COMPARATIVE STUDY OF VISION-BASED METHODS FOR REAL-TIME TRAFFIC MONITORING

Diksha Aggarwal,* Surafel T. Anshebo,* Kevin Kochersberger,† and A. Lynn Abbott‡

Uncrewed Aerial Vehicles (UAV) technology is advancing dramatically with applications ranging from surveillance to disaster response. One such area is real-time traffic monitoring where UAVs with their vision-based methods can play a significant role in streamlining traffic flow, mitigating congestion, and facilitating quick emergency response in accidents. However, such an implementation is accompanied by challenges related to accuracy and computational overload. Therefore, a comparison of available methodologies in detection and tracking using computer vision is presented in this paper over different traffic scenarios.

Keywords: *Uncrewed Aerial Vehicles (UAVs), Multiple Object Tracking, Accuracy, Single Object Tracking, Computer Vision*

INTRODUCTION

Vision-based surveillance systems are ubiquitous in urban areas. The role played by cameras in providing data for crowd counting, urban planning, traffic management studies as well as disaster response management is extremely valuable[1]. Small Uncrewed Aerial Vehicles (sUAVs) are emerging as viable platforms for surveillance applications, given their ability to maneuver in three dimensions and fly through narrow passages. Furthermore, with the advancement of the capabilities of vision algorithms, equipping sUAVs with vision sensors makes them a suitable technology for aiding in ad-hoc scenarios in traffic safety, streamlining traffic flow, and most importantly, improving responsiveness to emergencies. All these applications are predicated on pattern recognition, object detection, and tracking, which are essentially techniques within the framework of computer vision.

The state-of-the-art techniques in vision-based algorithms can be classified into traditional computer vision algorithms and deep learning-based algorithms. While the former boasts computational efficiency, the latter exhibits robustness in terms of solutions and remains less affected by scaling and occlusion. Both these categories of algorithms are widely studied for traffic applications[2]. While they are distinct in terms of their solution approaches, a common challenge confronted by both approaches is their ability to respond to different lighting conditions and occlusions. Given that the current research aims to deploy UAV-based vision systems for traffic safety, it is imperative that the deployed algorithms can overcome such challenges. Hence, a comprehensive understanding of the performance of each category of algorithms under such challenging conditions becomes crucial.

* Graduate Student, Department of Mechanical Engineering, Virginia Tech
† Professor, Department of Mechanical Engineering, Virginia Tech
‡ Professor, Bradley Department of Electrical and Computer Engineering, Virginia Tech

https://doi.org/10.52202/075106-0004

This paper thus proposes to evaluate and compare the performance of the traditional computer vision algorithms and deep learning algorithms, by applying them to vehicle traffic monitoring. This entails both detection as well as tracking of vehicles. Specifically, the Haar Cascade[3], Lucas-Kanade optical flow[4], and SORT[5] algorithms are explored in terms of traditional algorithms, whereas YOLOv5[6] and Deep SORT[7] are the deep learning algorithms that are investigated. Moreover, this work presents the performance of several combinations of the above algorithms under various lighting conditions and views, quantified via state-of-the-art metrics. Since the overarching goal is to understand the performance of the algorithms under challenging conditions like lighting and occlusion, the VisDrone dataset appears to be the most suitable, as it offers the expected challenging data to test the algorithms. The remainder of the article is organized as follows: The next section discusses the motivation for this research and reviews the related work. Following that we provide a brief overview of the dataset and algorithms evaluated. We then present the evaluation metrics with results and finally, we conclude the article with a summary and conclusion.

RELATED WORK

Traditional computer vision methods leverage inherent image attributes such as texture, color, and shape and have been the mainstay for years. These approaches include template matching[3,8] and feature extraction methods like HOG, SIFT, and ORB. Researchers have combined these extracted features with machine learning-based classifiers like Haar Cascade, KNN[9], and SVM[10] to detect vehicles. These detectors perform well but are sensitive to scale variations and less effective in cluttered environments[11]. Vehicle detection has also been explored using deep learning-based methods. These include two-staged approaches like Fast RCNN[12], Faster RCNN[13], and Mask RCNN[14], and one-staged methods like SSD[15], and YOLO[16]. These advanced methods offer improved performance and accuracy in vehicle detection tasks.

Detected objects can be tracked using methods such as Lucas-Kanade optical flow[17], mean-shift tracking, and Kalman filter-based SORT[7]. These tracking approaches are mostly based on predicting the next probable position based on tracking features. However, they can encounter issues related to scaling and occlusion which can affect their performance. These methods have been used in both single and multi-object tracking. In single object tracking (SOT), a vehicle is detected in the first frame, or its position is known a priori. As a result, the performance entirely depends on the tracking algorithm used. However, in multi-object tracking (MOT), a detection method is required at each frame to identify the targets that are leaving or entering the frame. Each detected object is assigned a unique ID and is tracked across frames. The IDs are assigned based on the matching detections with existing tracks based on criteria such as distance, appearance, and motion patterns. As a result, the accuracy of the tracker largely depends on the accuracy and precision of the detection algorithm. Therefore, researchers have been integrating deep learning-based object detection methods with traditional trackers to make them robust. For example, Gomaa et al.[18] combined optical flow with CNN and K-means clustering to remove background and generate features on the segmented vehicles to track. Similarly, other researchers have leveraged deep learning to track the objects for a longer time and reduce the number of ID switches when occlusion occurs. For example, Wojke et al.[7] introduced Deep SORT, an extension of the SORT algorithm with the integration of appearance information by using a re-identification CNN model.

While working with deep learning-based algorithms, one of the major challenges is obtaining a high-quality dataset that provides sufficient labeled images in different conditions. In recent years, several challenges have been organized worldwide to promote research and development in object detection and tracking algorithms like Computer Vision and Pattern Recognition (CVPR) challenges, MOT challenges, and workshops. As a result, datasets such as KITTI[19] for autonomous

https://doi.org/10.52202/075106-0004

vehicles in cities, and MOT15[20] and MOT16[21] for tracking pedestrians in urban environments are developed. These datasets typically focus on the images taken from a ground vehicle. To perform operations safety-critical operations using sUAVs, an aerial dataset is necessary. The underlying challenge with the drone dataset is that it can produce blurry images due to the camera motion. The scale of the objects also varies with the altitude of the drone. Small angular variations can produce large changes in the camera view which can make it difficult for the algorithm to keep track of the objects. Hsieh et al.[15] presented the first paper with an aerial dataset for counting cars in a parking lot. Du et al.[22] developed UAVDT aerial manually annotated datasets for vehicles for single and multiple object tracking while considering different weather conditions and camera angles. Zhu et al.[23] created the VisDrone dataset with 10 object categories including people, bicycles, cars, vans, trucks, etc. in different traffic scenarios from various angles.

OVERVIEW: DATASET AND ALGORITHMS

The VisDrone2019 dataset[23], developed by the AISKYEYE team at Tianjin University's Lab of Machine Learning and Data Mining in China is used in this research. The dataset comprises 288 video clips, 10,209 static images, and 261,908 frames captured by diverse drone-mounted cameras. The dataset spans a broad spectrum, including various geographic locations (drawn from 14 different cities across China, spanning thousands of kilometers), environments (urban and rural), objects (pedestrians, vehicles, bicycles, etc.), as well as scene densities (ranging from sparse to crowded). The dataset features manual annotations, with over 2.6 million bounding boxes delineating targets of frequent interest. In this study, three image sequences have been chosen. For ease of reference, we have used seq1 for uav0000137_00458_v, seq2 for uav0000268_05773_v, and seq3 for uav0000305_00000_v as shown in Figures 1, 2, and 3 respectively.

Figure 1. VisDrone Dataset Seq1 (uav0000137_00458_v)

Figure 2. VisDrone Dataset Seq2 (uav0000268_05773_v)

Figure 3. VisDrone Dataset Seq3 (uav0000305_00000_v)

Under the category of traditional computer vision, three algorithms are employed, namely, the Haar Cascade for vehicle detection, Lucas-Kanade optical flow, and Kalman filtering-based SORT for vehicle tracking. In the deep learning-based approach, this paper uses YOLOv5 for vehicle detection and leverages a Kalman filtering and reidentification-based tracker, Deep SORT to perform tracking of the detected vehicles.

Haar Cascade

Viola and Jones developed the Haar Cascade[3] method which is distinguished by three key contributions- First, the feature selection process uses Haar features, simple patterns derived from the image, to capture changes in texture or edge orientation. Second, it employs integral images for fast feature calculation, and third, the training uses AdaBoost, an algorithm for selecting the most critical features from a large set and combining them into a strong classifier. This classifier is trained using two kinds of images: Positive images that include objects of interest and negative images that do not include the objects. The process builds a cascade of classifiers, each more complex than the previous, designed to eliminate any incorrect detections.

Optical Flow

The Lucas-Kanade optical flow[4,24] method is used for tracking motion between consecutive images or video frames. It calculates the motion in terms of the displacement of pixels from one frame to the next. It is based on the assumption that the brightness of any object remains constant over time. The algorithm analyzes the pattern and magnitude of these motion vectors to estimate the direction and speed of moving objects in the images. This information is then used to predict the location of bounding boxes in the subsequent frames.

SORT

Simple Online and Realtime Tracking (SORT)[5] developed by Bewley et al. is an object detection-based tracker. It uses an object detection algorithm to detect objects in each frame of a video or image sequence and create bounding boxes on the targets. In the first frame, these bounding boxes will initiate the object tracks, and an identification (ID) is assigned to each object. In subsequent frames, the goal is to associate the detected bounding boxes with existing object tracks produced by the Kalman filter. This is achieved by using an association matrix which is based on Intersection Over Union (IOU). This matrix is then used by the Hungarian algorithm to associate the valid tracks. When an object is occluded or moved out of the frame, the tracking algorithm allows trackers to exist for a short time without new matching detections, giving the object a chance to reappear. If a tracker does not receive a matching detection during this time, it is removed.

https://doi.org/10.52202/075106-0004

YOLOv5

YOLOv5 is an advanced version of the YOLO architecture[6]. Unlike traditional two-stage detectors like R-CNN, which first select region proposals and then classify them, YOLOv5 is a single-stage detector that predicts both bounding boxes and class probabilities directly from the image in one evaluation. This makes it exceptionally fast and suitable for real-time applications. YOLOv5 is different from its previous versions with the use of a more efficient neural network EfficientDet[25]. Moreover, YOLOv5 integrates well with modern deep learning libraries such as PyTorch, which makes the development and deployment easier.

Deep SORT

Deep SORT is an extension of the SORT algorithm[7]. While SORT uses Kalman filtering and the Hungarian algorithm for object tracking, Deep SORT also incorporates deep learning features in the association matrix. Deep SORT uses a reidentification-based convolutional neural network (CNN) to extract deep features from detected objects. These features help in recognizing the same object across different frames, even when its appearance changes due to lighting, viewpoint, or other factors. This helps to improve tracking performance, especially in cases of occlusions or interactions between objects.

EVALUATION

The evaluation of the different combinations of algorithms is divided as follows: evaluation of detection algorithms Haar Cascade and YOLOv5 over all frames of video, evaluation of optical flow for SOT, evaluation of SORT and Deep SORT on SOT with Haar Cascade and YOLOv5.

Evaluation Of Detection Algorithms

YOLOv5 is not trained on a vehicle dataset, thus the weights found on the official website of the YOLOv5 repository could not be used effectively for detecting the vehicles. Therefore, a transfer learning approach has been utilized in this work. The pre-trained YOLOv5 model is trained on a new dataset of vehicles. The vehicle dataset is taken from a GitHub repository[23]. The dataset consists of a series of images containing around 1300 images of vehicles. The dataset is divided into 70% training images, 20% validation images, and 10% testing images. The training is done for 50 epochs in the batch of 16 and it took 1.25 hours to train the model. The metrics used to evaluate the training accuracy of YOLOv5 are precision, recall, and Mean Average Precision (mAP). Precision measures the percentage of the model's positive identifications that were correct. Recall estimates how good the model is at detecting positive cases. Finally, mAP is calculated as the mean of average precision for each category in the dataset. The results of the training are shown in Figure 4. The mAP from this training is 0.773.

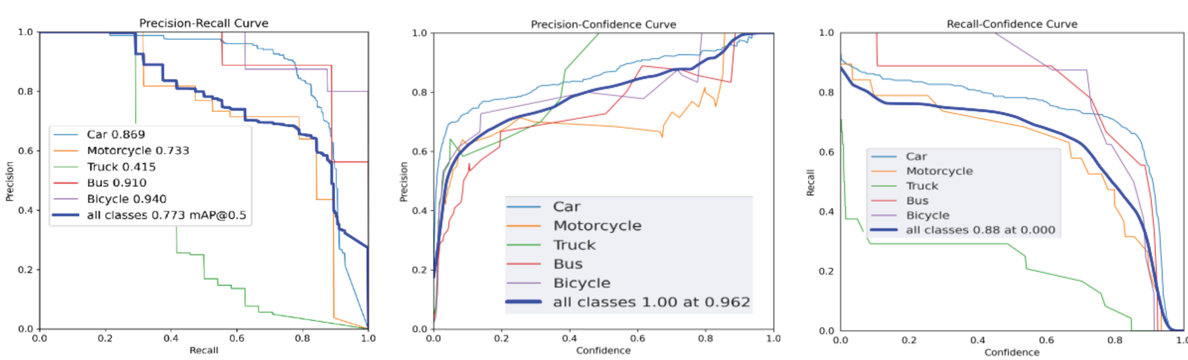


Figure 4. YOLOv5 Training Precision And Recall Curve.

The training weights for the Haar Cascade model are taken from this GitHub repository*. The model was trained using 526 images of cars from the rear (360 x 240 pixels, no scale). The images were extracted from the Car dataset proposed by Brad Philip and Paul Updike taken of the freeways of southern California†.

The detection algorithms are evaluated using recall and precision. The value changes with the tracking algorithms since the tracking algorithms used in this paper generate detection using Kalman filtering. Thus, Haar Cascade and YOLOv5 are implemented on the video and compared with the ground truth data, and recall and precision have been reported as shown in Table 1. It can be that the values are higher in YOLOv5. This is due to the high number of False Positives and the size of the bounding box generated by the Haar Cascade method which highly affects its performance.

Table 1. Performance Metrics For Detection Algorithms Without Tracker On Seq1.

Method	Recall	Precision
Haar Cascade	0.0718	0.0956
YOLOv5	0.5871	0.8999

Evaluation Of MOT Algorithms

For tracking, pre-trained weights of the Deep-SORT model are used and are obtained from the official Deep SORT repository. Evaluation of MOT algorithms is done using the metrics taken from CLEAR MOT[26]. The metrics for identification include ID Precision (IDP), ID Recall (IDR), and IDF1 (ID F1 score). IDF1 is the harmonic mean of IDP and IDR, reflecting the balance between accurately identifying objects and minimizing missed or incorrect identifications. MOT Accuracy (MOTA) consolidates three types of tracking errors into a single performance indicator: the detection errors of False Positives (FPs) and False Negatives (FNs) as well as association errors using ID switches (IDSWs). Since MOTA does not include the localization error, MOTP is defined by CLEAR MOT as the overall tracking precision in terms of average overlap between the detected object and its ground truth. The other trajectory-based metrics[27] used are Mostly Tracked (MT), Partially Tracked (PT), Mostly Lost (ML), and Fragmentations (FM). MT is the number of objects tracked for at least 80% of their span, PT is the number of objects tracked between 20 and 80% of their span, ML is the number of objects tracked for at most 20% of their span and FM is the number of times a track was interrupted by a missing detection.

MOT is performed using three combinations of algorithms: Haar Cascade with SORT, YOLOv5 with SORT, and YOLOv5 with Deep SORT. The results corresponding to each case on seq1, seq2, and seq3, are shown in Tables 2, 3, and 4 respectively. The analysis shows that Haar Cascade with SORT exhibits lower performance as compared to YOLOv5 with SORT. One of the reasons for this observation is the scaling effect and the high number of FNs. The Haar Cascade performs optimally when the objects match the size it was trained on. Another observation is that both trackers demonstrate lower accuracy on seq2 compared to seq1. This may be attributed to the brighter images in seq2, which could have impacted the performance of detection algorithms. Both these findings emphasize that the accuracy of the tracking algorithm depends on the accuracy of

* https://github.com/andrewssobral/vehicle_detection_haarcascades?tab=readme-ov-file
† https://data.caltech.edu/records/dvx6b-vsc46

https://doi.org/10.52202/075106-0004

the detection algorithm. Overall, the evaluation highlights the importance of choosing the right combination of detection and tracking algorithms for optimal performance in MOT.

Table 2. Performance Metrics For MOT On Seq1.

Method	MOTA [%]	MOTP [%]	FM	MT	ML	FP	FN	IDSW	IDF1	IDP	IDR
Haar Cascade + SORT	-2.37	56.7	**12**	0	44	**193**	7023	**11**	0.0044	0.0696	0.0023
YOLOv5 + SORT	**49.6**	**83.83**	62	**18**	14	239	**3261**	26	**0.6244**	**0.8619**	**0.4896**
YOLOv5 + Deep SORT	45.6	79.61	135	**18**	**13**	400	3295	109	0.5438	0.7357	0.4313

Table 3. Performance Metrics For MOT On Seq2.

Method	MOTA [%]	MOTP [%]	FM	MT	ML	FP	FN	IDSW	IDF1	IDP	IDR
Haar Cascade + SORT	-3.94	52.7	98	0	52	712	13350	51	0.0118	0.0915	0.0063
YOLOv5 + SORT	1.39	**84.64**	**19**	0	**48**	615	**12767**	**7**	**0.0993**	0.5224	**0.0549**
YOLOv5 + Deep SORT	**1.75**	70.47	39	0	50	**47**	13286	8	0.0365	**0.7492**	0.0187

Table 4. Performance Metrics For MOT On Seq3.

Method	MOTA [%]	MOTP [%]	FM	MT	ML	FP	FN	IDSW	IDF1	IDP	IDR
Haar Cascade + SORT	-1.8	47.23	**4**	0	69	**131**	6376	**1**	0.0037	0.082	0.0019
YOLOv5 + SORT	**30.09**	**86.05**	21	**13**	**44**	235	**4232**	**1**	**0.4871**	**0.8939**	**0.3348**
YOLOv5 + Deep SORT	13.31	76.71	39	5	53	250	5283	7	0.2697	0.7695	0.1635

In the comparison between SORT and Deep SORT with YOLOv5, the accuracy is nearly consistent for both seq1 and seq2. However, the accuracy of Deep SORT is significantly low on seq3. This can be attributed to the reidentification model being trained on oblique images of cars while seq3 has images taken from a downward-looking camera. This exhibits the challenge of using

https://doi.org/10.52202/075106-0004

the same deep learning-based model for different camera views. Moreover, it can be inferred that in MOT, the deep learning-based algorithms are more effective for object detection while traditional algorithms perform better in object tracking. This harmoniously agrees with the evaluation performed by Pereira et al.[29].

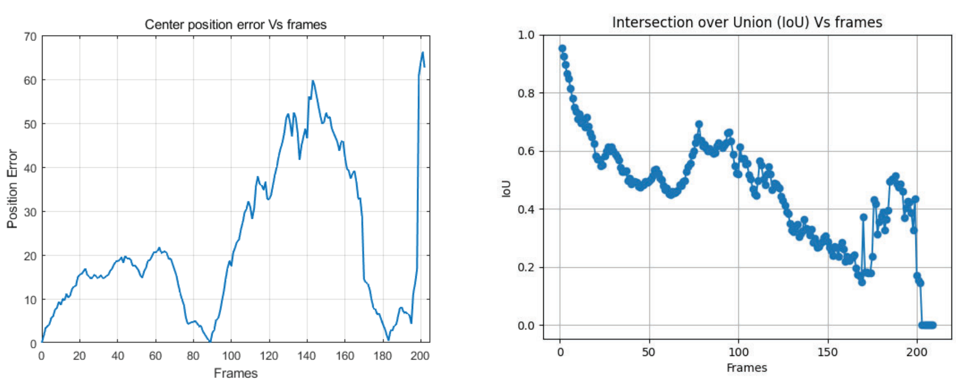
Table 5. Performance Metrics For MOT On Seq3 For Yolov5 With Deep SORT With Change In Max_dist.

Parameter	MOTA [%]	MOTP [%]	FM	MT	ML	FP	FN	IDSW	IDF1	IDP	IDR
Max_dist = 0.2	**13.31**	**76.71**	**39**	**5**	**53**	**250**	5283	**7**	0.2697	**0.7695**	0.1635
Max_dist = 0.5	**13.31**	76.6	40	**5**	**53**	255	**5277**	8	**0.2703**	0.7662	**0.1641**

Deep SORT has various parameters that affect its performance like Max_dist, maximum IOU, and minimum confidence. Max_dist is the maximum allowed Mahalanobis distance for matching a new detection to an existing track. These parameters are varied to check the accuracy of MOT. It can be seen that with an increase in Max_dist, its FNs have reduced as shown in Table 5. The Max_dist value increased from 0.2 to 0.7. However, the results didn't show any improvement after 0.5. Varying the minimum confidence parameter of the detection algorithm with Deep SORT did not affect its performance.

Evaluation Of SOT Algorithms

Optical flow is used for performing SOT on a known car location. Its performance is evaluated using the metric given by Soleimanitaleb et al.[28]. The Euclidean distance between the center detected object and its ground truth link as shown in Figure 5 is plotted. This gives a measure similar to MOTP where the overlap between the bounding boxes is measured to find the precision of detection and mAP where the ground truth bounding box is compared with the detected box and returns a score. The other metric to evaluate the performance is using IOU over the frames of the video. IOU metric is used for checking how well our prediction bounding box aligns with our ground truth.



(a) (b)

Figure 5: Using Haar Cascade With Optical Flow (a) Center Position Error, And (b) IOU.

https://doi.org/10.52202/075106-0004

We can observe that as the position error increased after the 90th frame, the corresponding IOU value also decreased. The performance of optical flow in SOT is degrading due to the occlusion and scaling. As the features in the initial bounding box got occluded, the size of the bounding box changed and increased the IOU and hence increased the center position error.

CONCLUSION

We have presented a comprehensive performance evaluation of different detection and tracking algorithms for single object tracking (SOT) and multi-object tracking (MOT) of vehicles. The MOT algorithms used are SORT and Deep SORT. To evaluate their performance, the MOT metrics outlined by CLEAR MOT are used. These metrics give a clear idea about the performance of the two trackers. The algorithms are tested with the datasets of VisDrone with different camera views and lighting conditions. We have observed that in MOT, the deep learning-based algorithms are well suited for object detection while traditional algorithms perform better in tracking. Moving forward, training Deep SORT over a wide range of datasets may enhance its performance. Additionally, more algorithms can be tested for SOT and optical flow can be integrated with other algorithms to improve its performance.

NOTATION

MT	Mostly tracked
PT	Partially tracked
PT	Partially tracked
ML	Mostly lost
FM	Fragmentations
IDSW	ID switches
MOTA	Multiple Object Tracking Accuracy
t	Frame index
GT	Number of ground-truth objects
FN	False negatives
FP	False positives
Ct	Number of matches in frame t

REFERENCES

1. Liu Z, He Z, Wang L, et al. *VisDrone-CC2021: The Vision Meets Drone Crowd Counting Challenge Results*. http://www.aiskyeye.com/.

2. Butilă EV, Boboc RG. Urban Traffic Monitoring and Analysis Using Unmanned Aerial Vehicles (UAVs): A Systematic Literature Review. *Remote Sens (Basel)*. 2022;14(3). doi:10.3390/rs14030620

3. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol 1. ; 2001. doi:10.1109/cvpr.2001.990517

4. Lucas BD, Kanade T. ITERATIVE IMAGE REGISTRATION TECHNIQUE WITH AN APPLICATION TO STEREO VISION. In: Vol 2. ; 1981.

5. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: *Proceedings - International Conference on Image Processing, ICIP*. Vol 2016-August. IEEE Computer Society; 2016:3464-3468. doi:10.1109/ICIP.2016.7533003

6. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol 2016-December. ; 2016. doi:10.1109/CVPR.2016.91

7. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE; 2017:3645-3649. doi:10.1109/ICIP.2017.8296962

8. Zhuang X, Kang W, Wu Q. Real-time vehicle detection with foreground-based cascade classifier. *IET Image Process*. 2016;10(4). doi:10.1049/iet-ipr.2015.0333

9. Putri DIH, Martin, Riyanto, Machbub C. Object detection and tracking using SIFT-KNN classifier and Yaw-Pitch servo motor control on humanoid robot. In: *2018 International Conference on Signals and Systems, ICSigSys 2018 - Proceedings*. ; 2018. doi:10.1109/ICSIGSYS.2018.8373566

10. Moranduzzo T, Melgani F. A SIFT-SVM method for detecting cars in UAV images. In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. ; 2012. doi:10.1109/IGARSS.2012.6352585

11. Ward IR, Laga H, Bennamoun M. RGB-D Image-Based Object Detection: From Traditional Methods to Deep Learning Techniques. In: *Advances in Computer Vision and Pattern Recognition*. ; 2019. doi:10.1007/978-3-030-28603-3_8

12. Hsu SC, Huang CL, Chuang CH. Vehicle detection using simplified fast R-CNN. In: *2018 International Workshop on Advanced Image Technology, IWAIT 2018*. ; 2018. doi:10.1109/IWAIT.2018.8369767

13. Fan Q, Brown L, Smith J. A closer look at Faster R-CNN for vehicle detection. In: *IEEE Intelligent Vehicles Symposium, Proceedings*. Vol 2016-August. ; 2016. doi:10.1109/IVS.2016.7535375

14. Lou L, Zhang Q, Liu C, Sheng M, Liu J, Song H. Detecting and counting the moving vehicles using mask R-CNN. In: *Proceedings of 2019 IEEE 8th Data Driven Control and Learning Systems Conference, DDCLS 2019*. ; 2019. doi:10.1109/DDCLS.2019.8908877

 https://doi.org/10.52202/075106-0004

15. Chen Z, Guo H, Yang J, et al. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement (Lond)*. 2022;201. doi:10.1016/j.measurement.2022.111655

16. Zhang Y, Guo Z, Wu J, Tian Y, Tang H, Guo X. Real-Time Vehicle Detection Based on Improved YOLO v5. *Sustainability (Switzerland)*. 2022;14(19). doi:10.3390/su141912274

17. Liu Y, Lu Y, Shi Q, Ding J. Optical flow based urban road vehicle tracking. In: *Proceedings - 9th International Conference on Computational Intelligence and Security, CIS 2013*. ; 2013. doi:10.1109/CIS.2013.89

18. Gomaa A, Abdelwahab MM, Abo-Zahhad M, Minematsu T, Taniguchi RI. Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow. *Sensors (Switzerland)*. 2019;19(20). doi:10.3390/s19204588

19. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*. 2013;32(11). doi:10.1177/0278364913491297

20. Yang M, Jia Y. Temporal dynamic appearance modeling for online multi-person tracking. *Computer Vision and Image Understanding*. 2016;153. doi:10.1016/j.cviu.2016.05.003

21. Milan A, Leal-Taixe L, Reid I, Roth S, Schindler K. MOT16: A Benchmark for Multi-Object Tracking. Published online March 2, 2016. http://arxiv.org/abs/1603.00831

22. Hsieh MR, Lin YL, Hsu WH. Drone-based Object Counting by Spatially Regularized Regional Proposal Network. Published online July 19, 2017. http://arxiv.org/abs/1707.05972

23. Zhu P, Wen L, Du D, et al. *Detection and Tracking Meet Drones Challenge*. http://www.votchallenge.net/.

24. Xing X, Yongjie Y, Huang X. Real-Time Object Tracking Based on Optical Flow. In: *2021 International Conference on Computer, Control and Robotics, ICCCR 2021*. ; 2021. doi:10.1109/ICCCR49711.2021.9349376

25. Tan M, Pang R, Le Q V. EfficientDet: Scalable and efficient object detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ; 2020. doi:10.1109/CVPR42600.2020.01079

26. Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J Image Video Process*. 2008;2008. doi:10.1155/2008/246309

27. Wu B, Nevatia R. Tracking of multiple, partially occluded humans based on static body part detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol 1. ; 2006. doi:10.1109/CVPR.2006.312

28. Soleimanitaleb Z, Keyvanrad MA. *Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics*.

29. Pereira R, Carvalho G, Garrote L, Nunes U. *Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation and New Data Association Metrics*

https://doi.org/10.52202/075106-0004