Journal of Internet Services and Applications, 2024, 15:1, doi: 10.5753/jisa.2024.3809

MEDAVET: Traffic Vehicle Anomaly Detection Mechanism based on spatial and temporal structures in vehicle traffic

Ana Rosalía Huamán Reyna [**University of São Paulo** | ***arhuamanr@usp.br***]
Alex Josué Flórez Farfán [**University of São Paulo** | ***alex.josueff@usp.br***]
Geraldo P. Rocha Filho [**State University of Southwest Bahia** | ***geraldo.rocha@uesb.edu.br***]
Sandra Sampaio [**University of Manchester** | ***s.sampaio@manchester.ac.uk***]
Robson de Grande [**Brock University** | ***rdegrande@brocku.ca***]
Luis Hideo Vasconcelos Nakamura [**University of São Paulo** | ***nakamura@icmc.usp.br***]
Rodolfo Ipolito Meneguette [**University of São Paulo** | ***meneguette@icmc.usp.br***]

Institute of Mathematical and Computer Sciences, University of São Paulo (USP)
Av. Trabalhador São Carlense, 400 - Centro, São Carlos - SP, 13566-590, Brazil.

Received: 01 November 2023 • **Accepted:** 08 April 2024 • **Published:** 24 April 2024

Abstract Road traffic anomaly detection is vital for reducing the number of accidents and ensuring a more efficient and safer transportation system. In highways, where traffic volume and speed limits are high, anomaly detection is not only essential but also considerably more challenging, given the multitude of fast-moving vehicles, often observed from extended distances and diverse angles, occluded by other objects, and subjected to variations in illumination and adverse weather conditions. This complexity has meant that human error often limits anomaly detection, making the role of computer vision systems integral to its success. In light of these challenges, this paper introduces MEDAVET - a sophisticated computer vision system engineered with an innovative mechanism that leverages spatial and temporal structures for high-precision traffic anomaly detection on highways. MEDAVET is assessed in its object tracking and anomaly detection efficacy using the UA-DETRAC and Track 4 benchmarks and has its performance compared with that of an array of state-of-the-art systems. The results have shown that, when MEDAVET's ability to delimit relevant areas of the highway, through a bipartite graph and the Convex Hull algorithm, is paired with its QuadTree-based spatial and temporal approaches for detecting occluded and stationary vehicles, it emerges as superior in precision, compared to its counterparts, and with a competitive computational efficiency.

Keywords: Anomaly Detection, Vehicle Tracking, Computer Vision.

1 Introduction

Maintaining road traffic safety is paramount for the protection of individuals - by reducing the risk of accidents and promoting a sense of security - as well as society at large [Huk and Kurowski, 2022], as effective traffic safety measures contribute to the smooth functioning of transportation systems and minimize accident-related injuries and fatalities, thereby reducing the burden on healthcare systems. Despite concerted efforts, however, challenges in ensuring road traffic safety have persisted, as World Health Organization's Global Status Reports on Road Safety in recent years have highlighted. According to the 2018 report of Health [2018], the number of annual road traffic deaths reached 1.35 million. Among non-fatal victims, between 20 and 50 million people suffer from permanent consequences. More than half (54%) of all traffic-related deaths and injuries involve vulnerable road users such as pedestrians, cyclists, motorcyclists, and their passengers.

This information has sparked significant mobilization within the global community, leading to the development of action plans aimed at enhancing road safety and, consequently, reducing the number of traffic casualties. A prime example of such an initiative is the Second Decade of Action for Road Safety, defined by the UN General Assembly in 2020 [NATIONS., 2020]. This initiative aims to decrease traffic injuries and fatalities by at least 50% worldwide between 2021 and 2030.

More recently, the 2023 report of Health [2023] showed a slight decrease in annual road traffic deaths to 1.19 million. While this indicates that efforts to improve road safety are having an impact, it also emphasizes that the price paid for mobility remains excessively high. Therefore, urgent action is needed if the goal of halving road traffic deaths and injuries by 2030 is to be achieved.

Road traffic safety is ensured by the continuous monitoring and inspection of road traffic by authorities. This process requires the analysis of data captured through a variety of devices, including mounted cameras [Ferrante *et al.*, 2021; Meneguette and Boukerche, 2020], which are capable of recording road images in real-time. This data is primarily used to control traffic speed and congestion, as well as detect situations that pose a risk to road users, such as the presence of stranded or abandoned vehicles on the highway [Gomides *et al.*, 2022; Pereira *et al.*, 2020]. These activities require effective real-time monitoring and detection of events, as well as the tracking of anomalies in traffic patterns, such as sudden or unexpected changes in the flow of vehicles [Huk and

Kurowski, 2022; Meneguette *et al.*, 2012].

Traffic anomaly detection faces a myriad of challenges, including the presence of numerous fast-moving vehicles that change dynamically, occlusions caused by various objects, changes in illumination, adverse weather conditions, cluttered backgrounds, variable number of targets, etc. These factors can lead to errors in traffic monitoring and tracking due to human limitations, such as fatigue, distraction, and slow perception and reaction times. Moreover, these challenges require the rapid analysis and interpretation of large volumes of data [Pawar and Attar, 2021].

Computer vision systems have played a vital role in traffic monitoring and anomaly detection [Montanari, 2016] by processing complex data, typically in the form of images or videos obtained from roadside cameras. These systems are capable of distinguishing between anomalous and normal traffic behaviour as well as warning authorities of potential risks [Djenouri *et al.*, 2022], e.g., traffic accidents [Zhang *et al.*, 2021a]. Although video structure analysis is fundamental to successful traffic analysis [Zhao, 2021], it remains a very challenging problem due to the complexity of traffic scenarios [Pawar and Attar, 2021].

Despite the successful use of computer vision technology in various industrial applications, such as robotic vision, human-machine interfaces, information retrieval, medical image analysis, security systems, and traffic surveillance systems [Santos, 2014; Meneguette and Boukerche, 2017], computer vision or perception systems are strongly impacted by adverse conditions [Ge *et al.*, 2023]. Therefore, despite significant advances in computer vision technology in recent years, the development of more robust and reliable perception strategies is still necessary [Liu *et al.*, 2023].

Within this context, this article proposes a vehicle tracking model that uses computer vision to detect highway traffic anomalies. The model employs an innovative mechanism that leverages spatial and temporal structures for high-precision traffic anomaly detection and is implemented within the MEDAVET computer vision system. The main contributions are summarized as follows:

- A description of the proposed MEDAVET model, including the spatial and temporal structures that support its anomaly detection mechanism, which uses bipartite graphs and spatial-temporal positioning to optimize the anomaly detection and vehicle tracking processes.
- A thorough evaluation of MEDAVET in its ability to analyze traffic from road traffic video scenes, tracking both moving and stationary vehicles, and its ability to detect traffic flow anomalies, using the UA-DETRAC and Track 4 benchmarks. In this study, an anomaly is defined as a vehicle that has remained stationary for longer than the time allotted by the traffic light on a main road.
- A detailed comparison of the performance of MEDAVET with that of an array of state-of-the-art systems using multiple metrics.

This article is structured as follows. Section 2 describes recent works related to our approach and analyzes them comparatively. Section 3 presents the proposal developed for detecting anomalies in vehicle traffic on highways. Section 4 introduces our analyses, analyzing and discussing the obtained results of the developed detection model. Finally, Section 5 presents the final considerations highlighting future work directions.

2 Related Works

The use of computer vision to solve the most diverse problems is broad. We narrowed its scope by considering related works where computer vision is employed to solve problems associated with vehicle traffic.

In Bafghi and Shoushtarian [2020], the objective was to present a system for tracking multiple vehicles on a highway using appearance models and visual tracking. The method consists of three steps: vehicle detection, vehicle appearance modelling and vehicle tracking. In the first step, the Mask R-CNN detection algorithm [He *et al.*, 2017] identifies the presence of vehicles in each frame of the video. In the second step, the *SIFT* [Lowe, 2004] appearance model and colour histograms are created based on its visual characteristics from a reference image of each vehicle. In the third step, appearance models identify and track vehicles detected in multiple subsequent images over time. To achieve this, edge detection and visual feature matching techniques are used, such as bipartite graphs. A motion model was used to obtain greater accuracy in estimating the positions of objects. With these two models, appearance and motion, the edge weights are found in a linear combination. The method is evaluated on the UA-DETRAC dataset and shows promising results in accuracy and efficiency in detecting and tracking multiple vehicles moving on the highway.

In Bai *et al.* [2019], an anomaly detection system is proposed that includes three modules: Background modelling module, perspective detection module and spatiotemporal matrix discrimination module. Background modelling analyzes the traffic flow to obtain the Unsupervised road segmentation results based on traffic flow analysis that eliminates interference from off-road vehicles. The detection perspective model gets the perspective map by the first detection result, which is done by the Faster R-CNN model [Fan *et al.*, 2016], and together, the image is cropped into a uniform scale for different vehicles and re-detection. Finally, all anomalies are obtained by building a spatial-temporal information matrix with the detection results. Furthermore, all anomalies are combined through NMS and the re-identification model, including spatial and temporal dimensions.

In Li *et al.* [2020], a traffic anomaly detection method based on vehicle detection and tracking was developed. The Faster R-CNN algorithm enabled vehicle detection, and the DeepSORT algorithm supported tracking to obtain the vehicle trajectory. DeepSORT is an object tracking framework and is an extension of SORT (Simple Real-time Tracker) [Hou *et al.*, 2019]. With the detection and tracking results, the MoG2 model was presented, based on a Gaussian mixture model [Reynolds, 2009], which aims to remove moving vehicles and only analyze stationary vehicles. Then, a new mask extraction mechanism is implemented based on the difference in frames and the vehicle's tracking trajectory to remove secondary roads, such as parking lots, and thus avoid false detections. Next, a multi-granularity tracking structure contains a box-level tracking branch and a pixel-level tracking branch. Each component contributes to capturing abnormal abstractions at different levels of granularity to model abnormal concepts. Finally, a backtracking and anomaly fusion optimization method is proposed to refine the abnormal predictions, which can significantly improve the robustness and accuracy of the results.

In Zhao [2021], a simple and efficient structure is proposed that includes three steps: Pre-processing, A dynamic track module, and Post-processing. The pre-processing step aims to generate candidate anomalies and comprises four parts: video stabilization, background modelling, vehicle detection, and mask generation. Video stabilization aims to correct camera movement oscillations that occur in the acquisition process through software techniques such as point matching based on Good Features to Track (GFTT) [Shi *et al.*, 1994] and a sparse optical flow is calculated to generate frame-by-frame transformations, and thus improve visual quality and improve end applications such as vehicle detection and tracking. In background modelling, a background subtraction approach based on Gaussian mixture (MoG) is used. The objective is to compare the results of forward and backward background subtraction, which shows that the results of backward background subtraction make stationary vehicles clearer. Background-to-front subtraction is used as an auxiliary method to obtain a more accurate start time of the anomaly. Faster R-CNN is used to detect vehicles. When generating a mask, it is necessary to filter static vehicles on secondary roads and parking lots to detect anomalies on primary roads – segmenting the hypothetical anomalous regions of the mask is essential, which uses a mask extraction method based on movement and a mask based on trajectory. The dynamic tracking step searches for and locates the onset time of anomalies using vehicle movement patterns and spatiotemporal status. Finally, post-processing is used to fine-tune the time limit of anomalies.

The related works described above serve as a direct influence on the creation of our proposed approach. Table 1 summarizes and compares such works. Each work contributed ideas that can be applied to vehicle detection and vehicle tracking, whereas our proposed work focuses on techniques that contribute to detecting traffic anomalies.

In Bafghi and Shoushtarian [2020]'s work presented two distinct methods for extracting object features. The first method used SIFT and colour histogram features in each image to evaluate the similarities between neighbouring frames and different objects. On the other hand, the second method employed deep features obtained from the Mask R-CNN object detection network to achieve the same objective. In this work, we take bipartite graphs as a guide for object tracking but choose OPEN CLIP and structural similarity to determine connection weights. Unlike SIFT, which requires colour histograms to enhance object features, OPEN CLIP can extract features effectively in various scenarios. It is designed to understand and correlate visual and linguistic information, enabling machines to process visual media and text together. This ability to associate text and images makes it a valuable tool in several applications. Furthermore, structural similarity plays an important role when comparing and evaluating the quality of images resulting from different processes and transformations. It offers an objective metric to measure the structural similarity between original images and their modified versions.

In Li *et al.* [2020]'s work presented a solution to detect traffic anomalies using computer vision methods such as background subtraction, image segmentation, and modularized components to track vehicles at box and pixel levels. In this work, we use the idea of background subtraction and segmentation to remove vehicles on secondary roads. However, we chose to use Convex Hull to generate areas of both movement and lack of movement. We choose busy areas to analyze vehicle behaviour and identify possible vehicles stopped on these roads. An essential distinction between this work and the previous one is that, in our method, motion areas are generated during the tracking process. In the previous work, background subtraction and segmentation were performed separately in two different processes, which resulted in a higher computational cost.

In Bai *et al.* [2019]'s work introduced a solution for detecting traffic anomalies, analyzing anomaly events based on vehicle dynamic information. Specifically, they used six spatiotemporal information matrices to identify the start and end time of the detected anomaly. This information was related to the pixel level and was associated with time. Our work incorporates time-related information to determine when a vehicle remains on the scene. To do this, we monitor whether the vehicle is still present at the scene or has left it. If the vehicle is in the scene, we calculate its speed to determine whether it is stopped or moving. When the speed is low, we interpret that the vehicle is stationary, and, in this case, we create hypothetical positions for this object, aiming not to lose important information about it, considering it as an object with a possible anomaly. A distinction between our method and the previous method is that our process incorporates temporal information using a unified approach.

In contrast, the previous method uses background modelling and perspective detection as separate processes to obtain information about the start and end of potential anomalies. Our method offers a more integrated approach, using data from vehicle trajectories to identify the beginning and end of possible anomalies. It results in a more complete and efficient vehicle behaviour analysis in a single process.

Finally, the work by Zhao [2021] presents three stages to detect traffic anomalies: pre-processing, dynamic tracking, and post-processing. In our work, we are inspired by the pre-processing stage, as we face challenges related to noise in the data, such as camera instability and variations in lighting. However, we chose to approach anomaly detection differently, using object tracking in conjunction with a data structure known as QuadTree, in addition to a temporal approach. The QuadTree is employed to compare the positions and characteristics of nearby objects, while the temporal structure analyzes whether these objects fall into the anomaly category. This approach makes our method robust in detecting anomalies compared to the other method, which requires additional post-processing to adjust the temporal boundaries of anomalies. In short, our strategy simplifies the detection process and improves the effectiveness of identifying traffic anomalies.

Table 1. Summary of Related Works.

Proposals	***Dataset***	**Detection**	**Tracking**	**Traffic anomalies**
[Bafghi and Shoushtarian, 2020]	UA-DETRAC	*Mask* R-CNN	Appearance model and Visual Object Tracking	—
[Li *et al.*, 2020]	*Track*4	*Faster* R-CNN	—	Box and Pixel Level Tracking Model
[Bai *et al.*, 2019]	*Track3*	*Faster* R-CNN	—	Perspective Relationship Detection Model
[Zhao, 2021]	*Track*4	*Faster* R-CNN	—	Dynamic Tracking Model
This paper	**Different datasets**	**YOLOv7**	**Vehicle Tracking Using areas of interest**	**Area Anomaly Detection Spatiotemporal Interest**

Figure 1. MEDAVET application scenario.

3 MEDAVET: Traffic Vehicle Anomaly Detection

This section presents an anomaly detection mechanism called MEDAVET - Traffic Vehicle Anomaly Detection Mechanism. MEDAVET is based on detecting and tracking vehicles on urban roads, aiming to detect anomalies in highway traffic. Our approach focuses on identifying vehicles that remain stationary on main road lanes for more than one minute. In the road context, a ”main road” refers to a road of greater capacity and importance within a road network. It is generally designed to accommodate a large volume of traffic. Our research aims to significantly improve road safety and traffic efficiency by identifying anomalous situations that may pose risks, such as prolonged congestion or incidents that require rapid intervention.

Therefore, in this work, we consider that cameras monitor highways along them (Figure 1) and that the information captured is sent to a center where processing is carried out. It is essential to highlight that the cameras are fixed in the datasets. Thus, MEDAVET runs in the control center, allowing an operator to take action on time, such as calling a vehicle to check what is happening to the driver.

The following sections describe the components and functionalities that make up MEDAVET.

3.1 Overview

In this study, we implemented a video vehicle monitoring system comprising a video preprocessing process for camera stabilization and enhancement of visual characteristics. We then use the preprocessed frames to detect and track vehicles (Figure 2). To do this, we use an object detection tool to detect vehicles in the captured frames. Therefore, an object in this work would be a vehicle in the image. After object detection, we use graph theory to create a structure for representing and analyzing vehicle trajectories, allowing vehicles to be tracked over time. Based on the output graph in the vehicle detection and tracking component, we perform anomaly detection that will check the time that the vehicle will be out of mobility and the place that the vehicle is stopped to perform the inference, whether it is normal behaviour or not. In the following subsections, we will describe each component in more detail.

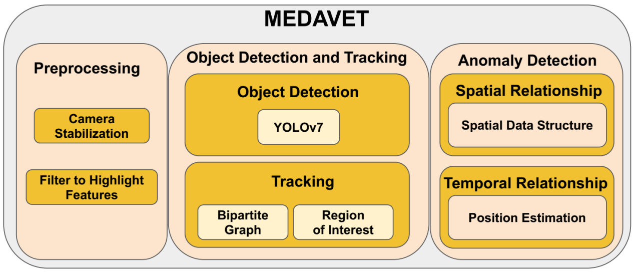

Figure 2. MEDAVET Overview.

3.2 Preprocessing

The image capture process can introduce undesirable noise, leading to issues such as undetected frames, unwanted camera movement, and the presence of artifacts, as demonstrated in Figure 3. These disturbances can negatively impact the functionality of vehicles, causing disruptions in the detection and tracking process, potentially resulting in multiple detections and the assignment of incorrect IDs to vehicles. To address these challenges, a pre-processing step becomes imperative. During our data analysis, we identified the need for camera stabilization and noise-filtering techniques to emphasize relevant image features.

The process of camera stabilization plays a crucial role in enhancing the visual quality of videos by mitigating the adverse effects of camera movement. In this approach, the goodFeaturesToTrack algorithm identifies and selects distinctive features within a frame, serving as crucial references for subsequent tracking, as illustrated in Figure 4. The Lucas-Kanade Optical Flow algorithm [P *et al.*, 2023] is then employed to track these key points across consecutive frames,

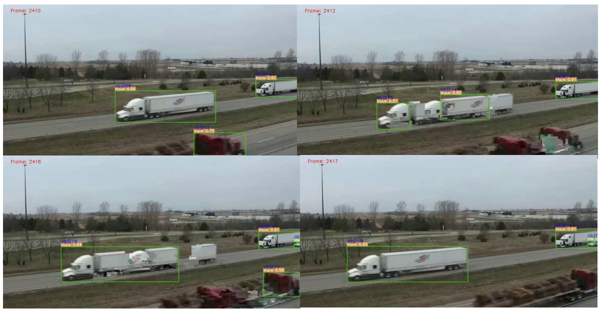

Figure 3. Images without preprocessing.

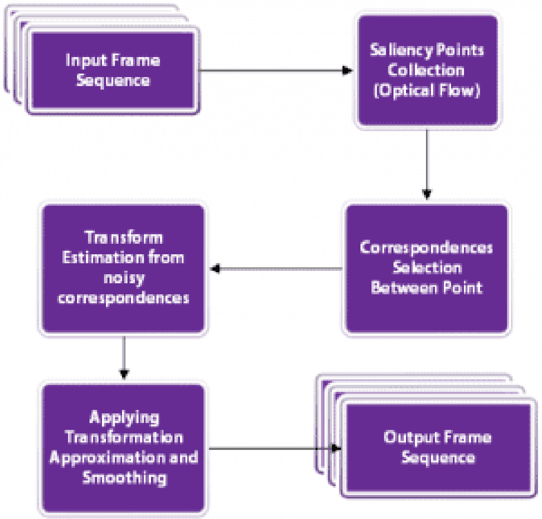
Input Frame Sequence
Saliency Points Collection (Optical Flow)
Correspondences Selection Between Point
Transform Estimation from noisy correspondences
Applying Transformation Approximation and Smoothing
Output Frame Sequence

Figure 4. Camera stabilization block diagram.

enabling precise determination of their displacements over time. Utilizing these displacements, it becomes feasible to compute a rigid (Euclidean) transformation, encompassing translation, rotation, and scale information. This correction is applied to each frame, resulting in a smoother and more stable final video.

We use the inter-frame motion estimation performed in the previous step to filter noise in the motion trajectory. In this step, we seek to determine the movement trajectory by incrementally accumulating the estimated differential movement between consecutive frames; it adds up the movement between frames to calculate the overall trajectory. The ultimate goal is to smooth this trajectory to make it more stable. For this smoothing, we use a moving average filter, which, as the name suggests, replaces the value of a function at a point with the average of the values of its neighbours. We apply this smoothed trajectory to obtain smoother motion transformations that can be applied to video frames to stabilize them. This stabilization is achieved by finding the difference between the smoothed and original trajectories and then adding this difference to the original transformations. The process involves iterating through the frames and applying these transformations, resulting in a final video that is stabilized and free from unwanted movements.

3.3 Vehicle Detection and Tracking

The object detection and tracking system is divided into two fundamental stages. We start by detecting objects in each frame of the videos using advanced computer vision algorithms, such as convolutional neural networks. We then create bipartite graphs to connect corresponding detections between consecutive frames, allowing continuous tracking of moving objects. To further improve efficiency, we delimited an area of interest around moving vehicles. Finally, the system assigns IDs to the tracked vehicles and records their trajectories over time. These trajectories provide valuable information for analyzing and understanding vehicle behaviour in the context of the video (Figure 5).

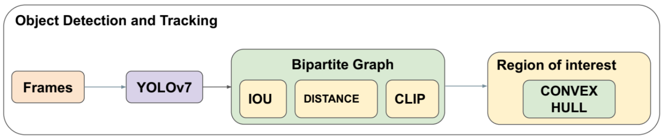

Figure 5. Detection and Tracking Component.

Initially, the objects in the image are detected; we use the YOLOv7 algorithm [Wang *et al.*, 2023] to perform object detection. The choice of YOLOv7 is due to its implementation of a new training algorithm called CrossEntropyLossWithLogits, which stands out for being faster and more efficient than the algorithms used in previous versions of YOLO. This optimization results in significantly reduced training time. Furthermore, this version incorporates different weights, trained on a vast image dataset, and can detect various objects, including vehicles. For this project, we used the YOLOv7-W6 model, which has proven highly effective in our quest for accurate object detection across multiple sizes, from minor to large-scale objects.

After object detection, the bipartite graph modulation begins, allowing the objects to be efficiently tracked over time and establishing connections between detections in different frames. For this detection, we consider the following:

- Each vertex in one of these graphs represents a specific detection of an object in a frame. Therefore, if we have frames $F1$ and $F2$, each vertex in $F1$ represents an object detection from that frame, and each vertex in $F2$ represents a corresponding detection in the subsequent frame $F2$.
- Each set of vertices represents a video frame; that is, each vertex represents a vehicle detected by the YOLO algorithm.
- The weights are used to determine how the edges will be connected between the vertices of each consecutive frame and the bipartite graph.

Therefore, for every two consecutive frames, we create a bipartite graph so that all components in the current frame are connected to components in the next frame. Then, the set F is divided into n disjoint sets, each representing a frame, and the vertices in each set represent objects, which are vehicles in the work.

For a better understanding, let us explain it in mathematical terms:

$$F = \cup_{i=1}^{n} F_i, \text{with } F_i \cap F_j = \emptyset, \forall i, j. \tag{1}$$

where F_i represents the i-th frame of the set F.

$$F_i = \left\{ f_c^i \mid c \in \{1, \ldots, p\} \right\} \tag{2}$$

Then, F_i is the set of vertices of frame i, where f_c^i denotes the c-th vertex. Having the set of frames and the set of vertices p, the set of edges can be defined as:

$$G = \{(f_c^i, f_q^{i+1}) \mid c \neq q\} \tag{3}$$

where f_c^i represents the c-th vertex of frame i and f_q^{i+1} represents the q-th vertex of frame $i + 1$

To calculate the weights, we use IOU (Intersection Over Union) metrics and feature extraction, with the aim of ensuring that the edges only connect vehicles with high similarity.

Thus, the metrics are defined as follows:

$$IOU(f_c^i, f_q^{i+1}) = \frac{Area((f_c^i) \cap Area(f_q^{i+1})}{Area((f_c^i) \cup Area(f_q^{i+1})} \geq limiou \tag{4}$$

where $limiou$ is a threshold given by a constant value.

To increase precision and assist the OR method, we also use the distance between the components of each vertex, giving the equation below:

$$dist(f_c^i, f_q^{i+1}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \leq limdis \tag{5}$$

where (x_1, y_1) is the position of the center of f_c^i and (x_2, y_2) is the position of the center of f_q^{i+1}, and $limdis$ is a threshold given by a constant value

To determine the vehicle similarity, we used OpenAI CLIP, which belongs to the Transformers family of models. Thus, the similarity metric we employ is defined as follows:

$$sim(f_c^i, f_q^{i+1}) \geq limsim \tag{6}$$

where $limsim$ is a threshold given by a constant value. Combining the three parameters, we have:

$$w(f_c^i, f_q^{i+1}) = \alpha \times IOU(f_c^i, f_q^{i+1}) + \beta \times sim(f_c^i, f_q^{i+1}) \tag{7}$$

With α and β measurement parameters, the weights are calibrated between 0 and 1.

Within this work, we are mainly focused on tracking areas of interest along roads where road accidents occur, some of which may be caused by vehicle movement. Our main area of interest covers these specific areas of road movement. The goal consists of continuously monitoring traffic in these regions to capture events such as accidents, incidents, or driving behaviours that could lead to dangerous situations. We use the Convex Hull algorithm to define this area of interest in the image. This algorithm creates convex polygons surrounding the moving areas of the highway lanes. The Convex Hull is a convex envelope encompassing a set of points in the plane or a multidimensional space.

Creating these polygons allows us to clearly define the regions where moving vehicles are present, as illustrated in Figure 6. This way, we can exclude areas beyond the shoulder, such as gas stations and other establishments adjacent to the highway, focusing our analysis on areas directly related to traffic flow. This procedure helps avoid confusion with vehicles on secondary roads. Secondary roads on a highway typically have less traffic capacity and are intended for specific purposes, such as access to parking lots.

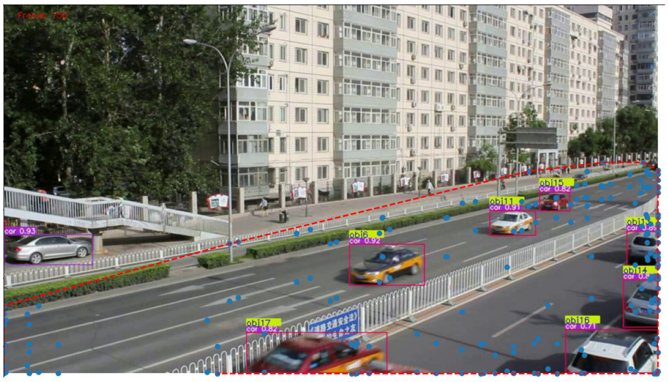

Figure 6. Vehicles within the region defined via ConvexHull.

It is essential to highlight that video frames undergo a pre-processing step before being passed to the YOLO algorithm. This pre-processing improves the quality of the images and the location of objects. After this preparation, the frames are forwarded to the YOLO algorithm responsible for object detection. The tracking phase begins with identifying objects in each frame, as shown in Algorithm 1. For each pair of consecutive frames, a bipartite graph is created. The vertices of each frame are analyzed to check whether they meet the conditions defined by the bipartite graph. The vertices that meet these conditions receive a unique ID (line 6 of Algorithm 1). Otherwise, the similarity, IOU, and distance conditions are checked (lines 8 to 10). If these conditions are met, the frame and vertex information are assigned to the corresponding object in the list (line 12). Otherwise, a new object is created (line 14). The values used in line 11 were objectified through experiments described in Section 4.

To minimize the number of vehicles to be analyzed, we use the Convex Hull algorithm to surround the area of moving vehicles and create a new list of objects that are in the area of interest (list_IA_obj). Thus, if there is a vehicle in the area of interest (lines 17 to 19 of Algorithm 1), an ID is assigned to that vehicle, excluding vehicles or other objects that are outside that area.

3.4 Anomaly detection

The anomaly detection component aims to detect vehicles stopped on the main roads. We implemented two essential structures: one focused on spatial analysis and the other on temporal analysis (Figure 7). Spatial analysis performs spatial searches, allowing the locations of objects to be obtained, especially to identify stationary vehicles. On the other hand, temporal analysis is crucial in dealing with challenges like object occlusion, ensuring we do not lose information about the object. Temporal analysis makes it possible to identify objects that have left the scene or are immobilized on the road for a prolonged period or even when the detector does not detect them. This combined approach allows us to detect stopped vehicles on busy roads, providing a robust solution for identifying anomalies in road traffic.

In this last step, data from objects (vehicles) within an area of interest calculated by Convex Hull is received. However,

Algorithm 1 Vehicle tracking

```
Input: frames
Output: list_IA_obj, trajectory_obj
i = 0
while $i \leq n\_frames - 1$ do
    Graf[i] = Get_obj($F_i$, $F_{i+1}$)
    obj = Graf[i].get_obj
    if Is_vértice ($obj_i$,$obj_{i+1}$) then
        update(list_obj($obj_i$,$obj_{i+1}$)
    else
        IOU = IOU($obj_i$,$obj_{i+1}$)
        dist = dist($obj_i$,$obj_{i+1}$)
        sim = sim($obj_i$,$obj_{i+1}$)
        if $IOU \geqslant 0.4$ and $dist \leqslant 30$ and $sim \geqslant 0.6$ then
            update(List_obj($obj_i$,$obj_{i+1}$))
        else
            list_obj.add(obj)
        end if
    end if
    if Is_obj(i) and Is_AI(obj) then
        list_IA_obj.add(obj)
    end if
end while
```

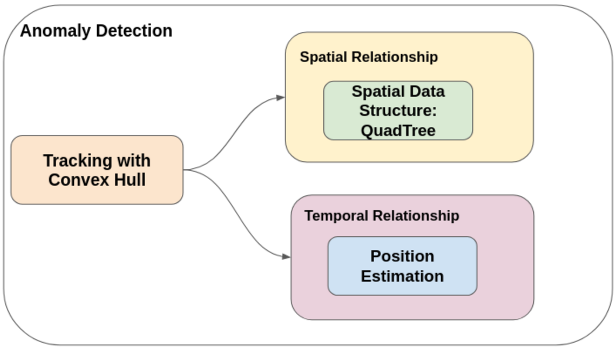

Figure 7. Anomaly detection component.

it was observed that when the vehicle is stopped at the scene, another vehicle can overlap the stopped vehicle, thus making a more accurate analysis difficult. Therefore, for the spatial component, we used the QuadTree method to improve spatial data organization within the delimited area. QuadTree performs a recursive division of the bounded region into smaller quadrants (Figure 8), creating a hierarchical representation of spatial information. This approach allows the analysis of vehicles that are close in terms of location and excludes regions where analysis is not necessary.

Furthermore, QuadTree plays an essential role in analyzing vehicle behaviour. By hierarchically grouping spatial information, we can identify movement patterns, average speeds, vehicle interactions, and other essential aspects of vehicle behaviour. This grouping contributes to a deeper and more refined understanding of the traffic landscape, enabling effective anomaly detection.

Therefore, for the space stage, QuadTree proves a reduction in computational cost and in analyzing the behaviour of vehicle trajectories, especially for stationary vehicles. It analyzes objects in the current frame about nearby objects in previous frames, considering a radius ’r’ as a limit for comparing objects in previous frames. The QuadTree is divided into four subtrees, each representing a spatial region of the field of view, and chooses the region where vehicles are close to the vehicles being analyzed. This data structure improves vehicle tracking and is used with another data structure dedicated to temporal analysis.

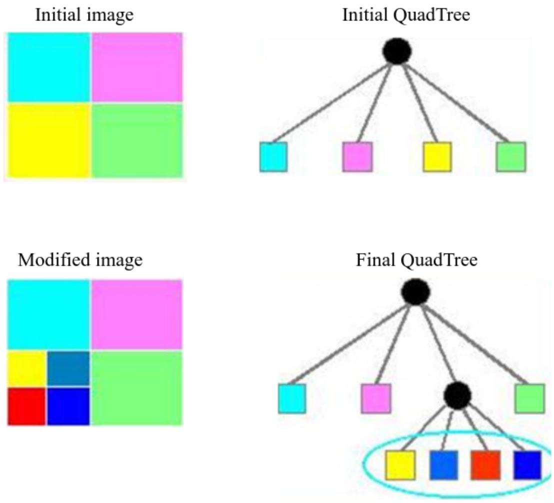

Figure 8. QuadTree construction process.

On the other hand, the temporal data structure handles situations where vehicles may be temporarily out of the scene. It approximates the positions of temporarily unavailable vehicles, ensuring we keep information about them. Furthermore, it analyzes the time a vehicle is in the scene, which is essential to avoid reassigning existing IDs to new vehicles that enter the field of view. This approach prevents vehicles parked in the main scene from being mistakenly re-identified, allowing us to keep a complete record of these vehicles and analyze whether or not there are anomalies based on dwell time. This temporal analysis is essential to identify vehicles representing an anomalous situation, such as prolonged congestion or an unscheduled stop on a busy road.

To deal with the temporal mechanism, we use the vehicle speed in each frame, checking whether the vehicle is moving or stationary. If the vehicle is stopped, we start monitoring this object; if it remains stopped for around 1800 frames, corresponding to 1 minute that the vehicle will be stopped on the highway, it will be classified as a vehicle with abnormal behaviour. A vehicle stopped on the highway can cause a severe accident.

The anomaly detection method, as summarized in Algorithm 2, receives the tracking of objects within an area of interest and their trajectory as input. With this information, the anomaly detection method will use the QuadTree to analyze nearby objects in previous frames (line 3 of Algorithm 2). To investigate proximity, an analysis of similarities and distance threshold criteria and IOU is carried out with previous objects in previous frames about the current frame (lines 4 to 7 of Algorithm 2). The object information is updated if these criteria are met (line 8 of Algorithm 2).

However, if the criteria are not met, the timing structure checks the object’s position to determine whether it is still in the scene. To do this, we construct a rectangle and check if the object is inside. If it is not, we assume it has left the scene

Algorithm 2 Anomaly Detection

```
Input: list_IA_obj, trajectory_obj
Output: list_ano
 1: for veh in trajectory_obj do
 2:     if veh in list_IA_obj then
 3:         quadtree = QuadTree(veh)
 4:         IOU = IOU(quadtree.obj)
 5:         dist = dist(quadtree.obj)
 6:         sim = sim(quadtree.obj)
 7:         if $IOU \geqslant 0.4$ and $dist \leqslant 30$ and $sim \geqslant 0.6$ then
 8:             update(List_Ia_obj(quadtree.obj))
 9:         else
10:             if Is_scena(quadree.obj) then
11:                 update(List_Ia_obj(quadtree.obj))
12:             else
13:                 if Velocity_Zero(quadtree.obj) then
14:                     if  Is_TheSameID(quadtree.obj) then
15:                         timestamp ++
16:                     end if
17:                 end if
18:             end if
19:         end if
20:         if timestamp ==1800 then
21:             list_ano.add(quadtree.obj)
22:         end if
23:     end if
24: end for
```

(lines 9 to 12 of Algorithm 2). However, if it is present, we analyze its speed over the last 100 frames. If the speed is deficient, almost zero, we infer that the vehicle is stopped at the scene. In this case, we understand that the vehicle is hidden, and in order not to lose information about its trajectory, we make hypothetical estimates in the next frame (lines 13 to 14 of Algorithm 2).

Again, we use QuadTree to evaluate the proximity and similarity of the approximated object to nearby objects in previous frames, keeping its ID along with all previous information. Finally, we consider the number of frames in which the vehicle was in the scene. If this quantity exceeds 1800 frames, we classify it as an anomaly (lines 18 to 19 of Algorithm 2).

To make the algorithm more understandable, imagine a set of consecutive frames. In the first frame, we can visualize a stopped vehicle and another vehicle overlapping the stopped vehicle (Figure 9). The analysis process starts with the first frame, using the QuadTree to examine the nearby objects in the previous frames. QuadTree is a technique that recursively divides the image region into smaller quadrants, creating a hierarchical representation of the spatial information of objects. We use similarity, distance, and IOU metrics to evaluate whether current objects are similar to objects in previous frames. If the current objects do not resemble the nearby objects in the previous frames, we start using the temporal data structure to check if the vehicles are still in the scene.

We define a rectangle inside the frame, with margins of -10 pixels on each side of the rectangle. If vehicles are within this rectangle, we assume they are still in the scene (frames two and three) but may need to be tracked correctly. In this case, we estimate their trajectories by calculating their speeds over the last 100 frames. If the speed is deficient, indicating that the vehicle is practically stopped, we assume it is hidden in the scene.

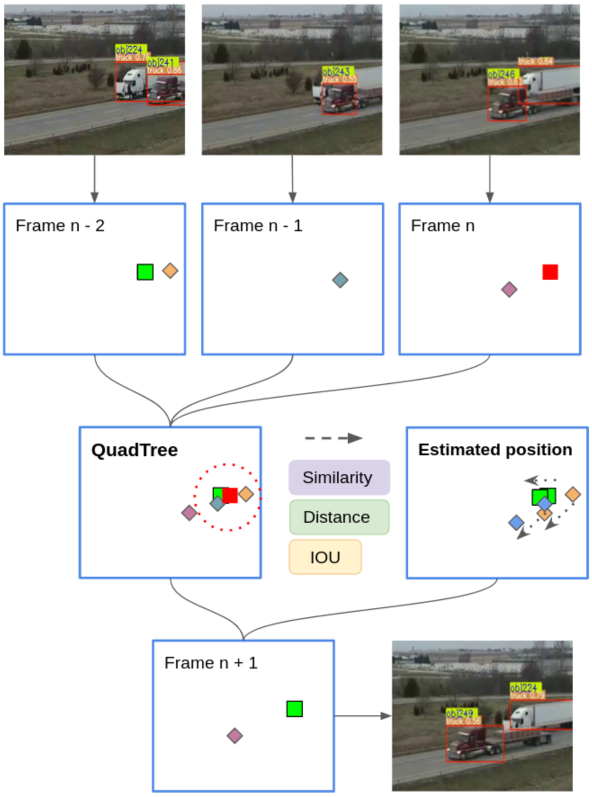

Figure 9. Example of the spatiotemporal relationship.

After estimation, we apply QuadTree again, using similarity, distance, and IOU metrics, to evaluate the proximity and similarity of the estimated object with nearby objects in previous frames. We keep the same ID and all previous object information throughout this process (last frame). To classify the situation as an anomaly, we count the number of frames in which the vehicle was present in the scene. If this quantity is more significant than 1800 frames, we consider the situation an anomaly. This approach allows us to identify situations where a vehicle remains on scene for an unusually long time, which may indicate an anomaly.

4 Performance Analysis

In this section, we describe in detail our experiments, presenting specific information about the proposed method for tracking, and detecting vehicle anomalies. This method incorporates an innovative approach that combines computer vision, machine learning, and deep learning algorithms to detect vehicles, track their trajectory, and identify anomalous behaviours, thus contributing to a safer and more efficient environment in transport and mobility.

4.1 Implementation Details

This work was implemented in the Python programming language, using version 3.11, and executed on a machine with the Linux operating system (Ubuntu 20.04). The hardware used to support development includes an 8-core Intel Core i7 processor, 16GB of RAM, and an NVidia RTX-3060 card with 12GB of video memory. This hardware configuration provided efficient performance during all phases of work implementation, guaranteeing the necessary processing capacity for the tasks of this work.

4.2 Object Tracking

The UA-DETRAC [Wen *et al.*, 2015] dataset was used to implement the tracking method. UA-DETRAC is a challenging benchmark for evaluating vehicle detection and tracking algorithms. Composed of 10 hours of videos recorded on the streets of China, this dataset includes a total of 140,000 frames, recorded at a rate of 25 frames per second (fps) and with a resolution of 960 × 540 pixels. This dataset covers the presence of 8,250 vehicles distributed across all frames.

4.2.1 Assessment Metrics

Evaluation metrics are intended to measure performance in the analysis of object-tracking systems. This work focuses on two essential areas: multi-object tracking and vehicle speed estimation. We use the CLEAR MOT metrics to evaluate object tracking, as proposed by [Bernardin and Stiefelhagen, 2008], which has several performance aspects. The principal metric we employ is MOTA (Multi-Object Tracking Accuracy), which is fundamental for the overall assessment of multi-object tracking performance. MOTA takes into account the number of false positives (FP), false negatives (FN), and identity changes (IDS) about Ground Truth (GT). In essence, MOTA quantifies how well our tracking system correctly detects and follows objects compared to actual annotations. A higher MOTA value indicates superior tracking performance with fewer object detection and identification errors. The formula for calculating MOTA is as follows:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t GT_t} \qquad (8)$$

Furthermore, we use metrics such as MOTP (Multi-Object Tracking Precision), which characterizes the misalignment between the annotated and predicted bounding boxes. MT (Mostly Tracked) and ML (Mostly Lost) are used to evaluate the continuity of the trajectory of tracked objects. Furthermore, identity switches (IDS) are used to measure the tracking quality regarding object identification. To evaluate computational performance, we also consider the number of frames processed per second (Hz), which describes the overall inference speed of the system.

4.2.2 Parameter Evaluation

To determine the appropriate values for the parameters of our method, we carried out a series of experiments, exploring different values of similarity, distance, and IOU, as seen in Table 2. We start with fixed settings, keeping a high and constant value for similarity (Sim) and a low and continuous value for distance (Dist) while varying the IOU values in the first three rows. However, in the last three lines, we decided not to fix specific values for these parameters, allowing us to observe the significant impact of distance when the IOU is reduced due to the substantial size of a few vehicles and their high speed. The most promising results from our experiments are highlighted in the last row of the table. Notably, it is observed that false negatives (FN) and identification misses (IDS) are lower compared to the other configurations.

Table 2. Assessment of tracking parameters.

Parameters			Results			
Sim	IOU	Dist	FP ↓	FN ↓	IDS ↓	MOTA ↑
0.9	0.9	5	35	2346	27	0.7
0.9	0.8	5	276	485	200	60.4
0.9	0.7	5	390	122	11	78.4
0.8	0.6	10	395	126	7	78.2
0.7	0.5	20	378	129	5	78.9
0.6	0.4	30	382	125	4	78.9

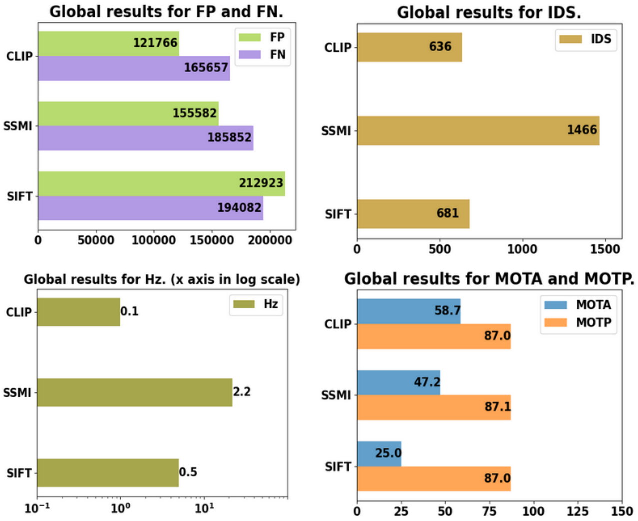

Figure 10. Feature extractor results.

However, it is essential to note that the false positives (FP) are slightly higher than in the penultimate line, with a minimum difference of just 4, making MOTA have a better result than the others. The goal is to minimize FP, FN, and IDS while maximizing the MOTA score. For our tests, we selected video 2 from the test set, consisting of 1120 frames. We used structural similarity as our feature extractor, mainly due to its shorter execution time. The weights of the bipartite graph were generated according to Equation 7, where α and β are parameters that adjust the weights to ensure that they remain in the range between zero and one. In this work, we determined that the appropriate values for α and β are, respectively, 0.4 and 0.6.

4.2.3 General Feature Extractor Results

The experiments on the UA-DETRAC test suite, using the OPEN CLIP, Structural Similarity (SSMI), and SIFT feature extractors, clearly demonstrated the superiority of OPEN CLIP over SSMI and SIFT. This superiority is reflected in significant improvements in metrics, including the reduction of false positives (FP), false negatives (FN), and erroneous identifications (IDS), as well as superior performance in terms of tracking accuracy (MOTP) and overall tracking metrics (MOTA). The results of these experiments are detailed in Figure 10.

This advantage of OPEN CLIP is attributed to its unique ability to analyze the semantic structure of each object, making it notably more robust than other extractors. Specifically, compared to MOTA, OPEN CLIP outperformed SSMI by 11.5% and SIFT by 33.7%, demonstrating its effectiveness in the object tracking task. It is important to note that, concerning MOTP, the OPEN CLIP presents a comparable performance to the other two extractors. This result occurs because MOTP evaluates the spatial difference between the bounding boxes of the actual data and those generated by the proposed method, and the Yolo algorithm plays a fundamental role in standardizing and providing this information, allowing a fair comparison between extractors. It is essential to mention that, in terms of processing rate (Hz), SSMI and SIFT surpassed OPEN CLIP. OPEN CLIP is more computationally intensive due to its detailed semantic analysis. However, considering the importance of tracking accuracy and performance, choosing OPEN CLIP as a feature extractor is justified.

4.2.4 General results of MEDAVET with other methods

Assessment of MEDAVET's performance compared to other models in the literature is vital to determine its effectiveness and relevance. This section provides a brief description of seven models found in the literature to make a comparison with them.

SORT [Bewley *et al.*, 2016] is an algorithm in real-time object tracking. It is based on using particle diffusion, a probabilistic approach, to estimate position and track objects in video sequences. One of the advantages of SORT is its simplicity, which makes it efficient for real-time use. It is beneficial in scenarios where it is necessary to track moving objects on video, such as in surveillance systems, autonomous vehicles, and video analytics. SORT may be less accurate in challenging situations, such as when objects are very close to each other or when occlusions occur.

IOU [Bochinski *et al.*, 2017] is an object tracking algorithm notable for its ability to track high-speed objects with high accuracy, all without the need to process image information. Instead, it relies on a simple motion model, making assumptions about objects' movements. The IOU's success is due to its efficiency and ability to handle high-speed objects. As it does not rely on video image analysis, it is computationally efficient and suitable for scenarios where processing speed is critical. However, it is worth noting that the IOU may be less robust in scenarios where the assumptions of straight-line motion and constant speed do not apply. It may also be less effective in scenarios with frequent occlusions or when objects abruptly change direction.

CMOT [Bae and Yoon, 2018] is an online object tracking algorithm that stands out for its robustness and effectiveness, especially in complex scenes with multiple objects and objects of similar appearance. It uses deep learning techniques to discriminate objects that share close visual characteristics and divides the object tracking problem into smaller subproblems based on the confidence of the tracks; this allows the algorithm to track objects accurately and efficiently, even in complex scenes with many objects.

Model2 [Munjal *et al.*, 2020] is a joint object detection and tracking algorithm in videos using identification features. The algorithm uses a deep learning model to detect and track objects simultaneously. The deep learning model is trained on a video dataset that contains objects labelled with identifying characteristics such as colour, texture, and shape. The algorithm improves the performance of both tasks (detection and tracking) compared to conventional approaches. The algorithm uses identifying characteristics to associate object detections with object tracks accurately. The algorithm is also efficient and robust to occlusion, lighting variations and changes in the appearance of objects. The deep learning model is trained on a challenging video dataset that contains a wide variety of objects and conditions.

JDE [Wang *et al.*, 2020] is a real-time multiple object tracking algorithm that stands out for its ability to perform both object detection and tracking simultaneously. It uses a deep neural network to learn these tasks jointly, resulting in accurate and efficient tracking, even in complex scenarios with multiple objects.

FairMOT [Zhang *et al.*, 2021b] is a multi-object tracking algorithm that treats detection and re-identification in a balanced way. This balanced tracking feature is vital because detection performs better in conventional approaches than re-identification, as it is easier. Task imbalance can lead to lower overall performance of the tracking algorithm. The FairMOT algorithm uses a unique neural network to perform detection and re-identification tasks simultaneously. This technique avoids the task imbalance problem and allows the algorithm to achieve state-of-the-art results on challenging object-tracking datasets. The FairMOT algorithm is also efficient and robust to occlusion, lighting variations, and changes in object appearance. The unique neural network is trained on a diverse and challenging video dataset.

ECCNet [Yu *et al.*, 2022] is an advanced multi-category vehicle real-time tracking and speed estimation algorithm. ECCNet uses an efficient deep neural network composed of three main modules: detection, tracking, and speed estimation. Its chained structure allows efficient reuse of feature map features, reducing computational cost and improving performance in challenging situations such as occlusions and lighting variations. This system offers a robust and effective solution for applications that require accurate real-time vehicle tracking, representing a significant advancement in computer vision.

These solutions are focused on detecting vehicles in frames, that is, on the contours of identified vehicles, to the detriment of vehicle tracking. Such an approach can increase computational resource requirements and decision-making time. However, the present study proposes processing the images sequentially, frame by frame, taking into account the trajectory of the vehicles in each frame. Therefore, it is necessary to verify the efficiency of the proposed solution in light of the work described.

The detailed experimental comparison on the test set of the UA-DETRAC dataset, as summarized in Figure 11, demonstrates that our tracking approach significantly outperforms several existing methods. MEDAVET achieves a MOTA of 58.7%, considerably outperforming the other methods, such as SORT, IOU, CMOT, Model2, JDE, FairMOT, and ECCNet by 42.3%, 39.3%, 46.1%, 3.6%, 34.2%, 27% and 3.2%, respectively. This result is due to CLIP's excellent feature extraction performance and ability to work effectively in ar-

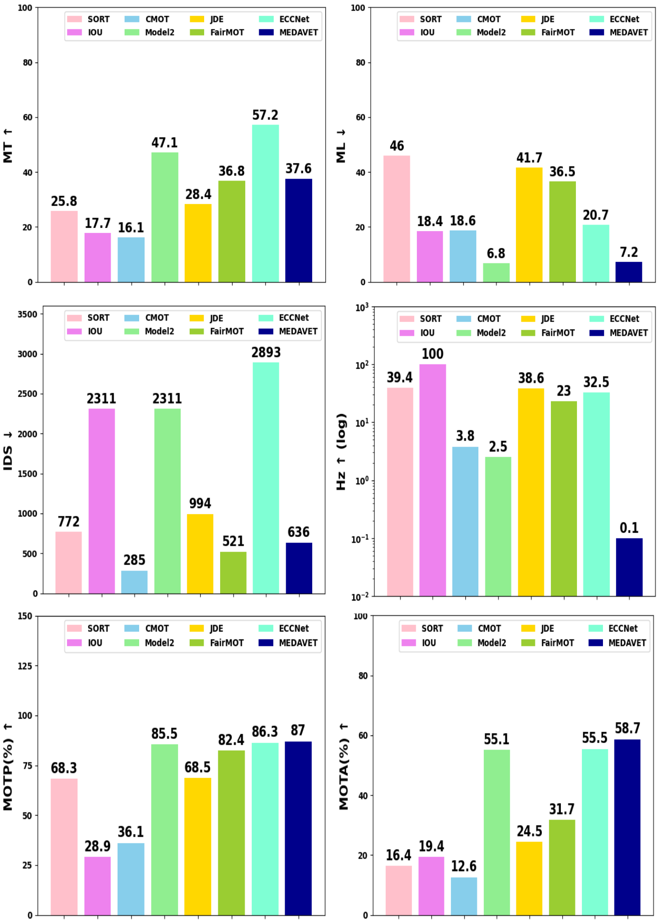

Figure 11. General results of tracking methods.

eas of interest. MEDAVET performs comparably to ECCNet and Model2 concerning MOTP. These three models use advanced detection methods, resulting in detection bounding boxes that are close to real objects. CMOT and FairMOT outperform MEDAVET in terms of ID Switches.

These methods employ re-identification models that can associate object detections based on unique characteristics, differentiating similar objects even in complex scenarios. On the other hand, MEDAVET uses a detection-based association method, which may result in association errors in challenging situations. MEDAVET is only outperformed by Model2 and ECCNet over MT and by Model2 over ML. This result indicates that MEDAVET is highly competitive in keeping track of most objects, but there is still room for improvement regarding lost objects. MEDAVET is surpassed in terms of processing rate by other models due to CLIP's computational complexity and detailed semantic structure analysis. It is essential to highlight that, in applications where precision is fundamental, MEDAVET is justified, even if it is less fast than other models.

4.3 Anomaly Detection

Track4 dataset from NVIDIA AI CITY CHALLENGE 2021 was used to verify the performance of the anomaly detection method [Naphade *et al.*, 2018]. The track4 is divided into a training set and a test set. The training set has 100 videos with a duration of approximately 15 min, and the test set has 150 videos the same length as the training videos. The videos in the dataset have a resolution of 800×410 *pixels*, recorded at a rate of 30 *frames* per second (fps).

4.3.1 Assessment Metrics

$S4$ metric is used to evaluate the anomaly detection performance on the test set. $S4$ combines two metrics: the F1 score and the normalized root mean square error (NRMSE).

$$S4 = F1 \times (1 - \text{NRMSE}) \tag{9}$$

The F1 score is the harmonic mean of recall and precision. Specifically, a true positive (TP) detection is considered the correct anomaly within ten seconds of an actual anomaly (before or after). A false negative (FN) is a real anomaly that our algorithm cannot correctly predict. A false positive (FP) represents the predicted anomaly but is not a real anomaly. The F1 score is summarized by:

$$F1 = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} \tag{10}$$

The normalized root mean square error (NRMSE) denotes the temporal error of the predicted time (by our method) and the ground truth time for all true positive predictions. NRMSE employs a max-min normalization with a maximum value of 300 and a minimum value of 0. In short, NRMSE is defined as follows:

$$\text{NRMSE} = \frac{\min\left(\sqrt{\frac{1}{\text{TP}}\sum_{i=1}^{\text{TP}}\left(t_i^p - t_i^{gt}\right)^2},300\right)}{300}, \tag{11}$$

where t_i^{gt} denotes the start time of the ground truth anomaly and t_i^p is the predicted start time proposed by our method.

4.3.2 Anomaly Detection Results

The algorithm identifies the largest number of anomalies present in the 150 videos in the test set. It provides the start and end time of the anomaly, in addition to the confidence score.

Unidentified anomalies are attributed to challenging video scenarios, such as adverse weather conditions such as fog or nighttime periods where the detector's capabilities are limited. Furthermore, the distance between the camera and the vehicles, together with the small size of the vehicles, contribute to the constant non-detection of these vehicles. Figure 12 presents a visual illustration of these scenarios. In the first three frames of video 45 of the test set, the detector cannot always identify the vehicles, resulting in the loss of information and, consequently, the non-detection of the anomaly. In the following three frames of video 41, also from the test set, the nighttime scenario makes detecting vehicles that are stopped for long periods even more challenging, contributing to the non-detection of anomalies.

Figure 13 presents frames from video 43 of the test set, allowing the visualization and illustration of the anomalies discussed in our study. To distinguish vehicle stopping periods, we adopted a colour signalling system. An alarm signal is displayed in green, indicating that the vehicle has been

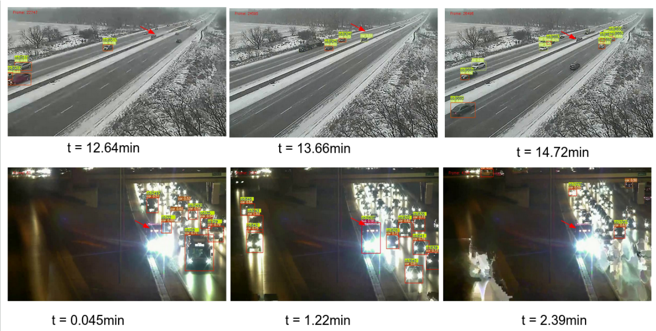

Figure 12. Undetected anomalies.

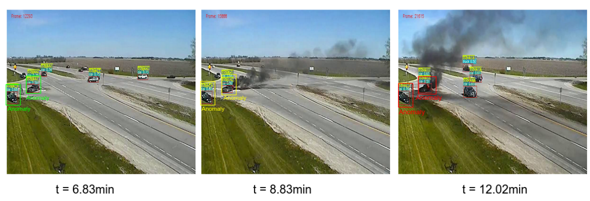

Figure 13. Detected anomalies.

stopped for one minute. After this period, the alarm changes to yellow and remains in that colour for two minutes. Finally, it turns red to indicate that the vehicle has been stopped for more than three minutes. The change of colours over time is intended to inform that the longer the stop, the greater the risk of accidents.

In the context of this work, the analysis of true positives, false negatives, and false positives is very relevant in evaluating system performance. Figure 14 illustrates the results of the entire test suite against these metrics. A true positive in this scenario indicates agreement between actual and predicted data, whether for anomalies or non-anomalies. In this context, we found a total of 198 true positives. On the other hand, false negatives indicate that, although the anomalies are present in the actual data, the predicted data does not correctly identify them. Otherwise, the vehicle that presents anomalies in the actual data does not coincide with the predictions, resulting in 13 false negatives. As far as false positives are concerned, they occur when predictions indicate anomalies that are not present in the actual data, and in this case, we found 12 false positives.

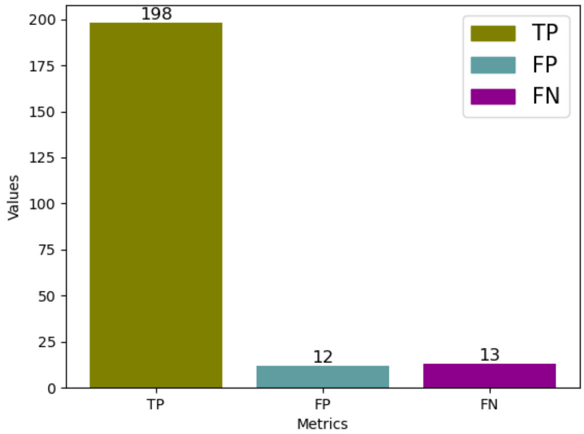

Figure 14. Overall results of TP, FN and FP.

Table 3. Anomaly detection results.

Work	S4	F1	RMSE
MAP	0.7545	0.8371	25.592
MULTI	0.7945	0.8671	25.332
METAVET	0.7845	0.8571	25.432

The presence of false negatives and false positives is undesirable in any scenario. However, if it were necessary to choose between the two, false positives are the least harmful option. This is because a false positive would lead to an anomaly alarm, which may result in a loss of time for accident assistance personnel when travelling to the scene. However, there would be no actual risk situation. However, the worst case scenario would be a false negative, as in this case, there would be a real anomaly, but the algorithm would not detect it, which could lead to severe accidents, including the risk of loss of life if appropriate assistance is not provided.

We evaluated the performance of our method on the NVIDIA AI CITY CHALLENGE 2021 Track4 test set, comparing with the work of Bai *et al*. [2019] here called MAP and the work of Li *et al*. [2020] here called MULTI. As evidenced in Table Table 3, we obtained an overall score of 0.7845 on the $S4$ metric, accompanied by a solid F1 score of 85.71%. The proposed solution performs better than MAP and below Multi than related works due to object tracking. However, the solutions have very similar behaviour.

5 Conclusion

This paper presents an innovative vehicle detection and tracking model, which uses bipartite graphs to model the tracking process. Furthermore, we incorporate the Convex Hull algorithm with the aim of clustering areas where vehicles are moving. To deal with vehicles that remain stationary for long periods and can be temporarily hidden, we employ the QuadTree data structure and a temporal structure, allowing us to group these vehicles and estimate their positions effectively. The results obtained are encouraging, revealing the potential of the proposed approach. Performance is evaluated based on a total score of 0.7845, F1 score of 85.71%, and NRMSE of 25.432. In future work, we plan to further improve our mechanism for classifying objects and detecting anomalies by considering darker scenarios and boisterous images.

Acknowledgements

A. R. H. Reyna would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES), Brazil, for the financial support grant #PROEX-12636374/M and #88887.661865/2022-00. Alex J. F. Farfán acknowledges the financial support grant #PROEX-9527567/D from CAPES, Brazil. R. I. Meneguette thanks the support from CAPES, CNPq and the São Paulo Research Foundation (FAPESP) under grant no. #2020/07162-0 and #2022/00660-0.

Funding

This work was supported by São Paulo Research Foundation (FAPESP) under grant no. #2020/07162-0 and #2022/00660-0, and collaboration of the CAPES.

Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Bae, S.-H. and Yoon, K.-J. (2018). Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):595–610. DOI: 10.1109/TPAMI.2017.2691769.

Bafghi, F. and Shoushtarian, B. (2020). Multiple-vehicle tracking in the highway using appearance model and visual object tracking. In *2020 International Conference on Machine Vision and Image Processing (MVIP)*, pages 1–6. DOI: 10.1109/MVIP49855.2020.9116905.

Bai, S., He, Z., Lei, Y., Wu, W., Zhu, C., Sun, M., and Yan, J. (2019). Traffic anomaly detection via perspective map based on spatial-temporal information matrix. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Available at:`https://openaccess.thecvf.com/content_CVPRW_2019/papers/AI%20City/Bai_Traffic_Anomaly_Detection_via_Perspective_Map_based_on_Spatial-temporal_Information_CVPRW_2019_paper.pdf`.

Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 246309:1–10. DOI: https://doi.org/10.1155/2008/246309.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE. DOI: 10.48550/arXiv.1602.00763.

Bochinski, E., Eiselein, V., and Sikora, T. (2017). High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE. DOI: 10.1109/AVSS.2017.8078516.

Djenouri, Y., Belhadi, A., Chen, H.-C., and Lin, J. C.-W. (2022). Intelligent deep fusion network for urban traffic flow anomaly identification. *Computer Communications*, 189:175–181. DOI: 10.1016/j.comcom.2022.03.021.

Fan, Q., Brown, L., and Smith, J. (2016). A closer look at faster r-cnn for vehicle detection. In *2016 IEEE intelligent vehicles symposium (IV)*, pages 124–129. IEEE. DOI: 10.1109/IVS.2016.7535375.

Ferrante, G. S., Rodrigues, F. M., Andrade, F. R., Goularte, R., and Meneguette, R. I. (2021). Understanding the state of the art in animal detection and classification using computer vision technologies. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3056–3065. Ieee. DOI: 10.1109/BigData52589.2021.9672049.

Ge, D.-y., Yao, X.-f., Xiang, W.-j., and Chen, Y.-p. (2023). Vehicle detection and tracking based on video image processing in intelligent transportation system. *Neural Computing and Applications*, 35(3):2197–2209. DOI: 10.1007/s00521-022-06979-y.

Gomides, T. S., Robson, E., Meneguette, R. I., de Souza, F. S., and Guidoni, D. L. (2022). Predictive congestion control based on collaborative information sharing for vehicular ad hoc networks. *Computer Networks*, 211:108955. DOI: 10.1016/j.comnet.2022.108955.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969. DOI: 10.1109/ICCV.1995.466933.

Hou, X., Wang, Y., and Chau, L.-P. (2019). Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE. DOI: 10.1109/AVSS.2019.8909903.

Huk, K. and Kurowski, M. (2022). Innovations and new possibilities of vehicle tracking in transport and forwarding. *Wireless Networks*, 28(1):481–491. DOI: 10.1007/s11276-021-02623-0.

Li, Y., Wu, J., Bai, X., Yang, X., Tan, X., Li, G., Wen, S., Zhang, H., and Ding, E. (2020). Multi-granularity tracking with modularlized components for unsupervised vehicles anomaly detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2501–2510. DOI: 10.1109/MVIP49855.2020.9116905.

Liu, B., Han, C., Liu, X., and Li, W. (2023). Vehicle artificial intelligence system based on intelligent image analysis and 5g network. *International Journal of Wireless Information Networks*, 30(1):86–102. DOI: 10.1007/s10776-021-00535-6.

Lowe, G. (2004). Sift-the scale invariant feature transform. *Int. J*, 2(91-110):2. Available at: `https://pdfs.semanticscholar.org/19d1c9a4546d840269ef534f6c1c8e3798ce81ac.pdf`.

Meneguette, R. I. and Boukerche, A. (2017). A cooperative and adaptive resource scheduling for vehicular cloud. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 398–403. DOI: 10.1109/ISCC.2017.8024562.

Meneguette, R. I. and Boukerche, A. (2020). Vehicular clouds leveraging mobile urban computing through resource discovery. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2640–2647. DOI: 10.1109/TITS.2019.2939249.

Meneguette, R. I., Madeira, E. R. M., and Bittencourt, L. F. (2012). Multi-network packet scheduling based on vehicular ad hoc network applications. In *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, pages 214–218. Available at: https://ieeexplore.ieee.org/document/6380017/citations#citations.

Montanari, R. (2016). *Detecção e classificação de objetos em imagens para rastreamento de veículos*. PhD thesis, Universidade de São Paulo. Available at: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-08012016-113715/publico/RaphaelMontanari_dissertacao_revisada.pdf.

Munjal, B., Aftab, A. R., Amin, S., Brandlmaier, M. D., Tombari, F., and Galasso, F. (2020). Joint detection and tracking in videos with identification features. *Image and Vision Computing*, 100:103932. DOI: 10.1016/j.imavis.2020.103932.

Naphade, M., Chang, M.-C., Sharma, A., Anastasiu, D. C., Jagarlamudi, V., Chakraborty, P., Huang, T., Wang, S., Liu, M.-Y., Chellappa, R., *et al.* (2018). The 2018 nvidia ai city challenge. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 53–60. DOI: 10.1109/CVPRW.2018.00015.

NATIONS., U. (2020). Seventy-fourth session of the united nations general assembly: Improving global road safety. *United Nations General Assembly*. Available at: https://digitallibrary.un.org/record/3879711?v=pdf.

of Health, S. D. (2018). Global status report on road safety 2018. *WHO*. Available at: https://www.who.int/publications/i/item/9789241565684.

of Health, S. D. (2023). Global status report on road safety 2018. *WHO*. Available at: https://www.who.int/publications/i/item/9789241565684.

P, G. L., P, A., Vinayan, G., G, G., M, P., and H, A. S. (2023). Lucas kanade based optical flow for vehicle motion tracking and velocity estimation. In *2023 International Conference on Control, Communication and Computing (ICCC)*, pages 1–6. DOI: 10.1109/ICCC57789.2023.10165227.

Pawar, K. and Attar, V. (2021). Deep learning based detection and localization of road accidents from traffic surveillance videos. *ICT Express*. DOI: 10.1016/j.icte.2021.11.004.

Pereira, R. S., Lieira, D. D., Silva, M. A. C. d., Pimenta, A. H. M., da Costa, J. B. D., RosÃ¡rio, D., Villas, L., and Meneguette, R. I. (2020). Reliable: Resource allocation mechanism for 5g network using mobile edge computing. *Sensors*, 20(19). DOI: 10.3390/s20195449.

Reynolds, D. A. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663). Available at: http://leap.ee.iisc.ac.in/sriram/teaching/MLSP_16/refs/GMM_Tutorial_Reynolds.pdf.

Santos, E. d. (2014). O uso de visão computacional para o controle de um manipulador robótico. B.S. thesis, Universidade Tecnológica Federal do Paraná. Available at: http://repositorio.utfpr.edu.br/jspui/handle/1/14630.

Shi, J. *et al.* (1994). Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE. DOI: 10.1109/CVPR.1994.323794.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475. DOI: 10.48550/arXiv.2207.02696.

Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer. DOI: 10.48550/arXiv.1909.12605.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., Lim, J., Yang, M., and Lyu, S. (2015). DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136. DOI: 10.48550/arXiv.1511.04136.

Yu, C., Yang, J., Jiang, S., Zhang, Y., Li, H., and Du, L. (2022). Eccnet: Efficient chained centre network for real-time multi-category vehicle tracking and vehicle speed estimation. *IET Intelligent Transport Systems*, 16(11):1489–1503. DOI: 10.1049/itr2.12227.

Zhang, X., Zheng, Y., Zhao, Z., Liu, Y., Blumenstein, M., and Li, J. (2021a). Deep learning detection of anomalous patterns from bus trajectories for traffic insight analysis. *Knowledge-Based Systems*, 217:106833. DOI: 10.1016/j.knosys.2021.106833.

Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021b). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vision*, 129(11):3069–3087. DOI: 10.1007/s11263-021-01513-4.

Zhao, Y. (2021). *Good Practices and A Strong Baseline for Traffic Anomaly Detection*. DOI: 10.48550/arXiv.2105.03827.