
An Interpretable and Explainable AI Framework for Urban-Suburban Traffic Analysis and Understanding

[Stefano Ferilli](#)*, [Davide Di Pierro](#), [Domenico Redavid](#), [Eleonora Bernasconi](#)

Posted Date: 1 August 2023

doi: 10.20944/preprints202308.0062.v1

Keywords: Road Traffic Understanding; Artificial Intelligence, Clustering, Process Mining, Automated Reasoning, Computer vision



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

An Interpretable and Explainable AI Framework for Urban-Suburban Traffic Analysis and Understanding

Stefano Ferilli ^{1,*},[†] , Davide Di Piero ^{1,†} , Domenico Redavid ^{2,†} 
and Eleonora Bernasconi ^{1,†} 

¹ Department of Computer Science, University of Bari Aldo Moro, 70125 Bari, Italy

² Department of Economic and Finance, University of Bari Aldo Moro, 70124 Bari, Italy

* Correspondence: stefano.ferilli@uniba.it

† These authors contributed equally to this work.

Abstract: Studying and understanding the behavior of people and vehicles on public roads can be of utmost importance for supporting the activities of many institutional stakeholders. It may allow automated supervision of the ongoing situation in a given place, with warnings or alarms raised in case of anomalies. It may be used to plan their interventions on road and town organization. It may provide them with advanced support to decision-making. The number of involved entities and places to manage makes it infeasible to manually handle all the traffic-related tasks. Moreover, the complexity of the tasks to be carried out requires the adoption of advanced approaches. Many AI solutions are nowadays mature to support these requirements. In some cases, the motivations and objectives of traffic management require the AI outcomes to be understandable, interpretable and explainable. In this paper, we propose TrAnSIT (TRaffic ANalysis Supervision and Interpretation Tool), an AI-based framework that combines several modules, each aimed at tackling a specific traffic-related task, so as to cover a wide landscape of traffic-related issues, from overall urban or suburban traffic management to surveying specific road segments that fall under the scope of one camera. Most of these modules are based on AI techniques that support a human-level understanding of the outcomes.

Keywords: road traffic understanding; artificial intelligence; clustering; process mining; automated reasoning; computer vision

1. Introduction

Traffic monitoring and understanding is becoming extremely relevant for everyday life in modern societies. Despite technological and methodological efforts in the field, road crashes represent the 8th leading cause of death worldwide [1]. Traffic management is an urgent and essential task to assist in dealing with massive traffic crowds [2]. While the traffic load has increased considerably, traffic infrastructures have not kept pace with this evolution. Economic calculations show that effectively managing traffic on existing roads is more profitable than building new ones [3]. Many organizations and agencies are putting effort into this task for solving different kinds of problems. E.g., the Italian Police showed much interest [4] in understanding and preventing criminal activities (related to security) as well as traffic congestion and accidents (related to society) that happen on the roads. It was considered also in connection with the increasing attention towards social and community issues, as well as towards pollution and the green economy [5]. Both academia and industrial companies are putting much effort into the tasks of prediction, analysis, and understanding of traffic data, possibly to make new sustainable ideas and initiatives emerge. This vision represents a long-term ambition but many circumstances require immediate solutions.

Hence, the need for automatic tools to grasp traffic peculiarities. Depending on the specific analysis, traffic can be inspected at different level of granularity: local roads, highways, specific cities or even entire countries. In all cases, traffic understanding requires cutting-edge algorithms

to deal with large, complex and heterogeneous data. The relevant tasks involve the main Big Data dimensions [6] where Data Mining KDD processes can be seen as an important aspect [7]:

- **Volume:** the amount of traffic data is enormous due to the number of vehicles crossing roads all day, which is the main reason the community is interested in solving this problem.
- **Variety:** available data range from videos to database records or textual logs, often depending on the specific task; also the available technologies (e.g., sensors and cameras) do not provide the same structure.
- **Velocity:** it goes without saying that traffic flow data is collected at a rapid pace.

Furthermore, in relation to our domain we can identify the following characteristics:

- **Vinculation:** Streams of traffic data are strongly interrelated: any traffic lights, pedestrians, or accidents that come into play affect the whole scenario.
- **Validity:** it is assured by the technologies used. More specifically, cameras provide instantaneous feedback on what happens; for sensors, specific tests assure the good quality of the detected facts.

The main objective of this work is to define a complete and comprehensible framework for road traffic analysis. It would provide advanced support for the activities of relevant stakeholders (typically, institutional stakeholders involved in traffic planning, management, safety and security). E.g., early detection, or even prediction, of road traffic anomalies or road-related crimes may allow the enactment of proactive measures to prevent them or at least mitigate their effects. Building models of normal or abnormal traffic behaviour (s) may allow the development of strategies and plans in road development or organization, that ensure smoother fruition by its end-users and optimize the tasks of other stakeholders (e.g., the opportunities and revenues of commercial activities, the reduction of pollution and of traffic jams, etc.).

Road traffic can take place in very different environments, each characterized by peculiar features, posing its own issues and requiring tailored solutions. E.g., suburban traffic, such as highway one, is characterized by a 'linear' sequence of places that can be traversed, with few possible variations. On the other hand, urban traffic leaves road users much more freedom in building their paths along a network of interconnected places. As a consequence, it can be easier to define the relevant places to be used as checkpoints in a suburban road, but it may be much more complex within a town. These checkpoints may be relevant just for detecting the specific vehicles or pedestrians that pass through them, or a deeper understanding of what happens in a limited section of the roads might be required.

This paper proposes TrAnSIT (TRAffic ANalysis Supervision and Interpretation Tool), an integrated AI-based framework for traffic analysis and understanding. It encompasses the following approaches, involving suburban or urban traffic environments.

- At the urban or suburban level, several cameras spread along the main roads are already present, which can detect the passage of vehicles from pre-defined points of interest, also recognizing the specific vehicle identity based on their plate numbers. Additional data can be obtained from location sensors (e.g., based on GPS) carried by vehicles (especially public transportation ones) and pedestrians, that continuously track their position. The vehicle passage events generated by this network of sensors can be used to automatically create models of traffic behaviour, to be used for supervision and/or prediction. These models may concern normal behaviour or abnormal behaviours of interest. In the former case, the aim might be checking that everything is compliant with the normal model, and raising alarms in case of deviations. In the latter case, the aim might be recognizing anomalies and notifying the target users (e.g., a vehicle on highways carrying out burglary or other kinds of illegal activities).
- As said, inside towns, the network of streets allows for an extremely large number of path variations. So, a relevant issue here may be identifying automatically the most relevant locations in the town to be supervised, where a more fine-grained analysis of traffic behaviour can be carried out.

- Given an (urban or suburban) place of interest, cameras can be placed not only to detect vehicle passage but also to monitor the behaviour of people and vehicles. Here, both the above-mentioned tasks can be needed:
 - The most relevant zones (places of interest) in the camera framing can be manually defined or automatically detected, in order to specifically check the vehicles or pedestrians traversing them.
 - Once more, models of normal or abnormal behaviour(s) in traversing these zones can be learned and used for supervision or prediction purposes.

Additionally, the same data coming from the camera can also be fed into an automated reasoning system that reasons on the detected events as long as they take place, interprets them according to the perspectives of interest to the system's users, and raises warnings or notifications for noteworthy situations, also being able to explain why the situation was recognized and suggesting possible actions to handle them.

These approaches can be combined, in order to have a more comprehensive understanding and management of traffic. In such a way, it can ideally provide seamless, continuous and integrated traffic monitoring along both sub-urban and urban roads.

To be useful and trustworthy, such an integrated system should be transparent, so as to allow its users to 'look into' the AI machinery and behaviour, check and validate its outcomes, and make consequently thoughtful decisions. We distinguish 3 levels of transparency for AI systems: understandability, interpretability, and explainability, based on the following definitions.

understandability the outcomes of the AI system are expressed in human-level terms;

interpretability the output of the AI system can be easily traced back to the input;

explainability the full rationale connecting the output of the AI system to the input can be explicitly reported.

Given the complexity of the tasks, interactive and visual tools can also be helpful for supporting the decision makers process.

Our framework aims at ensuring all the above levels and features wherever possible, so as to act as a road traffic expert and an advisor for its users and stakeholders. Its main advantages are that it covers all kinds of traffic settings, providing human-like insights at different levels of granularity, and mixing sub-symbolic and symbolic AI algorithms.

The rest of this paper is organized as follows. In the next section, we recall and summarize related works. Then, in Section 3 we discuss in some detail the AI techniques we selected to support our framework and how they were used in the framework, while Section 4 describes the application of our framework to a sample scenario, before concluding the paper in Section 5.

2. Related Works

In the urban traffic setting, anomalies may vary according to the specific scenario: an anomaly may refer to a specific vehicle or a specific dangerous situation (e.g., accidents, fires, congestion).

A common need for traffic management is the prevention, or at least forecasting, of accidents. Many factors come into play when dealing with traffic accidents like age [8] or environmental factors [9]; however, most works do not take these factors into account. The possibility of taking action against road accidents can make a sort of big impact on security, and healthy but also from an economic point of view. It is estimated that in 2014 Australia spent \$27 billion for traffic analysis [10].

Many solutions provide non-explainable approaches like Sofuoglu et al. [11] that developed a tensor-based detector for anomaly detection, while Kong et al. [12] proposed LSTM [13] with autoencoders, suitable also for suburban scenarios. Both works were focused on urban scenarios with spatiotemporal data. On the other hand, Di Mauro et al. [14] used LSTM networks in highway traffic

analysis to detect anomalies in vehicle trajectories. In this work, data was provided as logs by the Italian Police.

Traffic understanding resides in the Big Data domain; hence, the format of data available varies. As an example, in many scenarios, the detection of congested areas or accidents in a city requires the use of GPS. D'Andrea et al. [15] proposed a segment traffic classification to distinguish dangerous (or suspect) areas. More recently, solutions based on satellite video are gaining momentum. Ahmadi et al. [16] proposed strategies to recognise a vehicle through Computer Vision (CV) techniques and compute metrics like average speed and trajectories by comparing subsequent frames.

Following this trend, many applications in the field of traffic understanding have a visual approach, using a camera and recording and/or shooting traffic roads. In some applications, the visual approach is fundamental because of the calibration of instruments based on vehicle and lane dimensions like in [17]. Traffic understanding may concern the forecasting of traffic in specific areas in some periods. For this reason, they are mainly modelled with time series approaches [18,19]. In some approaches [20,21] another type of traffic information is analysed, that is the mobile traffic. There is a strict connection between the two as shown in [22,23].

Most modern AI approaches are pushing towards introducing explainable models in non-explainable contexts [24–26] but it is still an open issue and far from being solved. Introducing explanations is essential when decisions have a strong social impact. The decision about the kind of approach enormously affects the instruments to be employed, performance, kinds of explanations, and costs. Current data analytics approaches revolve around the idea of using data to characterize and predict traffic risk in order to prescribe better (safer) routes, driver assignments, rest breaks, etc. With the advances in information technology, it is possible to collect ever-increasing amounts of relevant data, such as comprehensive incident databases, real-time driving data feeds, or relevant factor characteristics [27].

Among the non-explainable solutions, traffic understanding in videos plays a key role. The technological support is given by cameras. Computer Vision algorithms allow us to detect relevant road elements and visualize the objects in the scene. Visual multiple object tracking (VMOT) aims at locating multiple targets of interest, inferring their trajectories and maintaining their identities in a video sequence [28]. Nowadays, many recorded videos or images are available thanks to visual surveillance and displays on autonomous vehicles. Santhosh et al. [29] provided a survey about anomaly detection techniques capturing photos from video surveillance systems. One of the main points of the detection is represented by the tracking system.

Further applications regard motion planning in autonomous vehicles. Motion planning for autonomous vehicles in road traffic is essential for occupant safety and comfort. Karle et al. [30] group the motion prediction approaches into three categories:

- physics-based prediction: studying kinematics models to estimate the quality of the possible state transitions.
- pattern-based prediction: a cluster of trajectory data to determine trajectories or manoeuvres.
- planning-based prediction: learning optimal behavior by cost function estimation (Inverse Reinforcement Learning).

Apart from recordings, one may find himself surprised by the variety of simulated data which can be considered extremely helpful. This scenario includes computer games, urban visualisation, and urban planning simulation in autonomous driving settings.

Generally, most of these works lack the capability of explaining the obtained results, affecting how the end user can trust and rely on automatic object detection systems. In the context of autonomous vehicles, Umbrello et al. [31] try to mitigate the issue thanks to the Value Sensitive Design (VSD) [32] in combination with the Belief-Desire-Intention (BDI) [33] model. The combination leads to making the vehicle behave following some design principles which, in turn, follow our ethical schemes and values on the road. Other approaches rely on very cutting-edge technologies like transformer-based architecture [34]. Dong et al. [35] combined visual and textual information to build a network capable

of giving predefined explanations when making decisions. Finally, it should be assessed when explanations are really desired or needed. Shen et al. [36] investigate this need in order to better understand users' expectations and increase their reliability in non-explainable complex systems.

Given this need, many works rely on symbolic solutions to provide human-like explanations of the involved process. An interesting solution, albeit to network traffic, combined a visual and a Knowledge Representation and Reasoning (KRR) [37] approach, continuously updating the KRR module with patterns extracted by visual features [38].

The KRR setting often requires a technological change in the process of data acquisition. Symbolic data is structured and, as such, is not prone to be extracted by images or videos. Conversely, sensors are able to produce, manipulate, and transfer information in a structured way, similar to what KRR frameworks privilege. As part of a bigger project (the TRYS system [39]), Cuenca et al. [40] developed a framework for traffic management based on classical logic and the Prolog programming language, where data is gathered by sensors and rules are supported by common knowledge. It is a knowledge representation environment supporting models to perform traffic management. The aim is to improve traffic or reduce the severity of existing problems. For instance, they can recommend increasing the duration of a traffic light phase (e.g., green), or suggest displaying certain messages on some Variable Message Panels to divert traffic.

Hülßen et al. [41] designed an ontology-based solution for managing traffic situations by means of a query in a knowledge base. Since many situations are frequent, we are likely able to categorise them. Choi et al. [42] developed a spatiotemporal ontology selecting verbs describing recorded movements in videos.

In the context of visual surveillance, the description of object behaviour needs real-time analysis. Toal et al. [43] proposed a solution to be applied when most objects and behaviours are known. The authors merged a visual-based perspective with a KRR one thanks to the given knowledge about specific patterns of the expected trajectories.

As previously mentioned, some research trends prefer KRR solutions thanks to their explanations. Logi et al. [44] introduced an assessment of the KB system for real-time scenarios. The solution aims to discover traffic management solutions, as well as justify them in a certain way and be trustworthy. KRR solutions are extremely relevant in the context of multi-agent systems. Modelling route choices is a primary task in urban modelling. Manley et al. [45] developed a spatial data-based approach to simulate traffic schemes.

One of the first approaches to traffic monitoring in urban scenarios applying rule-based reasoning on visual data was proposed in [46], based on two separate modules, an image processing one for extracting visual data, and a high-level one for knowledge-based tracking of vehicles. The advantages of combining the two perspectives are:

- symbolic rules, differently from visual operators, better describe the semantics of traffic scenes (a vehicle being a physical entity that must exist from its entrance in the scene until its exit; it cannot overlap with other vehicles, split or change shape or size, and so on);
- symbolic rules have a clear and simple syntax and are a natural interface with human experts, helping with decision-making processes;
- the reasoning module is independent of the specific facts and rules added; hence, the base can be incremented by simply adding new rules;
- the high-level module is able to correct errors of the low-level modules by exploiting temporal and scene consistency along the image sequence.

We strongly believe in the effectiveness of this solution, and propose a similar solution, but based on a more comprehensive reasoning engine.

In spite of their relevance and novelty, none of these works aimed at creating a framework that covers the whole landscape of traffic issues, from suburban to urban settings, from location-based to video-based processing.

3. AI-based Framework Description

This section describes the AI techniques and approaches we included in our traffic analysis framework, in order to support the various tasks involved.

3.1. Computer Vision for Information Extraction from Traffic Videos

Given videos obtained through cameras placed along urban and suburban roads, we exploit CV [47] to process the video streams and obtain relevant information. Through a multiclass classification task, we extract relevant objects from the videos, leveraging object detection—a widely used application in CV that has already been applied to problems such as security [48,49] and car tracking [50].

3.1.1. Vehicle Detection & Tracking

For object detection, we employ one of the most popular neural network-based models, namely You Only Look Once (YOLO). YOLO stands out for its compact size, fast computation, and direct output of bounding box positions and object categories through the neural network. Unlike other approaches, YOLO considers the entire image for detection, encoding global information and reducing the background detection errors [51]. It was successfully used in traffic-related applications [52,53]. Additionally, we incorporate object tracking algorithms, specifically *ByteTrack* [54] and *BoT-SORT* [55], to track objects within the videos. Object tracking is valuable as it associates a unique ID with each detected object, enabling its continuous monitoring and analysis throughout the video sequence.

Our framework includes an application dedicated to vehicle detection [56] in videos and processing the data to be sent to reasoning and process mining algorithms. This application features an intuitive interface that allows users interested in traffic analysis and study to easily and quickly utilize it.

Specifically, the functions comprising this application are as follows:

- *Load Model*: This function handles loading an object detection model. In our example, we used the YOLOv8 model based on the YOLO algorithm. This model was trained on the COCO128 dataset to identify objects in videos.
- *Display Tracker Options*: This function enables users to select options for object-tracking visualization. Users can choose whether to display tracking and select the tracking algorithm to use.
- *Log Detection*: This function records object detection information in a log. It takes input such as the bounding box coordinates of the object, object ID, object type, detection confidence, and a dedicated object for log writing.
- *Display Detected Frames*: This function displays video frames with the detected objects (see Figure 3). It accepts input such as the detection confidence level, object detection model, frame to display, object for log writing, frame image, and other options for object tracking. The following operations are performed within this function:
 - *Drawing RoI*: made to allow users to directly highlight points of interest on the frame (if they want). It visualizes object tracking if specified by the user, otherwise predicting objects using the YOLOv8 model. Displaying detection results on the frame. Recording detections in the log using the "Log Detection" function.
 - *Play Stored Video*: This function manages video playback and analysis of detected objects. It takes inputs such as the detection confidence level, object detection model, start and stop buttons, and the video to be analysed.

Finally, a log of detections is displayed in a tabular format. This log provides detailed information about the detected objects in the video, including the bounding box coordinates, object ID, vehicle type, detection timestamp, and associated confidence level. To evaluate the frequency of recognition

point collection, we adopted an experimental approach using a dataset of videos representative of our use cases. The videos were played back at a rate of 30 frames per second (FPS), in line with common industry standards. For the processing of each frame, we employed an object recognition algorithm that takes approximately 0.1 seconds. These data will be used to identify the Regions of Interests (RoI) as explained in the next subsection.

3.1.2. Trajectory analysis and event detection

Once the RoI is identified, the trajectories of vehicles within that region can be analyzed to detect events or significant changes. We monitored changes in vehicle speed, which can help identify potential hazards such as sudden braking or rapid acceleration, and vehicle directions, which can reveal abnormal behaviors such as U-turns or sudden lane changes. These analyses can serve various purposes, including road safety, traffic control, urban planning, or infrastructure management. The *analyze vehicle trajectories* function performs trajectory analysis on each vehicle by using the logs generated by the computer vision step. It calculates the relative velocity for each log entry by computing the Euclidean distance between the coordinates at times t and $t - 1$ and dividing it by the corresponding time difference. Subsequently, average relative velocities associated with each vehicle are computed. To classify speed and direction events a movement threshold is calculated using a specified percentage value based on the average velocities of vehicles passing through the regions of interest. Speeds are then classified based on the computed thresholds. Direction classification of vehicle trajectories is also performed (see Figure 1).

vehicle_ID	vehicle_Type	timestamp	confidence	bbox_x1	bbox_y1	velocity	average_velocity	Velocity Event	Direction Event
1.0	2.0	2023-06-15 16:39:22	0.7312	219.3291	341.7372	None	27.2005	Stationary vehicle	
1.0	2.0	2023-06-15 16:39:23	0.8093	220.9778	345.5794	15.7647	27.2005	Moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:23	0.7388	223.7203	348.6507	23.3171	27.2005	Moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:23	0.6511	226.8869	352.6377	27.9546	27.2005	Fast-moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:23	0.7255	228.5649	355.7459	20.0576	27.2005	Moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:23	0.681	232.1532	360.7019	34.1191	27.2005	Fast-moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:24	0.7542	234.5149	363.948	22.9172	27.2005	Moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:24	0.6296	239.1604	368.3438	34.7292	27.2005	Fast-moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:24	0.579	241.1629	371.6958	20.6917	27.2005	Moving vehicle	\ Southeast
1.0	2.0	2023-06-15 16:39:24	0.6496	244.5113	374.9624	23.4194	27.2005	Moving vehicle	\ Southeast

Figure 1. Event log.

The information extracted from the video concerning velocity, detection type, process and activities identifier, incremental sequence of activities associated with a vehicle, and vehicle ID is crucial for generating events that will be used as inputs to the traffic modeling and interpretation modules described later in this section.

3.2. Clustering for the Identification of Noteworthy Areas

Given a map or a video framing, areas (Regions of Interest, or RoIs) that are relevant for traffic interpretation and understanding can be manually identified by the framework users to let the system know that it is relevant to detect vehicle interactions with those areas (typical cases may be a vehicle entering or exiting the area). As an alternative, or in addition to, manual definition of the areas, our framework allows to automatically identify the RoIs from a set of vehicle-related positions, as detected by sensors (e.g., GPS) in an open landscape or by video processing from specific locations. We carry out this task using clustering.

Clustering [57] is the process of grouping items in such a way items belonging to the same group (cluster) are closer (based on a distance metric) than items belonging to different groups. In our framework, we use it to identify traffic-dense areas and otherwise relevant road zones, based on positioning information (e.g., provided by GPS tools or other sensors) or on object identification in camera frames. More specifically, we produce for each cluster a RoI that encloses the corresponding points, to be later used in the analysis of vehicle trajectories and related events. We selected 5 different clustering techniques to try out in our approach:

DBSCAN [58] is a density-based algorithm that groups data points into dense regions, considering noise points as well.

K-means [59] is a clustering algorithm that assigns data points to clusters based on their similarity.

Agglomerative Clustering [60] is a hierarchical clustering algorithm that progressively combines data points into larger clusters.

Gaussian Mixture Models (GMM) [61] is a statistical model that assumes data are generated from a set of Gaussian distributions.

Spectral Clustering [62] is a graph-based clustering algorithm that utilizes the spectral representation of data.

A crucial choice in clustering, and an input to most techniques, is the number of clusters to generate. To automatically assess the optimal number of clusters, we try 4 different techniques:

The Elbow Method [63] is based on inertia, which represents the sum of squared distances between data points and the nearest cluster centroid. This method involves plotting a graph of inertia against the number of clusters and identifying the point where inertia stops decreasing rapidly and becomes more gradual. This point, often referred to as the *elbow* in the graph, is considered the optimal number of clusters.

The Silhouette Method [64] evaluates the cohesion within clusters and the separation between clusters. For each data point, the silhouette score is calculated, measuring how similar the object is to its own cluster compared to other clusters. The Silhouette Method entails plotting a graph of the silhouette score against the number of clusters and identifying the point where the score reaches its maximum. This point represents the optimal number of clusters.

The External Validity Index [65] compares clustering results with known class labels. Measures of completeness, homogeneity, and V-measure are calculated to assess how well the clusters correspond to the known class labels. The optimal number of clusters can be determined by considering the maximum value of these measures.

The Cohesion-Separation Index [66] evaluates the cohesion within clusters and the separation between clusters based on distances between data points. Cohesion is calculated as the sum of distances between points within the same cluster, while separation is calculated as the sum of distances between points in different clusters. The optimal number of clusters can be determined by considering the maximum value of the cohesion-separation index.

Given a clustering, the RoIs are determined by applying the convex hull [67] approach to the points in the most relevant clusters. The convex hull is the tightest geometric shape that encloses all the points in a cluster, providing a boundary for the RoI. This approach ensures that the RoI encompasses the entire cluster and excludes any outliers or noise points. By incorporating the convex hull into the analysis, we effectively define the spatial extent of the RoI and focus the subsequent trajectory analysis and event detection within this defined area.

3.3. Process Mining for Automatic (Urban or Suburban) Behavioral Modeling, Supervision, Prediction & Classification

A *process* is described as the set of *actions* performed by agents (also humans). The *workflow* is the formal description of the sequence of actions leading to a correct process. The execution of actions can be sequential, parallel, conditional, or iterative. The execution consists of a set of events and actions

among them. A *case* is an execution of actions compliant with a workflow. Case traces are described as a list of events associated with timestamps. A *task* is a subset of the whole work, that is likely to be executed for many cases of the same type. An activity is the execution of a task by an agent. Our framework relies on Process Mining and Management techniques to learn models of traffic behavior, based on the following associations [68]:

- A process corresponds to the behavior of vehicles along the road. Single-vehicle or overall traffic behaviors can be modeled.
- the agents are the vehicles.
- A case is a route on the road by one or more vehicles, from their first detection in to its last detection. More specifically, the first detection of the vehicle starts a case, that ends when the vehicle is not detected anymore.
- Activities are associated to the vehicle passing from the pre-defined places, e.g., under selected gates along the road or inside predefined RoIs; the driving behavior of the vehicle (in terms of direction and speed) may also be associated to activities.

The WoMan framework [69–71] is a logic-based framework for Process Mining and Management. It is based on First-Order Logic (FOL) descriptions, ensuring understandable models and behavior. WoMan provides features that are useful for traffic understanding purposes: complex processes management, noise handling, time management, contextual information management, efficiency, incrementality (not available in most other systems).

Since traffic behavior is extremely variable, incrementality is particularly relevant, because it can update the model without retraining from scratch.

The input to any process mining system is a sequence of events describing the execution(s) of a process. In WoMan each event is described in terms of:

- the event timestamp,
- the type of the event: begin or end of a process, begin or end of an activity (allowing to consider time span and concurrency), or context description,
- the name of the workflow the process refers to,
- a unique identifier for each process execution,
- the name of the activity,
- the progressive number of occurrences of that activity in that process,
- (optionally) the agent that carries out the activity.

Thanks to the fully incremental setting, each event can be provided to the system independently of the others (so far as the progression of timestamps is maintained), possibly even interleaving events for different cases; the system will take care of processing them as appropriate.

Some of the functions provided by WoMan are:

Model Learning WoMan's learning module is called WIND (Workflow INDucer) which learns models for a case, after the acquisition of events. It usually does not need a large dataset to obtain a stable model.

Conformance Check WoMan's supervision module WEST (Workflow Enactment Supervisor and Trainer) can check the compliance of each event in new process executions with respect to a given model. For each event, the system returns a compliance outcome among the following options:

- ok** the event is compliant with the process model;
- warning** the event is not compliant with the process model, for one of several reasons; each warning is associated with a degree of severity. Of course, warnings that encompass others have a greater severity degree than the others.
- error** the event cannot be processed, being syntactically or semantically wrong.

After all the events of a case have been processed, the case can be used to refine the model.

Activity Prediction During process execution, WoMan can be asked to predict the next activities that will be carried out using the SNAP (Suggester of Next Action in Process) module. The candidate predictions are ranked based on a combination of several parameters.

Process Prediction Given a case of an unknown workflow, the system may be asked for a classification of the process that is being executed, among a set of candidate processes whose models are available to the system. WoGue (Workflow Guesser) is the model in charge of making this prediction, based on a comparison of the events of the current process enactment to the candidate models.

Specifically, WoMan's process discovery module WIND is in charge of learning the models, while module WEST is used to check compliance of new cases to the currently available models. They may interact, calling WIND to incrementally refine the current model based on the information of a new (compliant or not) case processed by WEST. During a case execution, modules SNAP and WoGue can be used to obtain predictions on it: the former can predict the next activity that the vehicle will carry out, while the latter can classify the current partial execution based on different behavioral models (e.g., normal behavior, dangerous behavior, criminal behavior, etc.).

3.4. Automated MultiStrategy Reasoning for Traffic Interpretation

In addition to learning models of standard or abnormal traffic behaviors, and using these models to supervise future traffic situations, our framework also provides for traffic interpretation. This feature is designed in the form of an expert/decision support module, that uses automated reasoning to reproduce the inferences that an expert continuously looking at the traffic videos would carry out. In this way, dangerous or otherwise relevant situations can be detected in real time and notified to interested stakeholders, possibly also suggesting associated actions that they might accept, modify or reject. For this purpose, relevant domain knowledge must be expressed in a logical formalism and stored in a so-called Knowledge Base (KB). It may be provided by domain experts and formalised by knowledge engineers, or automatically learned using FOL approaches. Using the KB a traffic-related video can be continuously 'observed', and the information it provides can be formalized and provided to the automatic system, that interprets it raising warnings in case of relevant situations. Also, the system can answer specific queries by the stakeholders.

In particular, our framework uses GEAR (an acronym for 'General Engine for Automated Reasoning') [72], a logical inference engine capable of MultiStrategy Reasoning, i.e., of integrating and bringing to cooperation several inference strategies in order to cope with the several complexities posed by real-world tasks and problems. The current prototype includes the following strategies:

Deduction aims at making explicit knowledge that is implicit in the available knowledge but is a necessary consequence thereof.

Abstraction reduces the amount of information conveyed by a set of facts. This reduces the computational load needed to process the set of facts, provided that the information that is relevant to the achievement of a goal is preserved.

Abduction is devoted to coping with missing information, by guessing unknown facts that are not stated in the available knowledge but are needed to solve a given problem, provided that they satisfy some integrity constraints. Of course, there may be many plausible explanations for a given observation.

Uncertainty The possibility of handling uncertainty may dramatically improve flexibility and robustness of reasoning.

Argumentation deals with inconsistent knowledge, to distinguish which of several contrasting, but internally consistent, positions are justified, based on the relationships among the involved knowledge items and on their properties.

Induction is the inference of general knowledge starting from specific observations.

Ontological An ontology defines and describes the kinds of entities that are of interest in a domain, their properties and relationships. Typical ontology-based reasoning tasks are inheritance and consistency checking.

Similarity-based computation between FOL descriptions is complex due to non-unique mapping between the descriptions.

Analogy Analogy is the cognitive process of matching the characterizing features of two subjects, objects, situations, etc. After finding an analogy on some roles, the association can be extended to further features.

Reasoning operators act on the content of a so-called *Knowledge Base* (KB). Knowledge bases handled by GEAR may include various kinds of knowledge items, including Facts (simple statements), Rules (implication-like formulas), Integrity Constraints, Abstraction Operators, and Argumentative relationships. Rules may have a priority (a number used to determine which rule should be executed first in case of conflicts). The premises of rules can use any, possibly nested, composition of conjunction, disjunction and negation. Its conclusion can be a single atom or a conjunction of atoms or negated atoms. Additional components are available to express abducibles and integrity constraints for abduction, abstraction operators or argumentative relationships. Knowledge can be organized into modules. Other predicates can be used to specify system settings (e.g., to set flags that direct the system's behaviour), information related to user interaction (e.g., to specify the information that can be asked to the user if missing in the KB), calls to pre-defined procedures (e.g., to call Prolog to carry out some computations), etc.

3.5. Visual Analytics for Traffic Understanding

We also included in our framework a tool for visual analytics, described in [73], aimed at supporting traffic understanding by humans. While not based on explainable AI approaches, it is a nice and useful complement to the other techniques. For instance, it can be used to spot outliers by looking at some charts which show (agglomerate) features of the problem. The greater the amount of available data, the more relevant the visual analytics. In our scenario, visual analytics may also be used to confirm some anomalous behaviour patterns that were alleged to be true but not proven yet.

4. Experiments on a Sample Scenario

We tested and evaluated the proposed framework on various use cases taken from a sample hypothetical scenario involving *urban* and *suburban* traffic areas. Our scenario starts from an Italian highway, where we want to model standard traffic behaviour along the whole highway based on positioning-detecting sensors. Then, we use information coming from cameras located on critical places along the highway to model vehicle behaviour in these critical points and interpret this behaviour using an automated reasoning system. The Italian National Police identified a number of such locations worth further analysis, placed in critical points along the highway: high-risk segments, service stations, and junctions. From the highway, through junctions, vehicles can reach towns. We focused on Rome, and our first task was to automatically identify the critical points (in addition to those manually identified by the city administrators) where a more fine-grained analysis should be carried out using camera-based information. In these places, the task is the same as for the suburban case: modelling vehicle behaviour and interpreting it using an automated reasoning system. In the following, we will provide a detailed description of our experiments in these settings along with the achieved results, highlighting the challenges encountered, the solutions implemented, and the performance achieved by our framework. Note that throughout the case study, we used real-world data for all tasks, and so our experimental outcomes can be considered as meaningful for practical application of our framework.

4.1. Positioning-based Critical Regions Identification

To identify areas with high traffic density, we initiated the process by applying clustering techniques. Our analysis focused on the flow of taxis in the city of Rome. For this purpose, we

utilized the Crowdad dataset (<https://ieee-dataport.org/open-access/crowdad-romataxi>, consulted July 26, 2023). Figure 2 provides an example of clustering applied to this context. In the first part of our analysis, we examined the flow of taxis in Rome using clustering. Figure 2 on the right presents a bird's-eye view of Rome, with the highlighted clusters representing areas of traffic congestion. The city centre emerges as the focal point of high traffic volume. Notably, the enlarged area on the right showcases Piazza Venezia, a prominent location known for significant intersections and crossings. This area was specifically selected for further analysis as one of the critical zones. The dashed circle in Figure 2 signifies the chosen region, which serves as our first urban traffic use case. Additionally, the star denotes the camera's position. The view depicted aligns with the one showcased in Figure 3. Piazza Venezia was chosen to exemplify the workflow of our proposed framework in a challenging and intricate traffic scenario.

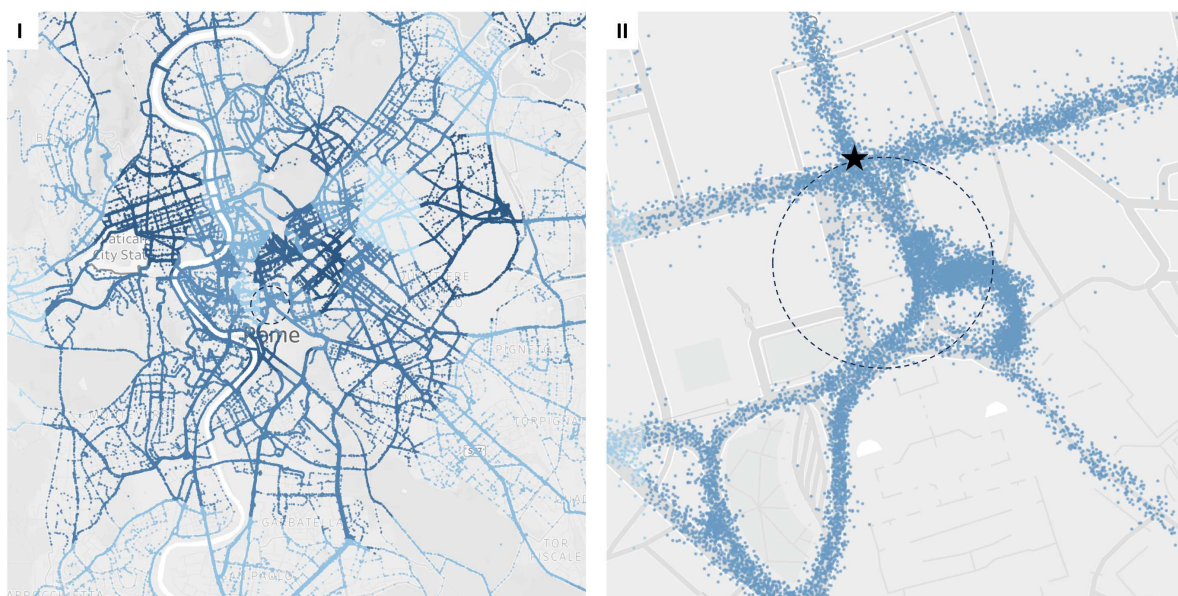


Figure 2. Clustering vehicle locations to identify jammed zones in Rome (Piazza Venezia).

4.2. Camera-based Critical Regions Identification

The vehicle identification interface described in Section 3.1.1 employs a diverse set of techniques to achieve real-time vehicle identification, boasting an average accuracy rate of over 80%. The platform exhibits the capability to identify various types of vehicles, including cars, trucks, buses, and motorcycles. Figure 3 showcases a screenshot capturing the tool in action. The right side of the screenshot presents real-time logs, featuring information such as the number of vehicles detected, the accuracy of the detection, and the duration taken to identify each vehicle. On the left side of the screenshot, frames are displayed, illustrating the bounding boxes enclosing the identified vehicles along with their corresponding accuracy percentages. Users are granted the flexibility to enable or disable vehicle tracking functionality, as well as the option to switch between different tracking algorithms, such as ByteTrack and BoT-SORT. The platform delivers real-time information pertaining to the vehicles present within a scene, ensuring up-to-date insights for monitoring purposes and decision-making.

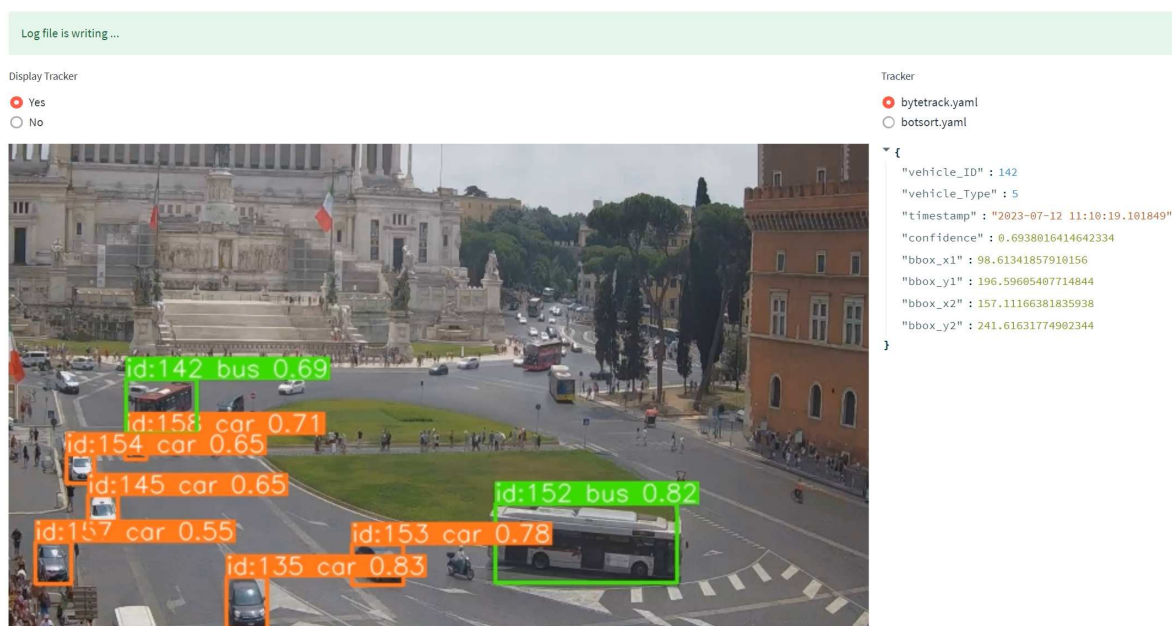


Figure 3. Vehicle detection application.

For this task, we used the streamed videos continuously provided by the SkylineWebcams streaming platform (www.skylinewebcams.com/webcam/italia.html, consulted July 26, 2023). Specifically, we selected two highway locations for the suburban setting (a two-way linear route, and a junction) and two urban locations in Rome for the urban setting (Piazza Venezia and Largo di Torre Argentina). Then, we ran the clustering algorithms discussed in the previous section on the whole set of points extracted from each video, obtaining the results shown in Figures 4 and 5 for the highway locations and those shown in Figures 6 and 7 for the locations in Rome. We note that in all cases the groups are more compact and well-defined for k-means and agglomerative clustering, while they are less, wider and with much more overlapping for spectral clustering, with db-scan and gaussian mixture models being somehow in the middle. So, we decided to adopt k-means clustering with a number of clusters automatically determined using the methods outlined in the previous section as the optimal one. Then, we considered the RoIs generated for each cluster by the convex hull algorithm to perform trajectory analysis within these specific areas of interest.

In particular, we obtained:

- 4 clusters/RoIs for the two-way linear highway location (Figure 4)
- 3 clusters/RoIs for the highway junction (Figure 5)
- 4 clusters/RoIs for Piazza Venezia (Figure 6)
- 4 clusters/RoIs for Largo di Torre Argentina (Figure 7)

Let's discuss each location specifically in the following subsections.

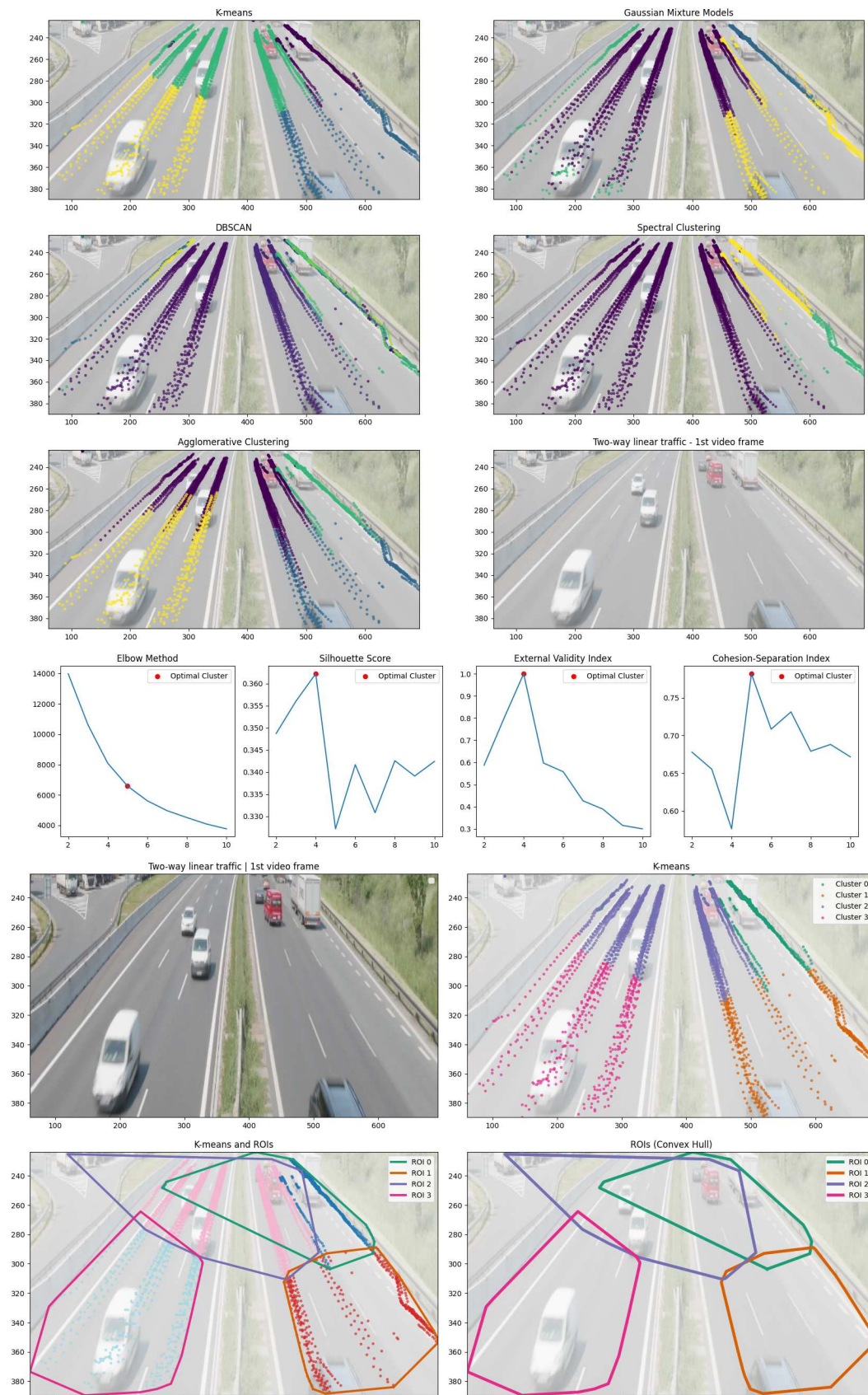


Figure 4. Suburban case: two-way highway segment.

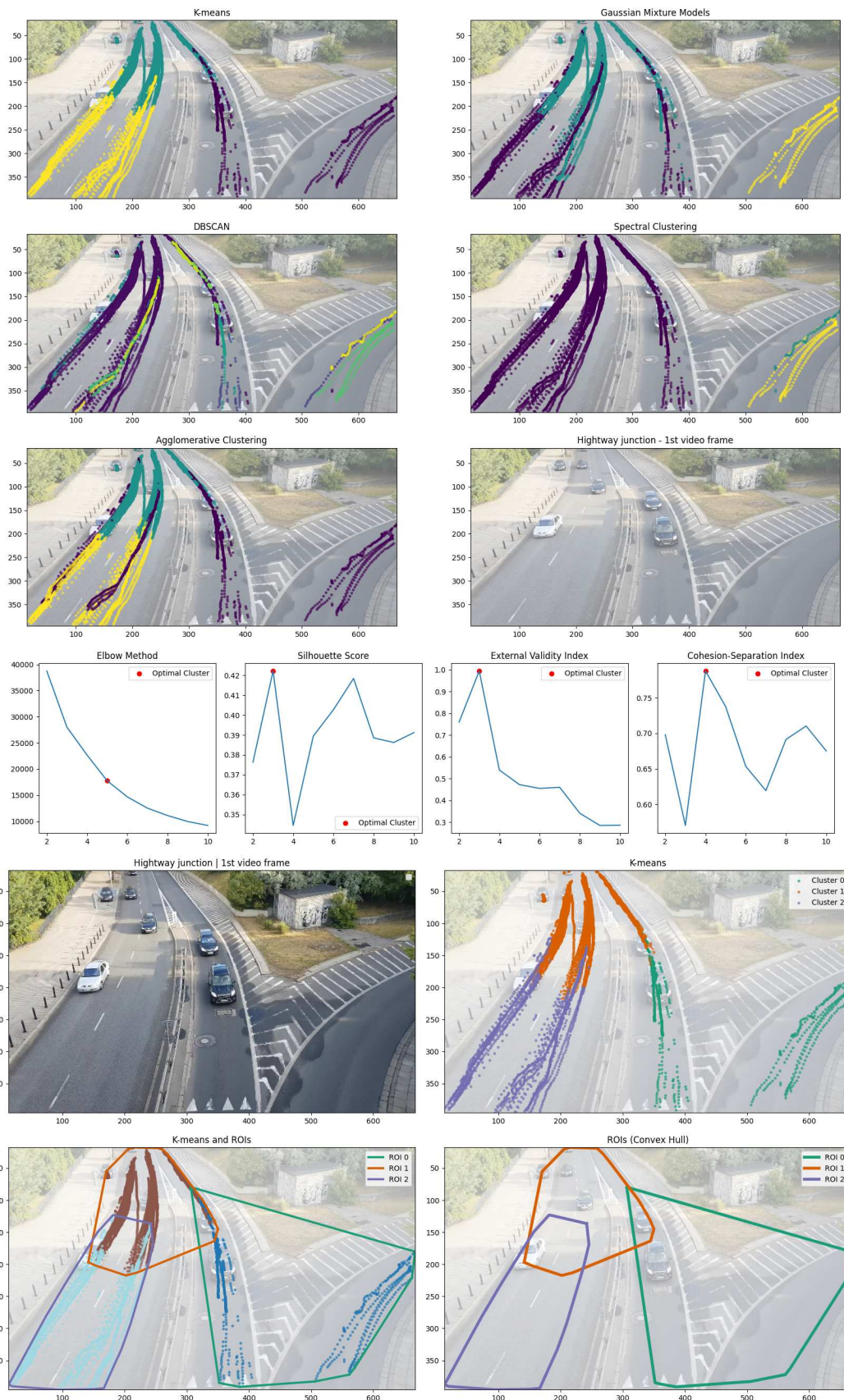


Figure 5. Sub-urban case: highway junction.

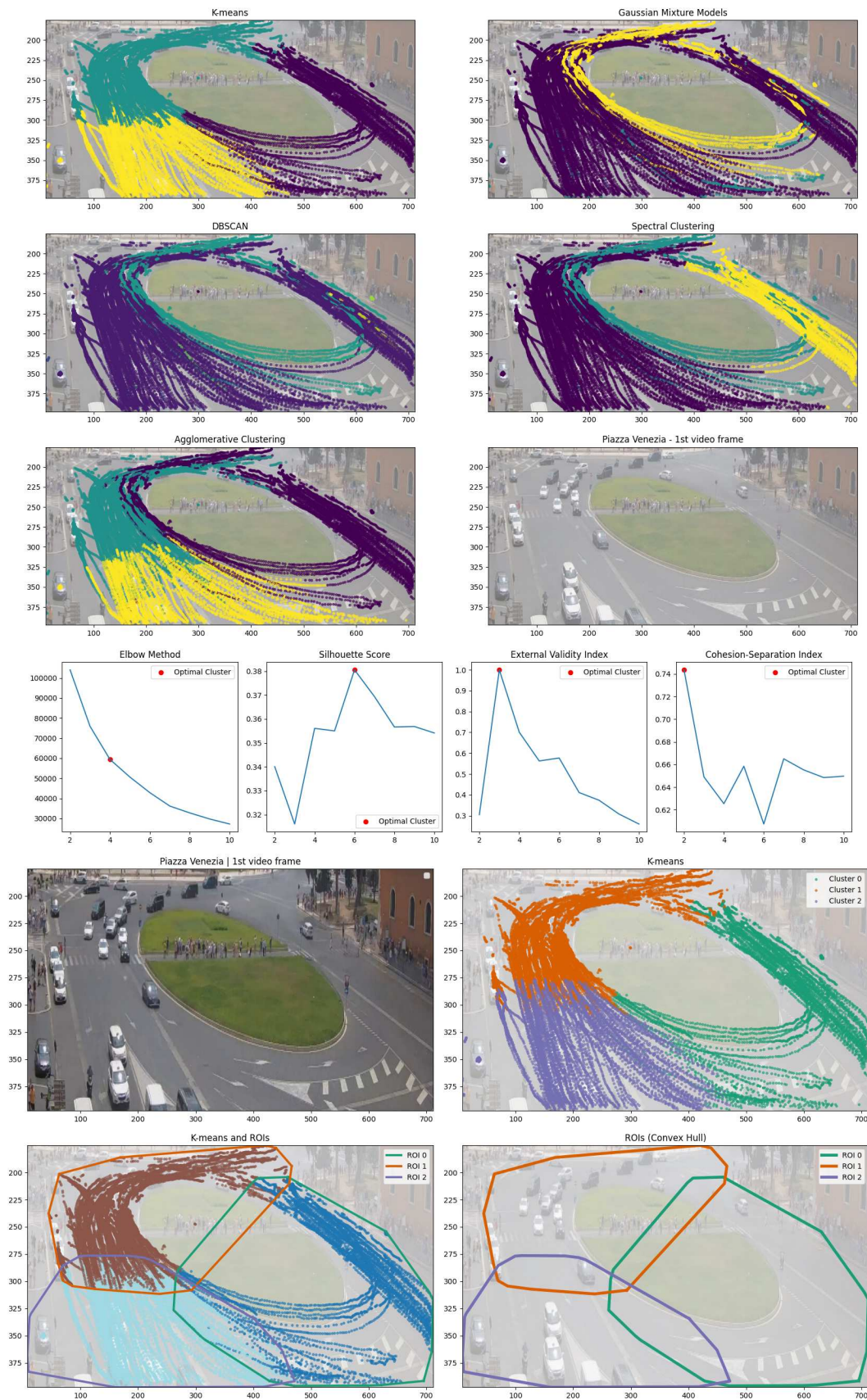


Figure 6. Urban case: Piazza Venezia in Rome.

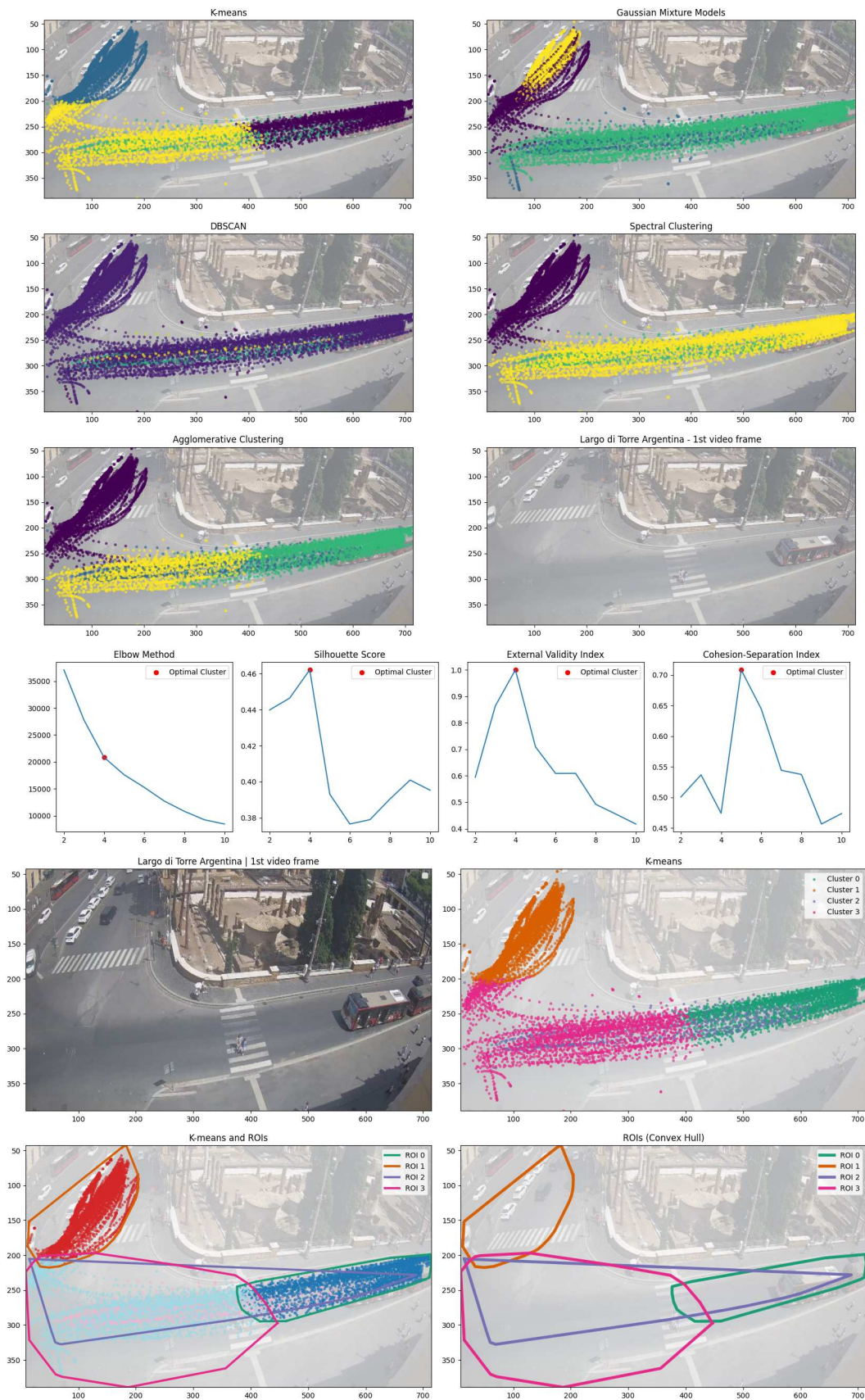


Figure 7. Urban case: Largo di Torre Argentina in Rome.

4.2.1. Suburban case: Two-way linear highway

The application of the k-means clustering algorithm to analyze the trajectories of vehicles on a two-lane, two-way highway captured by a camera yielded insightful results. The analyzed video was a 3-minute excerpt containing 3348 log records documenting vehicle detections and their respective positions. To determine the optimal number of clusters, various evaluation methods were employed: elbow method, silhouette analysis, external validity index, and cohesion separation index. After computing the results from all these methods, the average was found to be 4 as the optimal number of clusters. In addition to the generated average, the application's interface allows interested users to perform an analysis of trajectory graphs, compare individual vehicle speeds with the average speed at different points on the highway, and visualize the corresponding RoIs as the number of clusters varies. This enables the automatic extraction of the optimal number of clusters by the applied methods to be validated. From the analysis of the different evaluation methods, the silhouette emerged as the most reliable metric for determining the optimal number of clusters, as no pronounced elbow point was observed in the elbow method. The visual analysis of vehicle trajectories revealed that the two-way traffic led to the identification of two mandatory monitoring zones. The other two zones were considered an unnecessary addition. In Figure 4, four clusters are visible, with two defining the right lane and two defining the left lane. The inclusion of an exit within RoI2, which is primarily associated with the left lane, was due to the detection of a single vehicle falling below the convex hull. A clear distinction can be observed between RoI1 and RoI3, while RoI0 and RoI2 partially overlap. This suggests that detection confidence is lower in the distant region compared to the camera's direct field of view. Overall, the application of the k-means clustering algorithm compared to the others effectively and automatically identified vehicle trajectories, grouping them into meaningful clusters. This provides a solid foundation for further analysis and assessment of traffic conditions on the highway.

4.2.2. Suburban case: highway junction

The application of the k-means clustering algorithm to analyze the trajectories of vehicles on a two-lane highway with a junction, captured by a camera, yielded interesting results. The analyzed video was a 5-minute excerpt containing 9306 log records documenting vehicle detections and their respective positions. The optimal numbers of clusters returned by the evaluation methods applied to the k-means clustering algorithm are as follows: elbow method: 5, silhouette analysis: 3, external validity index: 3, cohesion separation index: 4. Averaging the results yielded 3.75, which was rounded up to 4 as the optimal number of clusters. Visual analysis supported this finding, confirming that the optimal number of clusters is 3, as indicated by the silhouette method. Among the three clusters visible in Figure 5, it can be observed that RoIs 1 and 2 define the left lane, while RoI 0 includes the area intersecting with the entrance for vehicles coming from the right. The area of particular interest is the exit area, represented by RoI 0, which was correctly highlighted by the clustering analysis. Overall, this analysis provided valuable insights into the trajectories of vehicles on a two-lane highway with an exit. The optimal number of clusters determined through various evaluation methods and confirmed by visual analysis allowed for the identification of relevant regions of interest, particularly in relation to the junction area.

4.2.3. Urban case: Piazza Venezia in Rome

The application of the k-means clustering algorithm to the analysis of vehicle trajectories captured by a camera in the urban area of Piazza Venezia in Rome yielded interesting results. The analyzed video was a 4.5 minute excerpt containing 24,331 log records of vehicle detection and their corresponding positions. The methods used to determine the optimal number of clusters for the k-means clustering returned the following values: elbow method: 4, silhouette method: 6, external validity index: 3, cohesion-separation index: 2. After considering the results from these methods, the average was

calculated, resulting in an optimal number of clusters of 3.75, rounded to 4. Although the optimal number suggested by the methods was 4, a visual analysis of the results led to the selection of 3 clusters as the optimal number. In Figure 6, three distinct clusters can be observed. The first RoI (RoI 0) encompasses the vehicles entering and exiting via del Corso (bottom right). The second RoI (RoI 1) includes the vehicles entering the roundabout and continuing along the road or turning towards the Fori Imperiali. The third RoI (RoI 2) comprises the vehicles turning left or proceeding straight ahead. RoIs 1 and 2 contain several intersections, crossing areas, entry points, and stops, making these areas particularly critical in terms of traffic density. Conversely, in RoI 0, the traffic flow, while consistently busy, appears more fluid. This differentiation highlighted by the clustering analysis confirms the effectiveness of the applied method in grouping vehicle trajectories based on spatial and movement characteristics. In conclusion, the application of the k-means clustering algorithm to vehicle trajectories in the urban area of Piazza Venezia allowed the identification of 4 clusters as the optimal number according to the employed methods. However, through a more in-depth visual analysis, it was determined that 3 clusters were more appropriate. These clusters revealed distinct areas of vehicle flow and specific characteristics of each RoI, validating the usefulness of clustering for analyzing vehicle trajectories in complex urban contexts like Piazza Venezia in Rome.

4.2.4. Urban case: Largo di Torre Argentina in Rome

The application of the k-means clustering algorithm to analyze the trajectories of vehicles in the urban area of Largo di Torre Argentina in Rome has yielded significant results. The analyzed video consisted of a 20-minute excerpt with 10,796 log records of vehicle detections and their corresponding positions. Several methods were employed to determine the optimal number of clusters (elbow, silhouette, external validity index, and cohesion separation index), resulting in an optimal number of 4 clusters. The average of the results from these different methods was 4.25, rounded down to 4. This optimal number of clusters was also confirmed through visual analysis of the data. Figure 7 displays the 4 identified clusters. RoI 0 encompasses vehicles that continue straight after a pedestrian crossing. RoI 1 comprises vehicles that stop at the traffic light and make a U-turn towards Piazza Venezia. RoI 2 represents vehicles that, after the traffic light, continue on via di Torre Argentina instead of making a U-turn. RoI 3 includes vehicles coming from Corso Vittorio Emanuele (bottom-right) or the traffic light (top-left) and proceeding on via di Torre Argentina (to the right) until the pedestrian crossing or straight until the pedestrian crossing before the taxi stop. The clustering analysis has highlighted critical areas, particularly pedestrian crossings at intersections. It has also identified the difference between uninterrupted routes and routes with stops due to pedestrian crossings or traffic lights.

4.3. Trajectory analysis and event detection

The user interface shows a graph of vehicle trajectories, with colour-coded points indicating velocity (see Figure 8). Additionally, histograms are included to depict the distribution of vehicle velocities.

These graphical representations play a vital role in conducting a comprehensive analysis and understanding of vehicle behaviour in various road scenarios. The visualization of vehicle trajectories provides valuable insights into movement directions and areas of congestion or smooth traffic flow. By utilizing velocity-based colour coding, areas with high or low speeds can be readily identified.

The histogram depicting vehicle velocities offers a holistic overview of the recorded speed distribution, enabling the detection of potential trends, such as predominant vehicle speeds or notable variations across different road situations.

Leveraging the data on vehicle speeds and trajectory directions enables the generation of meticulous event logs, serving as a solid foundation for input requirements in the behaviour modelling and traffic interpretation tasks.

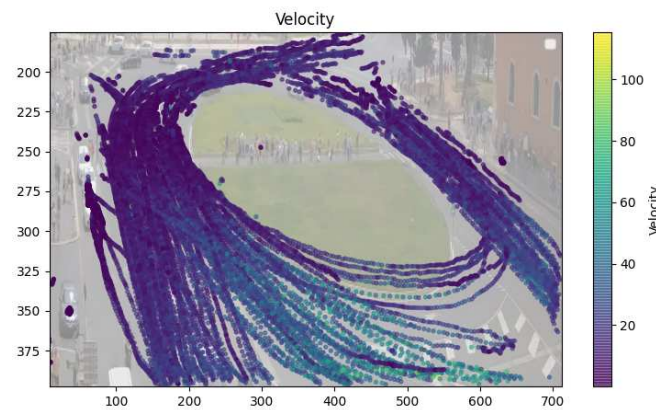


Figure 8. Piazza Venezia in Rome - velocity analysis.

4.4. Process-based Traffic Modeling and Supervision

Sub-urban traffic behaviour modelling was carried out on the real-world dataset TRAP (TRAffic Police), made available from the Italian National Police [4] and concerning traffic data collected every day along an Italian highway with the objective of using advanced data analysis techniques to prevent or predict road accidents, traffic congestion and road crimes. It includes the data obtained in the year 2016 by cameras placed on 27 highway gates, each recording the plate (by means of a Number Plate Reading Systems) and the time of any vehicle passing the gate. For privacy reasons, plate numbers are anonymized in the dataset. The highway has 16 service stations, each identified by the pair of gates between which it lies. On average, there are 2,086,919 plates and 12,965,495 detections per month. The aim is to learn the standard behaviour (i.e., routes) of single vehicles along the highway.

Here we show the behaviour of WoMan, used as proposed in [68], on the traffic of one day (01/02/2016). From the 102,840 cases (i.e., different vehicle routes along the highway) of the day, totalling 887,694 events (with an average of 8.63 events per route), we selected only those including more than one vehicle detection, considering as trivial the cases in which the vehicle entered the highway and then took the first available exit. More specifically, we took the first 1500 cases, spanning from midnight to 8:42 pm. They were processed by WoMan in 19.160 seconds, with the prequential behaviour depicted in Figure 9, where the X axis reports the cases and the Y axis the number of new behavioural components found in each case, with respect to the model learned from all previous cases.

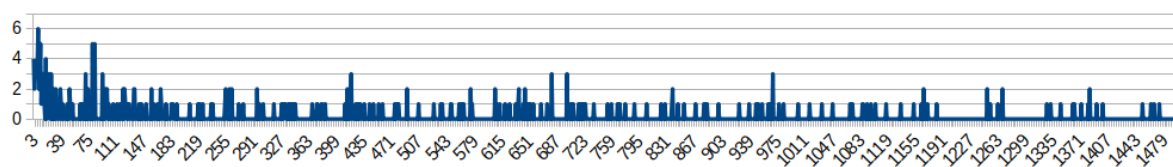


Figure 9. Learning trend for Italian highway gate-based vehicle behavior.

The trend to a convergence of the model is evident: peaks are lower and sparser as long as the plot proceeds to the right, with longer and longer blank spaces among peaks (meaning behaviours fully compliant with the learned model). Due to the compact size of the graphic, the full extent of blank spaces cannot be fully appreciated. The highest peaks happen before case 100, with no more than 3 new behavioural components after that case, and only 5 cases with 3 new components. The final model included 27 tasks (the gates) and 312 transitions. A study of the graphics for several days, and a comparison of the corresponding models, will allow defining when a model can be considered as sufficiently stable, and so when peaks should be considered as indicators of anomalous behaviours to be reported to the users.

For the urban case, we demonstrate this setting on the Piazza Venezia location, with the 4 identified RoIs. The experiment considered a video of 4.5 minutes, which produced 156 cases (i.e., vehicle traversals) for a total of 25,224 events and 12,456 activities, and an average of 79.85 activities per case. The model, learned in 300.554 seconds, included 23 tasks and 708 transitions. Figure 10 shows the learning trend using a prequential approach (each case is compared to the model learned so far, and used to refine the model before processing the next case, so that every case takes advantage of the most updated model available). Note that this approach is possible only thanks to the incremental feature provided by WoMan. The blue plot denotes the number of new tasks found in each case; the red plot the number of new transitions (i.e., basic task composition blocks in WoMan). We note that all tasks are learned by case 62, most of them are learned by case 11, with two new tasks found only in cases 23 and 62. Concerning transitions, 156 cases only are obviously insufficient to grasp all possible vehicle behaviours, still the indications of a convergence (lower and sparser peaks as long as the number of cases increases) are clearly visible in the plot. A few, isolated, high peaks are also visible on the right-hand side of the plot, where we would expect a convergence. These cases can be interpreted as outliers, and associated with anomalous vehicle behaviour. They might be proposed to human supervisors for checking and possible consequent actions. Note that this can be done in real-time, just while the video stream is flowing and being processed.

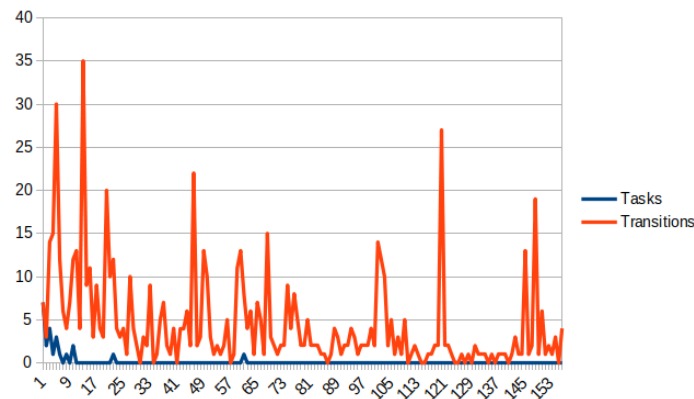


Figure 10. Learning trend for Piazza Venezia camera-based vehicle behavior.

4.5. Automated Reasoning for Traffic Interpretation

We provide a demonstrative use case for this feature concerning the Piazza Venezia location in Rome. While the video flows, the video analysis module generates vehicle-related information and asserts corresponding facts in the knowledge base based on the following predicates:

object(O,X0,X1,Y0,Y1,T) : object with identifier O is recognised in the scene at time T , enclosed in a bounding box with coordinates $X0, X1, Y0, Y1$.

next(T',T'') : time T'' follows time T' .

Whenever appropriate or useful (e.g., periodically every k frames processed) the reasoning engine GEAR is started to interpret what happened and return relevant notifications. It starts by deriving simple knowledge, as expressed by the following predicates that we have defined for this sample application:

move(O,X0,X1,Y0,Y1,T) : object O moved (by a considerable distance) at time T , where $X0, X1, Y0, Y1$ are the displacements of each coordinate of its bounding box.

enter(O,P,T) : object O entered place P (a RoI) at time T .

leave(O,P,T) : object O left place P (a RoI) at time T .

still(O,T) : object O stopped at time T .

halt(O,L,T) : object O stopped for a certain time period L starting from time T .

stay(O,P) : object O stayed in place P (a RoI).

placetime(O,P,T) : object O was in place P (a RoI) for a considerable time T .

status(O,T,S) : S is the status of the object O at time T , where S can be 'moving', 'still' or someplace (RoI) identifier.

meet(L,T,P) : objects in list L were in the same place P at time T .

wait(X,Y,T) : object X was still at time T , but is now in the same place as object Y .

distance(X00,X01,Y00,Y01,X10,X11,Y10,Y11,D) : D is the Euclidean distance between the coordinates $X00, X01, Y00, Y01$ and $X10, X11, Y10, Y11$ of two bounding boxes.

closetimes(X,Y,T,L) : L is the last timestamp, starting from T , in which X and Y were close to each other.

close(X,Y,T,L) : L is the amount of time for which X and Y were close to each other, starting from timestamp T .

accomplices(X,Y,T,D) : objects X and Y are close to each other for a certain amount of time D in the 'halt' state, and so still from a certain considerable time, at time T .

fast(O,T1,T2,D) : object O moved many times between timestamps $T1$ and $T2$ with distances greater than or equal to D .

Rules for defining these predicates, and for determining when a time period or displacement is 'relevant', are stored in the knowledge base using GEAR's formalism. These concepts are interrelated, meaning that some are defined upon others (e.g., *move/6* is defined in terms of *distance/9*).

Then, the reasoning may proceed with further concepts or situations of interest that the system is expected to identify (e.g., road traffic offences, etc.), at higher and higher levels of abstraction (i.e., not associated with a simple position occupied by a vehicle, but determined according to the overall vehicle behaviour and to its relationships to the road features and the behaviour of other vehicles). In our sample use case, we focused on the following situations:

- traffic jam;
- vehicle going faster than the maximum allowed speed;
- vehicle passing from forbidden zones of the road;
- vehicle stopping in places where the stop is forbidden;
- vehicle taking a wrong turn;
- vehicle going around the square in a loop;

and provided detailed descriptions thereof and of how they can be detected, possibly based on simpler situations that may not be relevant by themselves. These descriptions were formalized by a knowledge engineer and used to create a knowledge base, that consisted of several dozen rules. Applied to short videos taken from the Piazza Venezia location, this knowledge could allow the system to successfully identify several occurrences of those situations while the video was running and raise associated alarms, as well as to identify those situations upon specific requests of the stakeholder, such as "Did any vehicle go around the square in a loop from time X to time Y in the video?".

4.6. Visual Analytics for Traffic Understanding

In [73], a visual analytics approach was applied for outlier detection on the TRAP dataset [4], aiming to identify behavioural patterns of vehicles with anomalous routes. An anomalous route was defined as one where a vehicle visited multiple service areas, identified by irregular travel times in highway sections with service stations. The analysis was performed on a monthly basis. The method involved two main steps. First, Sequential Pattern Mining was used to discover drivers' behaviours. Each sequential pattern represented a cluster grouping vehicles with similar paths. Outlier detection was then applied to each cluster based on the time each vehicle spent traversing the path defined by the corresponding pattern. Vehicles deviating significantly from the meantime were considered outliers. Additional filters were applied to further reduce the number of outliers.

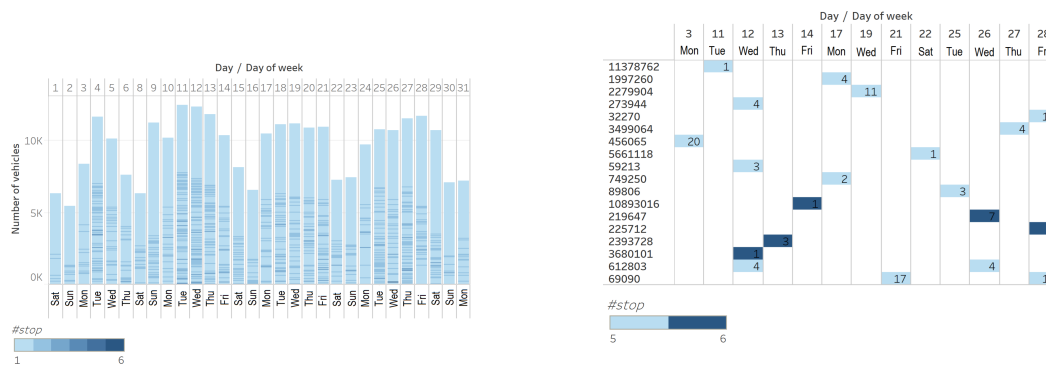


Figure 11. Graphics for visual analysis of traffic; left: number of outliers per day of a month (colour shows the number of service areas (#stop) for which a vehicle has been identified as an outlier); right: for each plate with anomalous behaviour, the day it had an anomalous behaviour, the number of days on which it was detected in the month and the #stop with the color [73].

The subsequent phase focused on the visual analysis of outliers involving only one service area. A bar chart was created for each month, showing the number of outliers per day. An example visualization for October is presented in Figure 11 (left), illustrating the distribution of outliers throughout the month. Further information about plates with anomalous behaviour was displayed in Figure 11 (right). This allowed for the classification of anomalous vehicles into two categories: sporadic irregularities and recurrent irregularities vehicles. For instance, some plates were detected on multiple days in October, consistently exhibiting the same anomalous behaviour, making them part of the recurrent irregularities category. Other plates showed anomalous behaviour only on specific days, indicating sporadic irregularities. The visual analytics approach proved effective in identifying patterns of anomalous behaviour and categorizing vehicles based on their recurrent or sporadic irregularities, which can be valuable for traffic understanding and anomaly detection in traffic management.

5. Conclusions

The efficient management of road traffic has always been a crucial challenge for urban transportation systems worldwide. Road traffic anomalies, such as accidents, congestion, and roadblocks, cause significant social and economic costs. Early detection of these anomalies is crucial for improving traffic safety, reducing travel time, and minimizing environmental pollution. Therefore, researchers have proposed various methods for detecting road traffic anomalies, including traditional statistical methods, machine learning, and computer vision.

In recent years, business process mining has emerged as an effective method for analyzing and optimizing complex systems. The method utilizes process models to identify deviations from expected patterns and improve system performance. In the context of road traffic anomaly detection, process mining can be used to analyze traffic flow data and identify deviations from expected traffic patterns.

Moreover, computer vision has also shown tremendous potential in traffic anomaly detection by analyzing video footage of the road network. By extracting relevant features and patterns, computer vision can detect various types of traffic anomalies in real time, such as accidents and traffic congestion.

However, the integration of computer vision, process mining and automated reasoning techniques for traffic anomaly detection remains a relatively unexplored research area. This paper proposes a framework that combines business process mining and computer vision techniques to detect road traffic anomalies in real-time. The framework can detect both sudden and gradual anomalies, such as accidents, congestion, and lane violations. The effectiveness of the proposed framework is evaluated using real-world traffic data collected from a major urban area.

Author Contributions: Conceptualization, E.B., D.D., S.F. and D.R.; methodology, E.B., D.D., S.F. and D.R.; software, E.B., D.D., S.F. and D.R.; validation, E.B., D.D., S.F. and D.R.; formal analysis, E.B., D.D., S.F. and D.R.; investigation, E.B., D.D., S.F. and D.R.; resources, E.B., D.D., S.F. and D.R.; data curation, E.B., D.D., S.F. and D.R.; writing—original draft preparation, E.B., D.D., S.F. and D.R.; writing—review and editing, E.B., D.D., S.F. and D.R.; visualization, E.B., D.D., S.F. and D.R.; supervision, S.F.; project administration, S.F.; funding acquisition, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by project FAIR – Future AI Research (PE00000013), spoke 6 – Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
KDD	Knowledge Discovery in Databases
GPS	Global Positioning System
CV	Computer Vision
VMOT	Visual multiple object tracking
VSD	Value Sensitive Design
BDI	Belief-Desire-Intention
KRR	Knowledge Representation and Reasoning
YOLO	You Only Look Once
FPS	frames per second
RoI	Regions of Interests
GMM	Gaussian Mixture Models
FOL	First-Order Logic
KB	Knowledge Base
WoMan	Workflow Manager
GEAR	General Engine for Automated Reasoning

References

1. The top 10 causes of death. <https://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death>. Accessed: 2023-02-27.
2. Savithamma, R.; Sumathi, R.; Sudhira, H. Reinforcement learning based traffic signal controller with state reduction. *Journal of Engineering Research* **2023**, *11*, 100017.
3. Elkin, D.; Vyatkin, V. IoT in Traffic Management: Review of Existing Methods of Road Traffic Regulation. In *Applied Informatics and Cybernetics in Intelligent Systems, Volume 3*; Silhavy, R., Ed.; Springer, 2020; Vol. 1226, *Advances in Intelligent Systems and Computing*, pp. 536–551. https://doi.org/10.1007/978-3-030-51974-2_50.
4. Leuzzi, F.; Ferilli, S., Eds. *Traffic Mining Applied to Police Activities - Proceedings of the 1st Italian Conference for the Traffic Police (TRAP-2017), Rome, Italy, October 25-26, 2017*, Vol. 728, *Advances in Intelligent Systems and Computing*. Springer, 2018. <https://doi.org/10.1007/978-3-319-75608-0>.
5. Loiseau, E.; Saikku, L.; Antikainen, R.; Droste, N.; Hansjürgens, B.; Pitkänen, K.; Leskinen, P.; Kuikman, P.; Thomsen, M. Green economy and related concepts: An overview. *Journal of cleaner production* **2016**, *139*, 361–371.
6. Sagioglu, S.; Sinanc, D. Big data: A review. In *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013, San Diego, CA, USA, May 20-24, 2013*; Fox, G.C.; Smari, W.W., Eds. IEEE, 2013, pp. 42–47. <https://doi.org/10.1109/CTS.2013.6567202>.
7. Tsai, C.; Lai, C.; Chao, H.; Vasilakos, A.V. Big data analytics: A survey. *J. Big Data* **2015**, *2*, 21. <https://doi.org/10.1186/s40537-015-0030-3>.

8. Tarlochan, F.; Ibrahim, M.I.M.; Gaben, B. Understanding traffic accidents among young drivers in Qatar. *International journal of environmental research and public health* **2022**, *19*, 514.
9. Hammad, H.M.; Ashraf, M.; Abbas, F.; Bakhat, H.F.; Qaisrani, S.A.; Mubeen, M.; Fahad, S.; Awais, M. Environmental factors affecting the frequency of road traffic accidents: A case study of sub-urban area of Pakistan. *Environmental Science and Pollution Research* **2019**, *26*, 11674–11685.
10. Government, A. *Impact of road trauma and measures to improve outcomes / Bureau of Infrastructure, Transport and Regional Economics*; Department of Infrastructure and Regional Development, Bureau of Infrastructure, Transport and Regional Economics Canberra, ACT, 2014; pp. x, 71 pages .:
11. Sofuoglu, S.E.; Aviyente, S. GLOSS: Tensor-based anomaly detection in spatiotemporal urban traffic data. *Signal Processing* **2022**, *192*, 108370.
12. Kong, X.; Gao, H.; Alfarraj, O.; Ni, Q.; Zheng, C.; Shen, G. Huad: Hierarchical urban anomaly detection based on spatio-temporal data. *IEEE Access* **2020**, *8*, 26573–26582.
13. Lindemans, B.; Maschler, B.; Sahlab, N.; Weyrich, M. A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry* **2021**, *131*, 103498.
14. Di Mauro, N.; Ferilli, S. Unsupervised LSTMs-based learning for anomaly detection in highway traffic data. In Proceedings of the 24th International Symposium on Foundations of Intelligent Systems, ISMIS 2018, Limassol, Cyprus, October 29–31, 2018, Proceedings 24. Springer, 2018, pp. 281–290.
15. D'Andrea, E.; Marcelloni, F. Detection of traffic congestion and incidents from GPS trace analysis. *Expert Systems with Applications* **2017**, *73*, 43–56.
16. Ahmadi, S.A.; Ghorbanian, A.; Mohammadzadeh, A. Moving vehicle detection, tracking and traffic parameter estimation from a satellite video: A perspective on a smarter city. *International journal of remote sensing* **2019**, *40*, 8379–8394.
17. Dubská, M.; Herout, A.; Sochor, J. Automatic Camera Calibration for Traffic Understanding. In Proceedings of the British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014; Valstar, M.F.; French, A.P.; Pridmore, T.P., Eds. BMVA Press, 2014.
18. Hamilton, J.D. *Time series analysis*; Princeton university press, 2020.
19. Cryer, J.D. *Time series analysis*; Vol. 286, Duxbury Press Boston, 1986.
20. Trinh, H.D.; Giupponi, L.; Dini, P. Mobile traffic prediction from raw data using LSTM networks. In Proceedings of the 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC). IEEE, 2018, pp. 1827–1832.
21. Feng, J.; Chen, X.; Gao, R.; Zeng, M.; Li, Y. Deeptp: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Network* **2018**, *32*, 108–115.
22. Thiagarajan, A.; Ravindranath, L.; LaCurts, K.; Madden, S.; Balakrishnan, H.; Toledo, S.; Eriksson, J. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In Proceedings of the Proceedings of the 7th ACM conference on embedded networked sensor systems, 2009, pp. 85–98.
23. Xu, F.; Lin, Y.; Huang, J.; Wu, D.; Shi, H.; Song, J.; Li, Y. Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach. *IEEE Transactions on Services Computing* **2016**, *9*, 796–805. <https://doi.org/10.1109/TSC.2016.2599878>.
24. Wang, J.; Li, Y.; Gao, R.X.; Zhang, F. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems* **2022**, *63*, 381–391.
25. Dias, T.; Oliveira, N.; Sousa, N.; Praça, I.; Sousa, O. A Hybrid Approach for an Interpretable and Explainable Intrusion Detection System. In Proceedings of the 21st International Conference on Intelligent Systems Design and Applications (ISDA 2021); Abraham, A.; Gandhi, N.; Hanne, T.; Hong, T.; Rios, T.N.; Ding, W., Eds. Springer, 2021, Vol. 418, *Lecture Notes in Networks and Systems*, pp. 1035–1045. https://doi.org/10.1007/978-3-030-96308-8_96.
26. De, T.; Giri, P.; Mevawala, A.; Nemani, R.; Deo, A. Explainable AI: A hybrid approach to generate human-interpretable explanation for deep learning prediction. *Procedia Computer Science* **2020**, *168*, 40–48.
27. Mehdizadeh, A.; Cai, M.; Hu, Q.; Alamdar Yazdi, M.A.; Mohabbati-Kalejahi, N.; Vinel, A.; Rigdon, S.E.; Davis, K.C.; Megahed, F.M. A review of data analytic applications in road traffic safety. Part 1: Descriptive and predictive modeling. *Sensors* **2020**, *20*, 1107.
28. Huang, C.; Li, Y.; Nevatia, R. Multiple target tracking by learning-based hierarchical association of detection responses. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *35*, 898–910.

29. Santhosh, K.K.; Dogra, D.P.; Roy, P.P. Anomaly detection in road traffic using visual surveillance: A survey. *ACM Computing Surveys (CSUR)* **2020**, *53*, 1–26.
30. Karle, P.; Geisslinger, M.; Betz, J.; Lienkamp, M. Scenario understanding and motion prediction for autonomous vehicles-review and comparison. *IEEE Transactions on Intelligent Transportation Systems* **2022**, *23*, 16962–16982.
31. Umbrello, S.; Yampolskiy, R.V. Designing AI for explainability and verifiability: A value sensitive design approach to avoid artificial stupidity in autonomous vehicles. *International Journal of Social Robotics* **2022**, *14*, 313–322.
32. Friedman, B. Value-sensitive design. *interactions* **1996**, *3*, 16–23.
33. Fichera, L.; Marletta, D.; Nicosia, V.; Santoro, C. Flexible robot strategy design using belief-desire-intention model. In Proceedings of the Research and Education in Robotics-EUROBOT 2010: International Conference, Rapperswil-Jona, Switzerland, May 27-30, 2010, Revised Selected Papers. Springer, 2011, pp. 57–71.
34. Li, N.; Liu, S.; Liu, Y.; Zhao, S.; Liu, M. Neural Speech Synthesis with Transformer Network. In Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press, 2019, pp. 6706–6713. <https://doi.org/10.1609/aaai.v33i01.33016706>.
35. Dong, J.; Chen, S.; Zong, S.; Chen, T.; Labi, S. Image transformer for explainable autonomous driving system. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021, pp. 2732–2737.
36. Shen, Y.; Jiang, S.; Chen, Y.; Yang, E.; Jin, X.; Fan, Y.; Campbell, K.D. To explain or not to explain: A study on the necessity of explanations for autonomous vehicles. *arXiv preprint arXiv:2006.11684* **2020**.
37. Levesque, H.J. Knowledge representation and reasoning. *Annual review of computer science* **1986**, *1*, 255–287.
38. Xiao, L.; Gerth, J.; Hanrahan, P. Enhancing visual analysis of network traffic using a knowledge representation. In Proceedings of the 2006 IEEE symposium on visual analytics science and technology. IEEE, 2006, pp. 107–114.
39. Hernández, J.; Cuenca, J.; Molina, M. Real-time traffic management through knowledge-based models: The TRYS approach. *ERUDIT Tutorial on Intelligent Traffic Management Models, Helsinki, Finland* **1999**.
40. Cuenca, J.; Hernández, J.; Molina, M. Knowledge-based models for adaptive traffic management systems. *Transportation Research Part C: Emerging Technologies* **1995**, *3*, 311–337.
41. Hülsen, M.; Zöllner, J.M.; Weiss, C. Traffic intersection situation description ontology for advanced driver assistance. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), 2011, pp. 993–999. <https://doi.org/10.1109/IVS.2011.5940415>.
42. Choi, C.; Wang, T.; Esposito, C.; Gupta, B.B.; Lee, K. Sensored Semantic Annotation for Traffic Control Based on Knowledge Inference in Video. *IEEE Sensors Journal* **2021**, *21*, 11758–11768. <https://doi.org/10.1109/JSEN.2020.3048758>.
43. Toal, A.F.; Buxton, H. Spatio-temporal Reasoning within a Traffic Surveillance System. In Proceedings of the Second European Conference on Computer Vision - ECCV'92,, Santa Margherita Ligure, Italy, May 19-22, 1992; Sandini, G., Ed. Springer, 1992, Vol. 588, *Lecture Notes in Computer Science*, pp. 884–892. https://doi.org/10.1007/3-540-55426-2_103.
44. Logi, F.; Ritchie, S.G. Development and evaluation of a knowledge-based system for traffic congestion management and control. *Transportation Research Part C: Emerging Technologies* **2001**, *9*, 433–459.
45. Manley, E.; Cheng, T. Exploring the role of spatial cognition in predicting urban traffic flow through agent-based modelling. *Transportation Research Part A: Policy and Practice* **2018**, *109*, 14–23.
46. Cucchiara, R.; Piccardi, M.; Mello, P. Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE transactions on intelligent transportation systems* **2000**, *1*, 119–130.
47. Stockman, G.; Shapiro, L.G. *Computer vision*; Prentice Hall PTR, 2001.
48. Rezaei, F.; Yazdi, M. Real-time crowd behavior recognition in surveillance videos based on deep learning methods. *Journal of Real-Time Image Processing* **2021**, *18*, 1669–1679.
49. Meynberg, O.; Cui, S.; Reinartz, P. Detection of high-density crowds in aerial images using texture classification. *Remote Sensing* **2016**, *8*, 470.
50. Meier, E.B.; Ade, F. Tracking cars in range image sequences. In Proceedings of the Proceedings of Conference on Intelligent Transportation Systems. IEEE, 1997, pp. 105–110.

51. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Computer Science* **2022**, *199*, 1066–1073.
52. Lan, W.; Dang, J.; Wang, Y.; Wang, S. Pedestrian detection based on YOLO network model. In Proceedings of the 2018 IEEE international conference on mechatronics and automation (ICMA). IEEE, 2018, pp. 1547–1551.
53. Aboah, A.; Wang, B.; Bagci, U.; Adu-Gyamfi, Y. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5349–5357.
54. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-object Tracking by Associating Every Detection Box. In Proceedings of the 17th European Conference on Computer Vision - ECCV 2022, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII; Avidan, S.; Brostow, G.J.; Cissé, M.; Farinella, G.M.; Hassner, T., Eds. Springer, 2022, Vol. 13682, *Lecture Notes in Computer Science*, pp. 1–21. https://doi.org/10.1007/978-3-031-20047-2_1.
55. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv preprint arXiv:2206.14651* **2022**.
56. Amit, Y.; Felzenszwalb, P.; Girshick, R. Object detection. *Computer Vision: A Reference Guide* **2020**, pp. 1–9.
57. Omran, M.G.; Engelbrecht, A.P.; Salman, A. An overview of clustering methods. *Intelligent Data Analysis* **2007**, *11*, 583–605.
58. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014). IEEE, 2014, pp. 232–238.
59. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *9*, 1295.
60. Ackermann, M.R.; Blömer, J.; Kuntze, D.; Sohler, C. Analysis of agglomerative clustering. *Algorithmica* **2014**, *69*, 184–215.
61. Reynolds, D.A.; et al. Gaussian mixture models. *Encyclopedia of biometrics* **2009**, 741.
62. Jia, H.; Ding, S.; Xu, X.; Nie, R. The latest research progress on spectral clustering. *Neural Computing and Applications* **2014**, *24*, 1477–1486.
63. Cui, M.; et al. Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance* **2020**, *1*, 5–8.
64. Dedeoğlu, Y.; Töreysin, B.U.; Güdükbay, U.; Çetin, A.E. Silhouette-based method for object classification and human action recognition in video. In Proceedings of the Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13, 2006. Proceedings 9. Springer, 2006, pp. 64–77.
65. Rendón, E.; Abundez, I.M.; Gutierrez, C.; Zagal, S.D.; Arizmendi, A.; Quiroz, E.M.; Arzate, H.E. A Comparison of Internal and External Cluster Validation Indexes. In Proceedings of the the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications; World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, Wisconsin, USA, 2011; AMERICAN-MATH'11/CEA'11, p. 158–163.
66. Lee, K.M.; Lee, K.M.; Lee, C.H. Statistical cluster validity indexes to consider cohesion and separation. In Proceedings of the 2012 international conference on fuzzy theory and its applications (ifuzzy2012). IEEE, 2012, pp. 228–232.
67. McCallum, D.; Avis, D. A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters* **1979**, *9*, 201–206. [https://doi.org/https://doi.org/10.1016/0020-0190\(79\)90069-3](https://doi.org/https://doi.org/10.1016/0020-0190(79)90069-3).
68. Ferilli, S.; Redavid, D. A process mining approach to the identification of normal and suspect traffic behavior. In Proceedings of the 1st Italian Conference for the Traffic Police (TRAP-2017): Traffic Mining Applied to Police Activities;. Springer, 2018, pp. 37–56.
69. Ferilli, S. Woman: Logic-based workflow learning and management. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2013**, *44*, 744–756.
70. Ferilli, S.; Esposito, F.; Redavid, D.; Angelastro, S. Predicting process behavior in woman. In Proceedings of the AI* IA 2016 Advances in Artificial Intelligence: XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29–December 1, 2016, Proceedings XV. Springer, 2016, pp. 308–320.
71. Ferilli, S.; Angelastro, S. Activity prediction in process mining using the WoMan framework. *Journal of Intelligent Information Systems* **2019**, *53*, 93–112.

72. Ferilli, S. GEAR: A General Inference Engine for Automated MultiStrategy Reasoning. *Electronics* **2023**, *12*.
73. Buono, P.; Legretto, A.; Ferilli, S.; Angelastro, S. A Visual Analytic Approach to Analyze Highway Vehicular Traffic. In Proceedings of the 22nd International Conference Information Visualisation (IV), 2018, pp. 204–209. <https://doi.org/10.1109/iV.2018.00044>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.