of a different data set for training; on-line use of the algorithm.

REFERENCES

[1] G. N. Bifulco, L. Pariota, F. Simonelli, and R. Di Pace, "Real-time smoothing of car-following data through sensor-fusion techniques," *Procedia, Social and Behavioral Sciences*, vol. 20, pp. 524–535, 2011.

[2] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 2020.

[3] G. N. Bifulco, B. Caiazzo, A. Coppola, and S. Santini, "Intersection crossing in mixed traffic flow environment leveraging v2x information," in *Proceedings of the 8th IEEE International Conference on Connected Vehicles*, November 2019.

[4] S. de Luca, R. Di Pace, S. Memoli, and L. Pariota, "Sustainable traffic management in an urban area: An integrated framework for real-time traffic control and route guidance design," *Sustainability*, vol. 12, no. 2, 2020.

[5] M. N. Azadani and A. Boukerche, "Driving behavior analysis guidelines for intelligent transportation systems," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 7, pp. 6027–6045, 2021.

[6] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, March 2017.

[7] E. N. Barmpounakis, E. I. Vlahogianni, and J. C. Golias, "Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges," *Int. J. Transp. Sci. Technol.*, vol. 5, pp. 111–122, 2016.

[8] M. A. Khan, W. Ectors, T. Bellemans, D. Janssens, and G. Wets, "Uav-based traffic analysis: A universal guiding framework based on literature survey," in *Transportation Research Procedia*. Amsterdam, The Netherlands: Elsevier, 2017.

[9] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolo (version 8.0.0) [computer software]," 2023, available at: https://github.com/ultralytics/ultralytics.

[10] Autel Robotics, *EVO II Dual 640T RTK User Manual*, 2021, available at: https://www.autelrobotics.com.

[11] G. W., O. L., and A. Beghdadi, "Video stabilization: Overview, challenges and perspectives," *Signal Processing: Image Communication*, vol. 90, p. 116015, 2021.

[12] Blackmagic Design, *DaVinci Resolve 17*, 2020, available at: https://www.blackmagicdesign.com/products/davinciresolve.

[13] N. A., R. O., and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *ArXiv*, vol. abs/2206.14651, 2022, available at: https://arxiv.org/abs/2206.14651.

[14] Y. Z., S. P., J. Y., D. Y., Z. Y., L. P., L. W., and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*, 2021. [Online]. Available: https://www.ecva.net/papers/eccv_2021/papers_ECCV/html/1234_ECCV_2021_paper.php

[15] X. Chen, Z. Li, Y. Yang, L. Qi, and R. Ke, "High-resolution vehicle trajectory extraction and denoising from aerial videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3190–3202, May 2021.