Interpretable Representation and Customizable Retrieval of Traffic Congestion Patterns Using Causal Graph-Based Feature Associations

Nguyen, Tin T.; Calvert, Simeon C.; Li, Guopeng; van Lint, Hans

DOI
10.1007/s42421-024-00106-0

Publication date
2024

Document Version
Final published version

Published in
Data Science for Transportation

Data Science for Transportation (2024) 6:18
https://doi.org/10.1007/s42421-024-00106-0

RESEARCH

Interpretable Representation and Customizable Retrieval of Traffic Congestion Patterns Using Causal Graph-Based Feature Associations

Tin T. Nguyen[1] · Simeon C. Calvert[1] · Guopeng Li[1,2] · Hans van Lint[1]

Received: 8 May 2024 / Revised: 10 July 2024 / Accepted: 7 August 2024 / Published online: 30 August 2024

Abstract

The substantial increase in traffic data offers new opportunities to inspect traffic congestion dynamics from different perspectives. This paper presents a novel framework for the interpretable representation and customizable retrieval of traffic congestion patterns using causal relation graphs, which harnesses many of these opportunities. By integrating domain knowledge with innovative data management techniques, we address the challenges of effectively handling and retrieving the growing volume of traffic data for diverse analytical purposes. The framework leverages causal graphs to encode traffic congestion patterns, capturing fundamental phenomena and their spatiotemporal relationships, thus facilitating an interpretable high-level view of traffic dynamics. Moreover, a customizable similarity measurement function is introduced based on inexact graph matching, allowing users to tailor the retrieval process to specific requirements. This framework's capability to retrieve customizable and interpretable congestion patterns is demonstrated through extensive experiments with real-world highway traffic data in the Netherlands, highlighting its value in supporting diverse data-driven studies and applications.

Keywords Traffic congestion patterns · Highway traffic · Knowledge-guided data retrieval · Graph matching

Introduction

The increasing mass of traffic data serves as a vital foundation for research and practical applications in traffic and transportation. Traffic state data encompass crucial information for understanding various aspects of traffic, with none more significant than traffic congestion, a major nuisance of traffic flow. Traffic congestion is characterized by low speed and high vehicle density, affecting both spatial and temporal dimensions. Hence, congestion can be effectively visualized on space-time maps. For instance, a low-speed region on space-time maps represents that a stretch of road is congested for a certain duration. These two-dimensional images are the so-called *traffic congestion patterns*.

Traffic congestion data are essential for numerous applications in travel services and traffic management. One of its primary uses is in traffic state forecasting and arrival time estimation. Platforms providing traffic and travel services use real-time data to predict the evolution of congestion (Li et al. 2017, 2021) and to estimate the arrival time (Van Lint et al. 2005). There has been significant research interest in data-driven forecasting models, particularly those based on deep learning, which effectively utilize large datasets of traffic congestion patterns. For example, Ma et al. (2017) build Convolutional Neural Networks (CNN) and train the model on congestion pattern datasets to predict corridor-level traffic states in the short future. A comprehensive review of this field is provided by Yin et al. (2021). Besides forecasting, traffic congestion data can assist road authorities in optimizing traffic control schemes (Calvert et al. 2018; van de Weg et al. 2018) and in evaluating the effectiveness of implemented controls (Wang et al. 2006; Kim et al. 2013).

✉ Guopeng Li
g.li-5@tudelft.nl

Tin T. Nguyen
nguyenthientin@gmail.com

Simeon C. Calvert
s.c.calvert@tudelft.nl

Hans van Lint
j.w.c.vanlint@tudelft.nl

[1] Civil Engineering and Geoscience, Delft University of Technology, Stevinweg 1, 2628 CN Delft, Zuid-Holland, The Netherlands

[2] Mechanical Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, Zuid-Holland, The Netherlands

Moreover, congestion patterns offer valuable insights for large-scale traffic management. Lopez et al. (2017) examines the network-level day-to-day regularity of highway congestion in the Netherlands, revealing a global picture of traffic dynamics. In addition, although congestion patterns are usually macroscopic traffic data, they also play a crucial role in understanding microscopic traffic phenomena. Researchers can calibrate and test driving models (Calvert et al. 2011) and traffic flow models (Vlahogianni et al. 2005) using congestion data. For instance, van Lint et al. (2020) analyze the relationship between highway Congestion Warning Systems (CWS) and congestion patterns. The authors find that differences in maximum deceleration distributions when vehicles enter the first stop-and-go wave in congestion can be used as a surrogate (un)safety indicator. These diverse uses highlight the significant value of congestion pattern data in both theoretical research and practical applications.

Meanwhile, as congestion pattern data is gaining more attention, its growing size poses novel challenges for developing an effective data management and retrieval system. From a practical perspective, the system should return the desired congestion patterns based on users' specific requirements on interested traffic phenomena, congestion scale, or congestion structures. For example, for studying relationships between congestion and accidents, heavy homogeneous congestion with one standing bottleneck is usually needed (Wang et al. 2009). If focusing on controlling multiple highway on-ramps, then large-scale congestion patterns containing multiple interactive bottlenecks are ideal data. Such a data retrieval system can save researchers time on data acquisition and expedite diverse research directions.

To build a data retrieval system that can support various analytical purposes, two functionalities are critically important:

- **Interpretability of pattern representation:** How to represent high-level features of congestion patterns in a way that can explain underlying traffic phenomena?
- **Customization of pattern retrieval:** How to define a customizable similarity measurement between congestion patterns based on the interpretable representation?

Interpretability enables users to specify what patterns they want based on domain knowledge, and customizability ensures that the system can cater for different requirements according to users' definitions of similarity. Many studies have addressed these two concerns from different perspectives in the literature.

In congestion pattern representation, commonly used features are low-level information such as colour, shape and texture (Andrews Sobral et al. 2013). Krishnakumari et al. (2017) used active shape methods to classify congestion patterns based on contours. Other methods, such as variational auto-encoders (Boquet et al. 2020) based on deep learning, compress high-dimensional images into low-dimensional concatenated feature vectors. These methods result in unexplainable visual features with no conceptual meaning. To imbue interpretability into congestion pattern representations, connecting extracted features to the underlying meaningful traffic phenomena is important. Typical traffic phenomena include congestion bottlenecks, transient traffic, traffic oscillations (stop-and-go waves), homogeneous congestion (Helbing et al. 2009), etc. For example, Zheng et al. (2011) apply wavelet transform to recognize these phenomena and analyze freeway traffic. Similarly, Nguyen et al. (2019) applied image segmentation methods to detect traffic oscillations and homogeneous congestion. The paper shows that compared to low-level 2D images, these traffic-specific elements can lead to sharper clusters of congestion patterns (Bay et al. 2008), revealing the regularity of highway traffic. Other studies focus on extracting one specific pattern feature. For example, Chen et al. (2004) considers all adjacent detector pairs and uses rule-based methods to recognize bottleneck activation. Wieczorek et al. (2010) further conducted a sensitivity analysis of the parameters used in Chen's method. Zhao et al. (2014) used extended spectral envelope method to detect stop-and-go waves. For more relevant papers, we refer the readers to Nguyen et al. (2019) for an overview.

Although there are many pattern representation and recognition studies, most of these methods focus on isolated congestion regions and separate sub-features. This approach results in local features that are not integrated into a cohesive structure, which is necessary for representing the broader, macroscopic evolution of complex congestion patterns. Such fragmentation significantly constrains the scalability of these representation methods, limiting their effectiveness in broader traffic management applications.

Compared to congestion pattern recognition, traffic data retrieval systems are less discussed. There are two common approaches in the literature, namely the text-based approach and content-based approach (Datta et al. 2008). The text-based approach labels each image by keywords when preparing the database. By specifying query keywords, relevant patterns can be easily identified and retrieved. One advantage of this approach is the simplicity of the implementation mechanism. However, image annotation is often done manually, which is time-consuming and prone to errors due to a large number of available items. On the other hand, the content-based approach is performed by providing a query example image when retrieving data. The system automatically extracts features from this query image and searches for matching images from the database. We refer the readers to Zhou et al. (2017) for a literature review.

However, in data retrieval, "similarity" is a subjective concept. Different studies may require "similar" data in

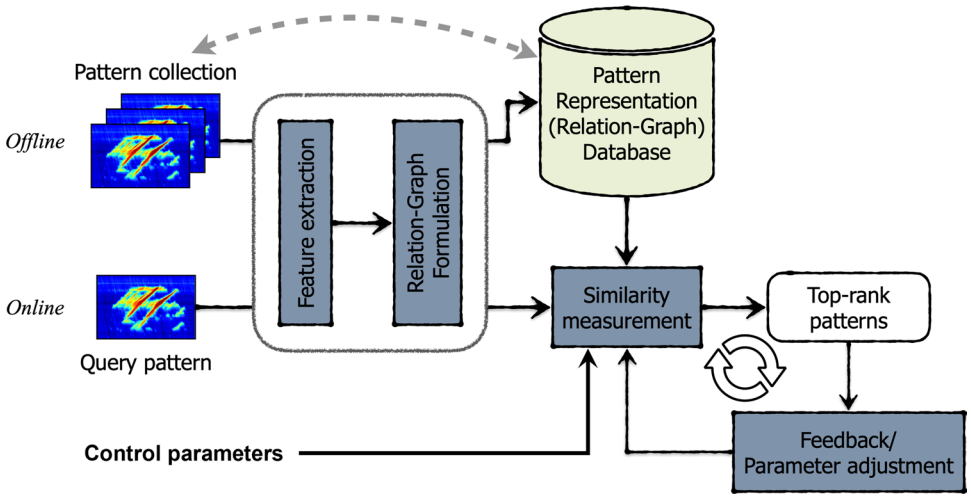


Fig. 1 The framework for congestion pattern retrieval

different aspects. Existing content-based approaches tend to use a pre-defined, fixed similarity measurement, which limits their capabilities to retrieve user-defined data. To the best of our knowledge, how to define customizable similarity measures based on interpretable congestion pattern representations has not been discussed in the literature.

In summary, two research gaps in interpretable pattern representation and customizable similarity measures restrict the potential benefits of congestion pattern retrieval systems in supporting different research purposes and applications. This paper aims to fill these gaps by developing a novel methodological framework.

This paper proposes a congestion pattern management and retrieval framework that combines domain-knowledge-driven interpretable congestion representation and customizable similar pattern retrievals. The contributions are summarized as follows:

- Proposition of a novel methodological framework to represent a congestion pattern as a causal relation graph of domain-specific traffic features.
- Proposition of a customizable similarity function between two relation graphs of congestion patterns based on inexact graph matching methods.
- Evaluation of the efficacy and efficiency of the built pattern management system using a large-scale real-world dataset.

The paper is organized as follows. "Methodological Framework" section firstly presents the methodological framework. Next, "Causal Relation-Graph-Based Pattern Representation" section describes the process of extracting relevant features and constructing the so-called relation graphs as the representation for congestion patterns. The measurement of similarity between two patterns is presented in "Customizable Similarity Measurement" section. In "Experiments" section, we describe an extensive experiment for evaluating the proposed method. Finally, "Conclusion and Perspectives" section concludes this study and gives several relevant research directions.

Methodological Framework

Figure 1 illustrates the framework of the proposed pattern retrieval system. This methodology comprises two key components, the so-called *pattern representation* and *similarity measurement*. The pattern representation module first determines and extracts core traffic features (or traits) from a congestion pattern. Next, these features are integrated into a single causal relation graph to interpret high-level characteristics of congestion patterns. The similarity measurement module determines the degree of resemblance between two patterns. Its inputs include two relation graphs (representing two patterns, one is a query pattern) *and* user's customizable control parameters (representing what "types" of similarity users want to use) that define the similarity function. The output is a corresponding similarity score.

In the offline stage, patterns are processed into relation graphs and registered in a database. In the online stage, the system retrieves top-ranked patterns that conform best to the query pattern based on the user-defined similarity measurement. If some retrieval outcomes do not meet certain expectations, users can adjust control parameter settings and repeat the retrieval process to improve results. The following sections dive deeper into the framework after the two key components of pattern representation and similarity measurement are explained. To better explain

how the methodology works, this data retrieval system that covers the congestion patterns collected on all highways in the Netherlands is made available online: https://mirrors-ndw.citg.tudelft.nl/webapp/app-cosi/.[1]

Causal Relation-Graph-Based Pattern Representation

This section presents the conversion of a low-level traffic congestion pattern into a high-level causal relation graph. For clarity, we first introduce several key concepts from the lowest to the highest level:

- **Congestion patterns:** A congestion pattern represents the traffic state on a particular road segment over a certain period. In essence, it is a two-dimensional matrix (spatio-temporal image) of traffic states (e.g., speed, flow or density). A pixel value represents the traffic state observed at a certain location and time.
- **Traffic primitives:** A traffic primitive refers to a spatio-temporal region representing a specified well-acknowledged traffic phenomenon or characteristic.
- **Causal relation graphs:** A causal relation graph is a directed acyclic graph whose nodes are traffic primitives and edges are their spatiotemporal relationships. The mathematical definition will be explained later.

Therefore, the pattern representation module contains two steps connecting the three levels. *Feature extraction* first extracts traffic primitives from original congestion patterns. Then *relation-graph formulation* synthesizes a causal relation-graph from extracted traffic primitives. The key principle for both steps is keeping the interpretability of the used representation.

Feature Extraction

Feature extraction aims to identify instances of widely acknowledged traffic phenomena from the original image of congestion patterns. In this work, we define three basic traffic primitives. They are *traffic bottlenecks*, *wide-moving traffic disturbances* (also called traffic oscillations in some papers), and *homogeneously heavy congestion*. These three components are the most important features that determine the evolution of traffic congestion. Their definitions and detection methods are described below:

Traffic Bottleneck (B)

A traffic bottleneck is a specific point or section of a road where traffic speeds are severely impeded or restricted, usually caused by over-saturated ramps. Traffic bottleneck detection has a large body of literature. In this study, we adopt the framework proposed by Nguyen et al. (2021), which identifies bottleneck location and activation time from speed maps. Furthermore, the used method also extracts the boundaries of upstream congestion regions. In principle, congestion regions are identified by applying the active contour model without edges (Chan and Vese 2001)—a well-known image segmentation technique in computer vision. The model formulates congestion as foreground and free-flowing regions as background in the segmentation problem. Bottleneck locations are detected by observing speed gradients along the direction of characteristic waves. Discontinuities (drops of speed upstream) of traffic speeds are associated with possible bottleneck activation. Both primary and secondary bottlenecks can be identified successfully using this method. We refer to the original paper for a complete description.

Traffic Disturbances (D)

Traffic disturbances are the observation of back-propagating stop-and-go waves, which occur regularly in traffic and can be visualised by spatiotemporal speed maps. Disturbances usually emerge from a bottleneck where approaching vehicles try to synchronise with slow traffic therein. These disturbances can propagate further upstream and form wide-moving jams (WMJ). Krishnakumari et al. (2017) proposed and successfully applied the Active Shape Model (Cootes et al. 1995) to identify WMJs in congestion patterns. The Active Shape Model technique describes a shape by utilizing a mean shape and its variations derived from a set of similar training shapes. Consequently, when presented with a new shape, the fitting error of the shape model to this form can be employed for shape identification or classification. To obtain these shapes, pattern images are segmented using the Watershed transformation (Nguyen et al. 2019) into different traffic state regions. The boundaries of these regions are identified and clustered by the Active Shape Model to detect traffic disturbances. We refer to the original paper (Nguyen et al. 2019) for further details.

Homogeneous Congestion (H)

Homogeneous congestion represents the spreading of congested traffic over space and time with consistently low vehicular speeds. They are normally associated with extreme demand or accidents. The regions associated with homogeneous congestion are referred to as Demand–Supply

[1] Please contact the authors for a guest account.

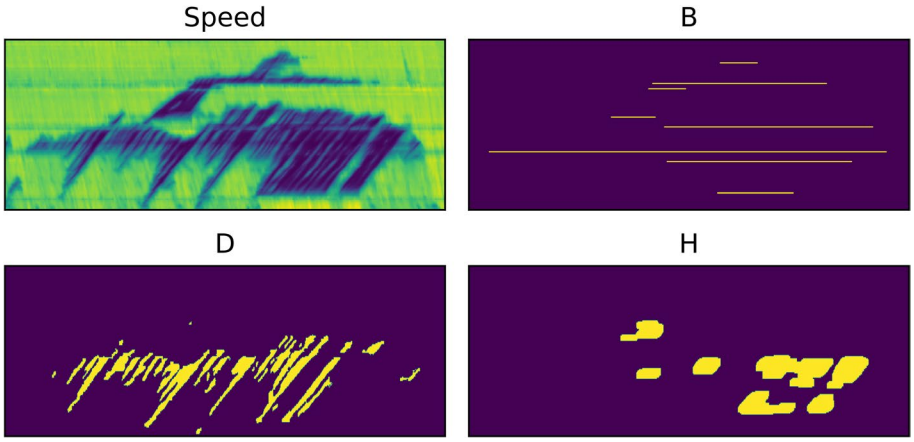


Fig. 2 An example of feature extraction

elements in Nguyen et al. (2019). This paper uses texture analysis to identify homogeneous congestion. Haralick et al. (1973) proposed to derive various texture features of an image using a grey-level co-occurrence matrix (GLCM). This method calculates different statistics to quantify the texture characteristics of the related image. Each number in the GLCM shows how frequently the related pair of intensities is present in the related image for a pre-defined (two-dimensional) offset. Some widely used features are energy, contrast, homogeneity, and entropy. A preliminary analysis suggests that the energy feature is the most promising for identifying homogeneous regions. Energy is a measure of the homogeneity of an image, defined by:

$$\text{Energy} = \sum_{i=1}^{N}\sum_{j=1}^{N} p_d(i,j)^2 \tag{1}$$

The number of grey levels in a homogeneous region is expected to be low, shifting the whole distribution to a small group of $p_d(i,j)$ ($p_d(i,j)$ represents the frequency of having the co-occurrence of intensities i and j at a certain distance d). The more homogeneous a region is, the higher the energy feature gets. Thus, homogeneous congestion regions can be effectively identified.

Figure 2 presents an example of feature extraction. The top left plot shows the original speed map, a complex congestion pattern. The results show that our algorithm successfully identifies bottlenecks, homogeneous congestion regions, and stop-and-go disturbances. After recognizing these 3 fundamental traffic primitives, we can formulate high-level causal relation graphs.

It is useful for readers to note that there exist other methods that can also effectively extract desired features and traffic phenomena, as discussed in the introduction. The methods above are chosen mainly based on our previous research. Another point is that deep learning represents a potent method for identifying these traffic features. However, the application of supervised learning in this context is limited by the need for labelled data, and currently, there are no annotated datasets or pre-trained models available for traffic pattern recognition, to our knowledge. Our proposed method can also serve as an automated congestion pattern annotation pipeline for developing further deep-learning-based models.

Causal Relation-Graph Formulation

To make the high-level representation of traffic congestion patterns interpretable and conform to domain knowledge, extracted traffic primitives cannot be simply concatenated but must be structured by rules. In this study, we propose to use the so-called *causal relation-graph* to organize a set of traffic primitives into a directed acyclic graph causal representation. The key intuition is interpreting "Which bottleneck causes which type of congested region?" and "How does one bottleneck influence others by congestion propagation?"

Specifically, one node (or vertex) in our relation graph represents a traffic primitive. A node furthermore contains attributes describing the corresponding traffic primitive. In this study, the main attribute of a node is *type* (B, D, or H) and *size*, in either absolute form [km × hour] or relative form [proportion % of overall congested area]. The size is calculated from the area within the contour of a traffic primitive on the space-time map. A directed link (or edge) represents a spatiotemporal relation between primitives. This relation indicates a plausible *causality* if the starting point t, x of one primitive is associated with another primitive. For example, to represent many disturbances emerging from a single bottleneck, the corresponding relation graph has an edge from the related bottleneck node to the related disturbance node, with the edge weight indicating the number of disturbances. This example is shown in the Fig. 3. This relation graph has a tree-like structure. For convenience, we define that the

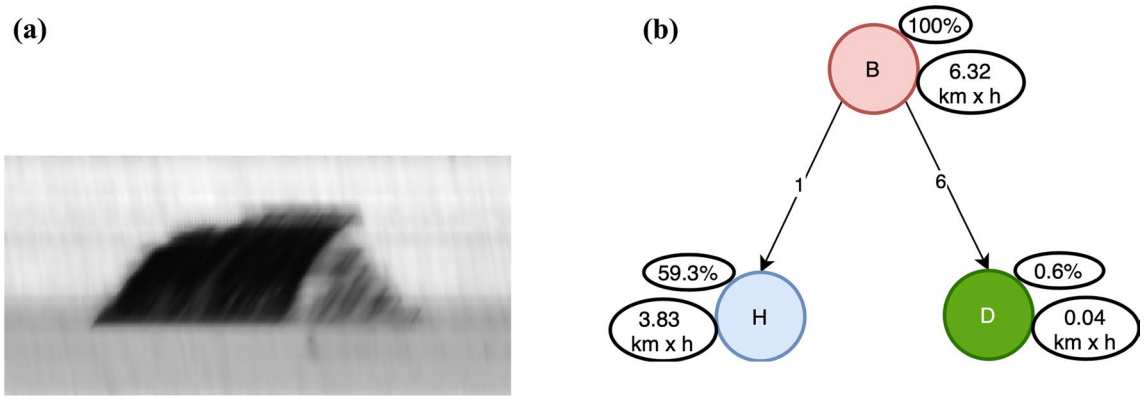


Fig. 3 An example of relation graph: **a** a pattern of congested traffic at a bottleneck which causes 1 heavily homogeneous congestion and later 6 small-scale disturbances, **b** its relation graph proposed by our method. This method can also be applied to multiple bottlenecks and complex congestion patterns

left branch (leaf) happens earlier if one bottleneck causes a sequence of congestion of different types.

A formal definition of this causal relation graph is described as follows:

Definition 1 The causal relation graph representing a congestion pattern is an attributed, directed graph $G = (E, V, A)$. Descriptions of these sets are as follows.

$\mathbf{V} = \{v | v \text{ is a traffic primitive}\}$

$\mathbf{E} = \{(v_i, v_j) | v_i \text{ (possibly) triggers } v_j\}$

$\mathbf{A} = (\tau, s^a, s^p, w)$ attribute set

$\tau : \mathbf{V} \rightarrow \{B - \text{bottleneck}, D - \text{disturbance}, H - \text{homogeneous congestion}\}$

$s^a : \mathbf{V} \rightarrow \mathbf{R}$ absolute size of a node

$s^p : \mathbf{V} \rightarrow \mathbf{R}$ relative size, i.e. the proportion (%), of a node

$w : \mathbf{E} \rightarrow \mathbf{R}$ number of connection instances represented by the edge

Figure 3 illustrates the principle by showing the traffic pattern (left) and the resulting relation graph (right). The pattern shows an example of traffic congestion patterns at a bottleneck which is likely related to an incident. At the onset, traffic is heavily homogeneously congested. After some time, a few (minor) disturbances emerge before traffic regains free-flow states. The corresponding relation graph is constructed by identifying three main elements in this pattern. These include bottlenecks—B node, homogeneous congestion node—H node, and disturbances—D node. Edges are associated with w. Specifically, the link (B–H) has a weight of 1 to represent 1 homogeneous region as shown in the pattern, whilst the link (B–D) has a weight of 6 that shows the number of disturbances detected. Furthermore, each node consists of attributes, namely absolute size and proportion.

So far, we have explained how to construct high-level causal relation graphs from congestion patterns in one-dimensional highway roads. Compared to feature vector representations, this relation-graph-based method is transparent and interpretable. The evolution of congestion is represented by an acyclic causal graph describing how preceding traffic phenomena influence the following.

Adapting the proposed representation method to road networks requires significant modifications. The network structure introduces complex spatial dependencies in congestion patterns due to interactions between road links (Ermagun and Levinson 2019). For instance, road networks that include cycles could lead to cyclic congestion patterns. This happens when congestion at one bottleneck circulates back to the same point via these cycles. Therefore, the relation graph is not cascading but cyclic. Therefore, the structure of the road network must be integrated into the relation graph. Further research is needed to effectively represent road networks (Fafoutellis and Vlahogianni 2023) and incorporate these complex features into the relation graphs. In this study, we only consider congestion patterns along 1D road stretches.

Customizable Similarity Measurement

The previous section shows how to construct an abstract but interpretable causal relation graph for congestion patterns. This section describes the second key module: how similarities between congestion patterns are measured based on those relation graphs, and how to make the similarity measurement customizable. Overall, a similarity function using adopted inexact graph matching is proposed. This function reflects several key aspects of a pattern, including the structure similarity and the proportions and frequencies of traffic primitives. A set of control parameters is made explicit as input to let users customize the desired similarity measurements.

Graph Matching

By representing congestion patterns using relation graphs, the measurement of similarities is transformed into graph similarity or the so-called graph matching problem. There are two main categories of methods, namely *exact* graph matching and *inexact* graph matching (also known as error-tolerant graph matching) (Conte et al. 2004; Foggia et al. 2014; Riesen 2015; Emmert-Streib et al. 2016). Exact graph matching strictly compares two graphs by nodes or edges. This type of method is mainly used to match identical graphs. Meanwhile, inexact methods are more flexible and allow differences in node/edge/subgraph mappings. In principle, these discrepancies are tolerated with a penalty. This property makes inexact graph-matching methods more practical in real problems. It also gives users some room for flexibility, which is key for customizable graph matching. In our application, we use an inexact matching method.

Graph matching is formulated as an optimisation problem between 2 graphs A and B. The cost is generally defined by Eq. (2) (rewritten from Foggia et al. (2014)):

$$
\begin{aligned}
C(f) = & \sum_{\substack{v \in V_A \\ f(v) \neq \varepsilon}} C_R^N(v, f(v)) + \sum_{\substack{v \in V_A \\ f(v) = \varepsilon}} C_D^N(v) + \sum_{\substack{v' \in V_B \\ f^{-1}(v') = \varepsilon}} C_D^N(v') \\
& + \sum_{\substack{e = (v_1, v_2) \in E_A \\ e' = (f(v_1), f(v_2)) \in E_B}} C_R^E(e, e') + \sum_{\substack{e = (v_1, v_2) \in E_A \\ e' = (f(v_1), f(v_2)) \notin E_B}} C_D^E(e) \\
& + \sum_{\substack{e' = (v_1', v_2') \in E_B \\ (f^{-1}(v_1'), f^{-1}(v_2')) \notin E_A}} C_D^E(e')
\end{aligned}
\tag{2}
$$

Note that, in inexact mapping, some nodes or edges of one graph might not have matches with the other graph. To formally describe this, a special *null node* ε is introduced. Accordingly, the mapping f is annotated as $f : V_A \mapsto V_B \cup \{\varepsilon\}$. It is injective for nodes in V_A that are not mapped to ε. For such nodes, the cost is called replacement cost C_R^N. Mapping a node to ε is reasonably seen as the deletion of that node, and the related cost is called deletion cost C_D^N. Besides, edges are also needed to be mapped. Two similar types of mapping, i.e. replacement and deletion, are relevant to edge mapping and are evaluated by the cost functions C_R^E, C_D^E, respectively. Note that these individual cost functions are specialised, which means their definitions depend greatly on specific applications.

Various approaches have been proposed for graph matching by reformulating an optimisation problem on the cost function $C(f)$ such as graph edit distance (Bunke 1997; Gao et al. 2010), graph kernels (Gärtner et al. 2003), iterative methods (Blondel et al. 2004; Zager and Verghese 2008). We refer to Foggia et al. (2014) and Emmert-Streib et al. (2016) for an in-depth survey of these approaches. Our work is motivated by the iterative approach. In principle, *similarities between nodes consider not only the two nodes but also their neighbour nodes*. Hence, this approach, to some extent, combines notations of different individual cost functions (Eq. (2)) into one similarity function.

We propose a two-phase algorithm for measuring the similarity of two congestion patterns based on their relation graphs. Firstly, similarities of all possible pairs of nodes between two graphs are calculated. Secondly, the total similarity score of mapping all available nodes is optimised. The obtained score represents how similar the two patterns are. The following subsections describe these two terms in detail.

Phase 1: Nodes Similarity

The similarity between two nodes (or source nodes) is measured in a recursive way as motivated by Zager and Verghese (2008). Specifically, the similarity of subsequent nodes recursively contributes to the similarity score of their source nodes. Unlike in the initial paper where scores from

Table 1 Control parameters for customising similarity measurement between relation-graphs

Parameter	Description
θ_s	Penalise size difference
θ_g	Regulate the trade-off between node size match (maximised when $\theta_g = 0$) and node type match (maximised when $\theta_g = 1$)
θ_d	Penalise node type difference, therefore, regulate the tolerance of having unmatched nodes
θ_w	Penalise the differences in frequency attribute: whether to focus on overall structure or details
θ_i	Regulate the contribution of subsequent-node similarities to the matching of two source nodes

all possible pairs of nodes are accumulated, our proposed method only considers those from the best mapping between subsequent nodes.

The overall similarity score between two nodes, $n_A \in V_A, n_B \in V_B$ from G_A, G_B respectively, is formulated as Eq. (3). The first part of the right-hand side of an equation, S_0, measures the similarity intrinsically based on their attributes (regardless of their neighbour nodes). The second part represents the accumulation of similarities from their subsequent nodes. Here, the parameter θ_i regulates how much of subsequent nodes' similarity attributes to the similarity of two source nodes. Note that the contribution of subsequent node similarities is, to some extent, equivalent to the similarity of matching corresponding links (which is related to function C_R^E in Eq. 2):

$$\begin{aligned} S(n_A, n_B) &= S_0(n_A, n_B) \\ &+ \theta_i \times \min[S_0(n_A, n_B), \arg\max_{f: C_A \to C_B} \sum_{c_i^A \in C_A} S(c_i^A, f(c_i^A))] \end{aligned} \tag{3}$$

where C_A, C_B represents two sets of subsequent nodes of n_A, n_B, respectively.

Similar to the overall cost defined in Eq. (2), the base similarity S_0 captures several possibilities of node matching, which depend on whether both nodes are in the original graphs. Accordingly, two similar evaluations need to be defined, namely the so-called *replacement*—$S_R(n_A, n_B)$ and *deletion*—$C_D(n)$. Eq (4) summarises these cases.

$$S_0(n_A, n_B) = \begin{cases} S_R(n_A, n_B), & \text{if } n_A \neq \varepsilon, n_B \neq \varepsilon \\ -C_D(n_A), & \text{if } n_B = \varepsilon \\ -C_D(n_B), & \text{if } n_A = \varepsilon \end{cases} \tag{4}$$

There are two cases when matching two non-null nodes regarding whether they represent the same primitive type. If these nodes are different types, their mapping is equivalent to two deletion operations (see Eq (5)).

$$S_R(n_A, n_B) = \begin{cases} M(n_A, n_B), & \text{if } \tau(n_A) = \tau(n_B) \\ -C_D(n_A) - C_D(n_B), & \text{if } \tau(n_A) \neq \tau(n_B) \end{cases} \tag{5}$$

The previous setup leads to defining two basic functions, i.e. $M(n_A, n_B)$ and $C_D(n)$. The choices of these functions are application-specific. In our proposed framework, we formulate these functions concerning selected attributes associated with nodes and edges in relation graphs. Also, these functions are parameterised by utilising certain parameters. The objective is to inject different perspectives when looking for similar characteristics from congestion patterns.

The proposed function for measuring similarity between two commonly labelled nodes accounts for both the resemblance between their attributes and the importance of their difference. For that, the designed function includes both their overlapping size and the size of the referenced node. The former acts as a proxy for the similarity of two nodes. The latter is used to compensate for the difference (if any) between two nodes. The first node is selected as a referenced node in our setup. The parameter θ_g regulates the scales of these two terms (see Eq. 6).

The detailed similarity between two nodes is measured based on the overlapping size. Note that a different function is possible when different properties are used for node attributes. On the other hand, the unmatched size is also taken into account as this assists in ranking the closeness of different pairs of nodes. In particular, a logistic function is formulated to translate the size difference (in terms of proportions to the total size) to a number (i.e. weight) that scales the overall similarity. The contribution of this difference is regulated by the parameter $\theta_s \geq 0$ (see Eq. (6)). In addition, the difference in the occurrences (w) of the two nodes is also dealt with. A 'virtual node' n_E, with relevant features, a (see Equation 13) and w, is created as shown in Eq. (11). The underlying idea is to apply deletion cost $C_D(n_E)$ to the occurrence difference when matching two nodes. Parameter $\theta_w \geq 0$ regulates the tolerance of this difference:

$$\begin{aligned} M(n_A, n_B) =& (1 - \theta_g) * \left[2 \times w_{min} \times a_{min} \times \mathcal{L}_{\beta_1, \beta_0}\left(\theta_s, \frac{\Delta a}{\sum a}\right)\right. \\ &\left. - C_D(n_E)\right] + \theta_g \times 2 \times w(n_A) \times a(n_A) \end{aligned} \tag{6}$$

where,

$$\text{Common size} \quad a_{min} = min\big(a(n_A), a(n_B)\big) \tag{7}$$

$$\text{Size difference} \quad \Delta a = |a(n_A) - a(n_B)| \tag{8}$$

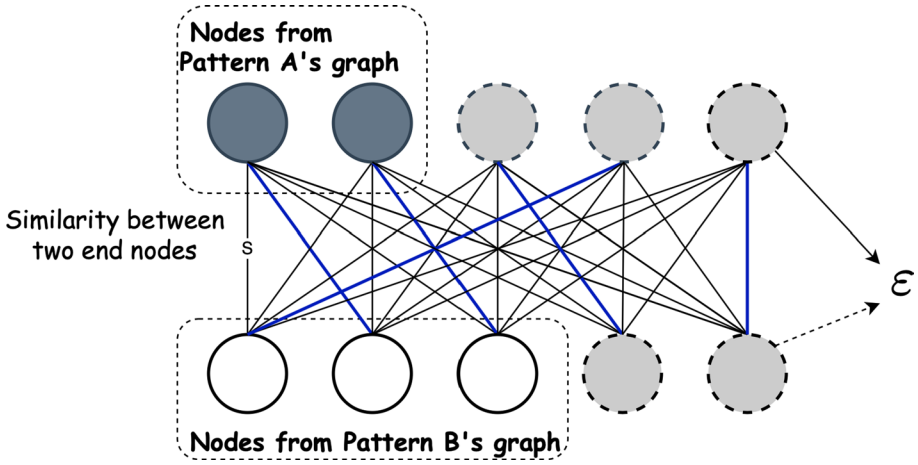


Fig. 4 An illustration of how to formulate the pattern matching as an assignment problem between their node sets. Edges' weights are the similarities between the corresponding end nodes using Eq. (3). A feasible assignment is highlighted in blue colour, in which one node is exactly matched to another node

$$\text{Total size} \quad \sum a = a(n_A) + a(n_B) \tag{9}$$

$$\text{Logistic function} \quad \mathcal{L}_{\beta_1,\beta_0}(\theta, x) = 1 - \frac{1}{1 + e^{\beta_1(\theta x)+\beta_0}} \tag{10}$$

$$\text{Occurrence-difference node} \quad n_E = \begin{cases} a = \begin{cases} a(n_A), \text{ if } w(n_A) > w(n_B) \\ a(n_B), \text{ if otherwise} \end{cases} \\ w = f_{\beta_1,\beta_0}(\theta_w, \Delta w) \end{cases} \tag{11}$$

$$\text{Occurrence difference} \quad \Delta w = |w(n_A) - w(n_B)| \tag{12}$$

$$\text{Size selection:} \quad a(n) = \begin{cases} s^a(n), & \text{for absolute size} \\ s^p(n), & \text{for proportion} \end{cases} \tag{13}$$

As overlapping sizes are used for attributing commonly labelled nodes, the cost of deleting a node can be justified by its size. A parameter θ_d is introduced here to regulate how much penalty is applied for not finding a match for a node. Equation (14) defines this cost:

$$C_D(n) = \theta_t \times a(n) \tag{14}$$

Table 1 summarises all the control parameters and their meanings in customizing a similarity measurement between any pair of nodes.

Phase 2: Nodes Mapping

Given two relation graphs that represent two congestion patterns, the previous section shows how to measure the similarity between any pairs of nodes therein. This section describes how to come up with a similarity score at the pattern level.

To evaluate how the two patterns match, we formulate the problem as an assignment problem which finds the so-called perfect matching between nodes from the two graphs. That assignment maximises the total scores from all pairs of matched nodes under the condition that one node is matched with exactly another one. This perfect matching (once found) is considered the best mapping between the two source nodes. The corresponding total score then indicates the similarity between the two patterns. An illustration of our assignment problem is depicted in Fig. 4. A complete bipartite graph is constructed to show all possible mapping of nodes from two graphs. The weight of each edge is associated with the similarity score of corresponding nodes. The solution of pattern mapping is the perfect matching with the maximum total edge' weights. Equation (15) formulates this assignment approach in mathematical terms.

$$S(p_A, p_B) = \arg\max_{f:\Omega_A \to \Omega_B} \sum_{n \in \Omega_A} S(n, f(n))$$

where,

$$\begin{aligned} \Omega_A &= V_A \cup \{\varepsilon, \ldots, \varepsilon\} \\ \Omega_B &= V_B \cup \{\varepsilon, \ldots, \varepsilon\} \\ |\Omega_A| &= |\Omega_B| = |V_A| + |V_B| \end{aligned} \tag{15}$$

The assignment problem is solved by applying the well-known Hungarian algorithm (also known as the Kuhn-Munkes algorithm), which was developed by Kuhn (1955). It has polynomial complexity, in particular, $\mathcal{O}(n^3)$.

In summary, this section describes how to match two relation graphs based on the customized graph similarities. 5 parameters (θ) are modifiable for users to control the desired similarity functions from different perspectives, such as congestion size differences, overall congestion pattern structures and so on. Next, we will carry out experiments on a real-world dataset to evaluate the proposed method.

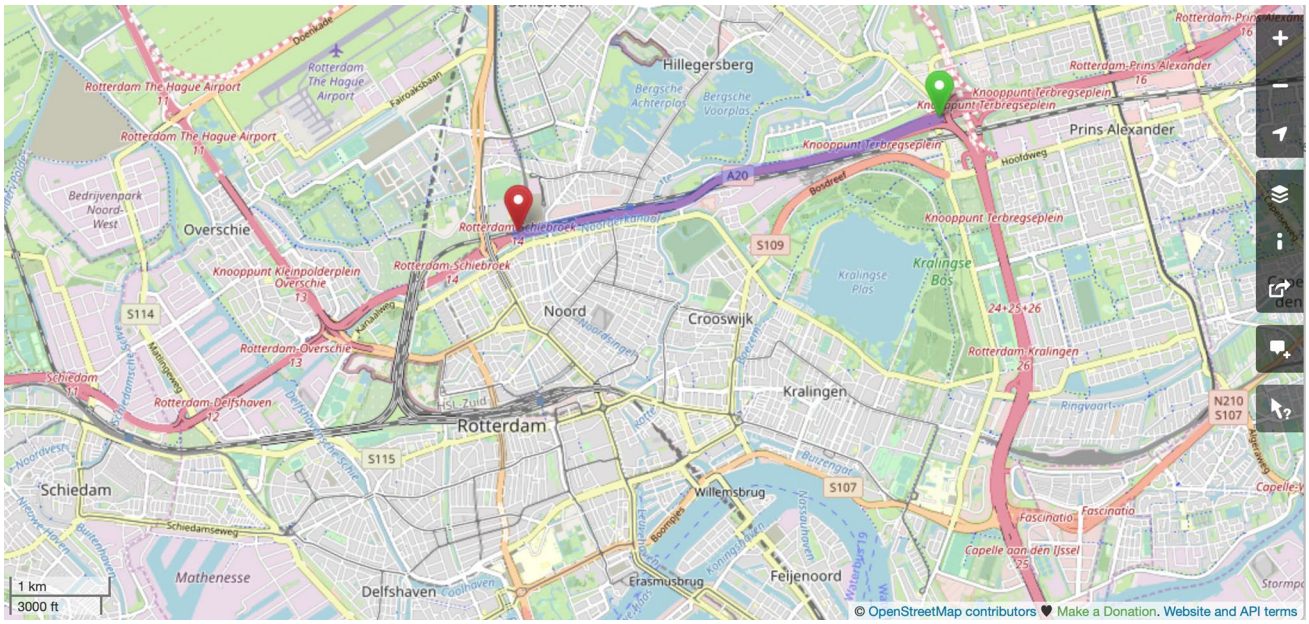


Fig. 5 A broad view of the selected corridor in the experiment. The image is taken from the Open Street Map (OSM)

Experiments

In this section, we demonstrate the proposed framework to retrieve similar patterns from a collection of traffic congestion patterns. The analysis includes three aspects. First, some exemplary queries are conducted, and their performances are investigated concerning the corresponding obtained patterns. Second, we discuss the impacts of the control parameters (in Table 1) for reflecting different perspectives on similarities between patterns. Third, we consider the computational complexity and its implication in applying it to large datasets of the proposed method.

Data and Parameter Settings

To evaluate the proposed method, a corridor on the ring of Rotterdam is selected, which is one of the busiest highways in the Netherlands. The total length of the selected road is approximately 19 km. Figure 5 shows a broad view of one stretch of the road. This segment is around 4 km long and comprises several active bottlenecks next to the Rotterdam central station. These bottlenecks and downstream bottlenecks have caused much recurrent traffic congestion, therefore, it is a suitable choice for evaluating our proposed framework.

Speed data are provided by the National Data Warehouse (NDW) (ndw.nl), in which each measurement is a one-minute aggregation of average speed (of all lanes at a location) surpassing the related induction-loop detector's implemented location. In the raw data, the space interval between two adjacent loop detectors is not uniform, ranging from 60 to hundreds of meters. To have a better view of resulting traffic, we apply the ASM method (Adaptive Smoothing Method) (Treiber and Helbing 2002; Schreiter et al. 2010) to map speeds into uniform grids at finer resolutions both spatially and temporally, namely 100 ms by 30 s. We have processed data from the entire year of 2018 to obtain 778 patterns, which constitute the collection of traffic congestion patterns for evaluating our proposed method.

For the similarity measurement, the settings for all parameters are given in Table 2 (as the first example). By setting θ_t, θ_s, and θ_w to 1, the total differences in type, size, and frequency, respectively, are fed to the logistic function to measure related penalties. As θ_i is set to 1, similarities from subsequent nodes are accumulated to the corresponding president nodes. This, to some extent, considers pattern structure. Therefore, we set θ_g to 0 to simplify the base similarity function. This leads to a full assessment of related attributes when matching two nodes. Chosen values of β set the changing point of the corresponding logistic function at the middle of input ranges. Note that there are no strict

Table 2 Control parameter settings in the conducted experiment

Parameter	Value
θ_s	1
θ_g	0
θ_t	1
θ_w	1
θ_i	1
logistics $\mathcal{L}$	
(β_1, β_0)	$(10, -5)$

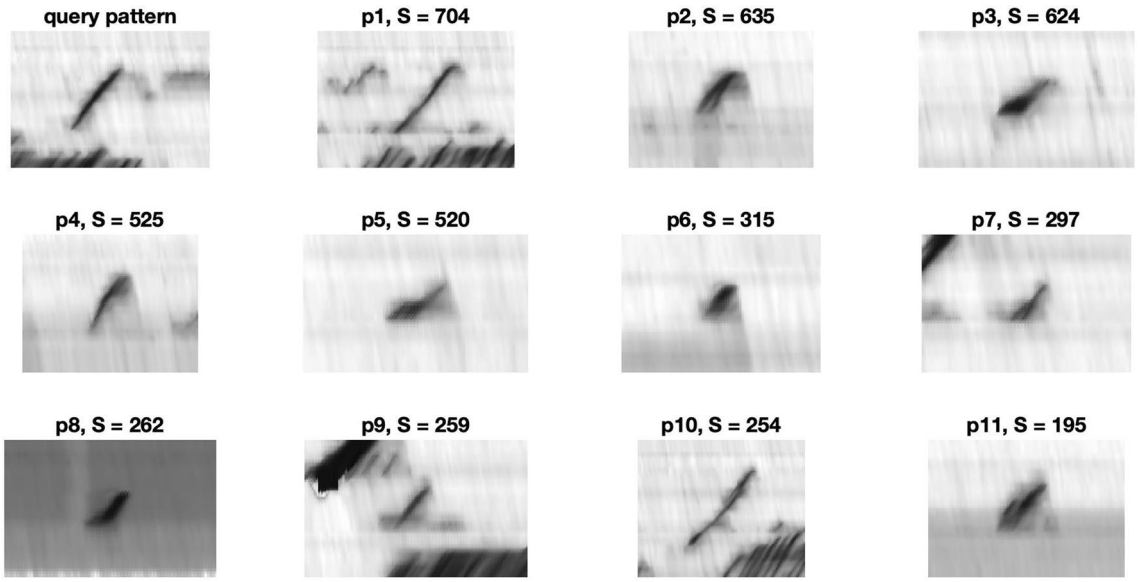


Fig. 6 The 11 most similar patterns returned from searching for a moving disturbance (shown in the top-left pattern). Patterns are shown in the same resolution, hence, their size differences can be relatively shown. Note that, regions of congestion at the edges of some patterns should be ignored because they are the results of cropping out the patterns, i.e. they are not included as (main) content of the patterns. Besides, similarity scores are given as *S* for each of the patterns

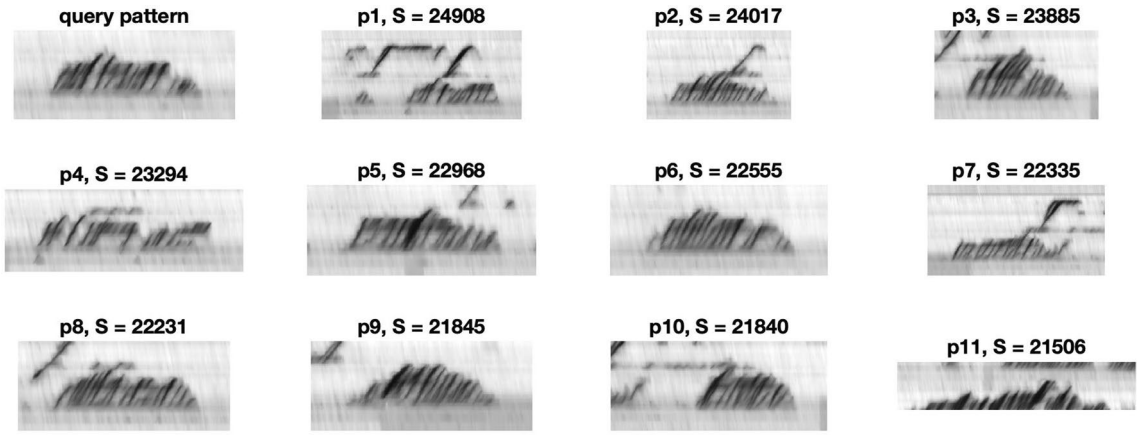


Fig. 7 Retrieval results of stop-and-go congestion

regulations in selecting these parameters. The presented settings are one of many possibilities.

Retrieval Result Examples

To demonstrate the feasibility of the proposed relation graph in the retrieval application, we analyse some example queries, namely for single disturbance, stop-and-go congestion, homogeneous congestion and a mix of these. These are typical patterns of congestion that are commonly observed in traffic data (Helbing et al. 2009; Nguyen et al. 2016, 2019; Krishnakumari et al. 2017).

Single Disturbance Retrieval

Figure 6 shows an example of retrieving patterns representing a single disturbance. The implemented framework successfully returned patterns representing small disturbances as indicated in the query pattern. Regarding the order of these patterns, some might seem more similar than those in higher ranks. For example, pattern p4 seems more resembling the query pattern than the above two patterns (in the order list). The reason is that by choosing areas as an attribute, we have reduced two dimensions, i.e. spatial and temporal, down to only one. Therefore, introducing propagating

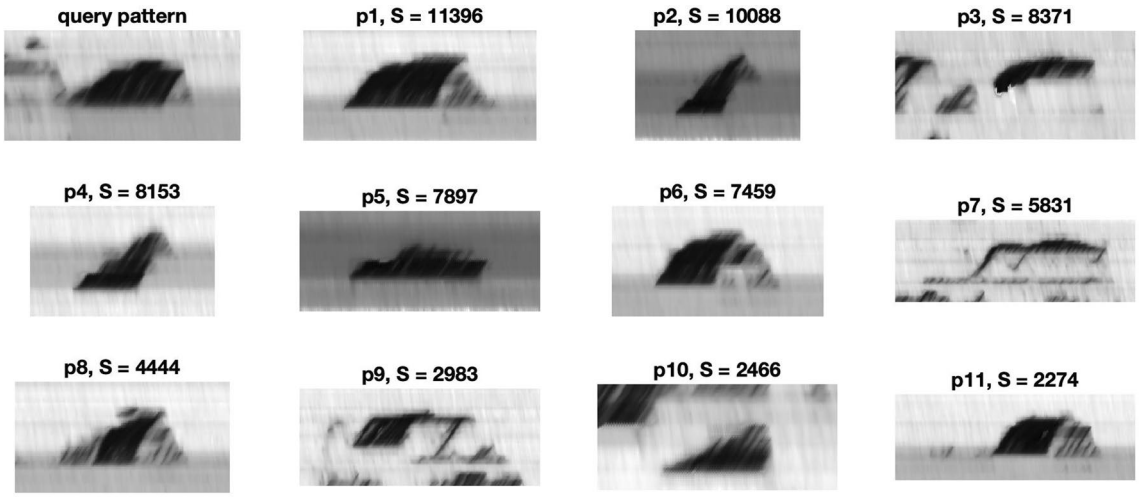


Fig. 8 Retrieval results of homogeneous congestion

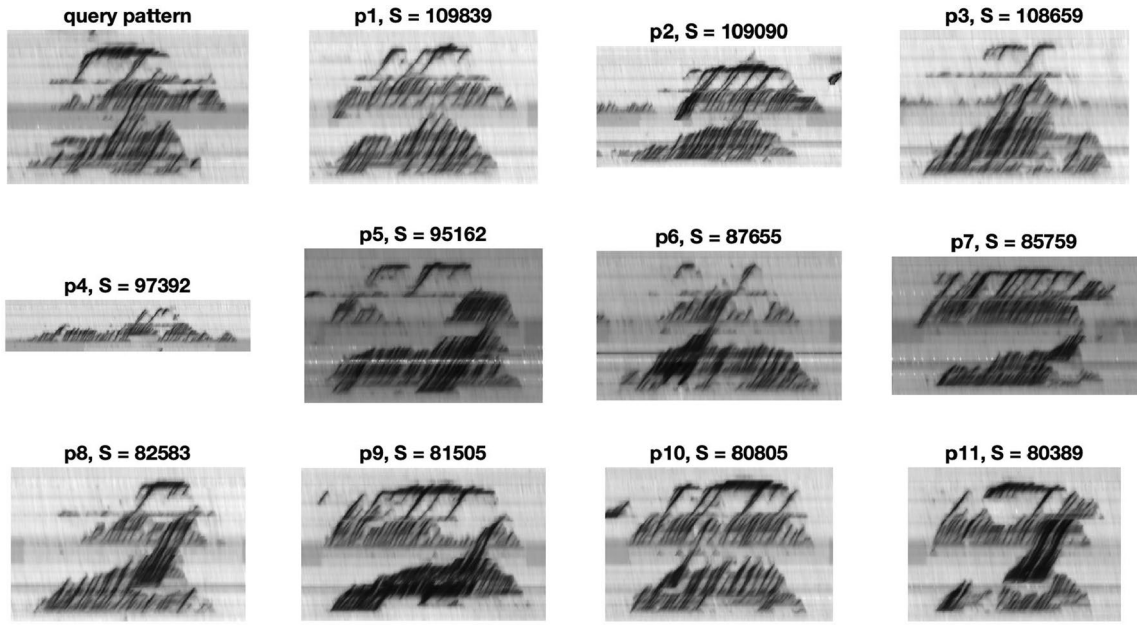


Fig. 9 Retrieval results of *meta* congestion

lengths as attributes for disturbance nodes could fine-tune the results further.

Stop-and-Go Congestion Retrieval

Stop-and-go traffic waves are another common type of congestion where multiple disturbances occur over time. An example of retrieving such patterns is shown in Fig. 7. In the query pattern, a bottleneck is activated, from which many disturbances emerge. All obtained patterns represent the same traffic phenomena. By detecting both the primary bottlenecks and probably the minor upstream secondary bottleneck, along with multiple disturbances, the obtained relation graphs are effective for locating patterns with the same topology.

Homogeneous Congestion Retrieval

An example of retrieving homogeneous congestion is illustrated in Fig. 8. The given pattern represents significantly slow traffic upstream of a bottleneck (probably due to incidents like accidents). Hence, the two most important

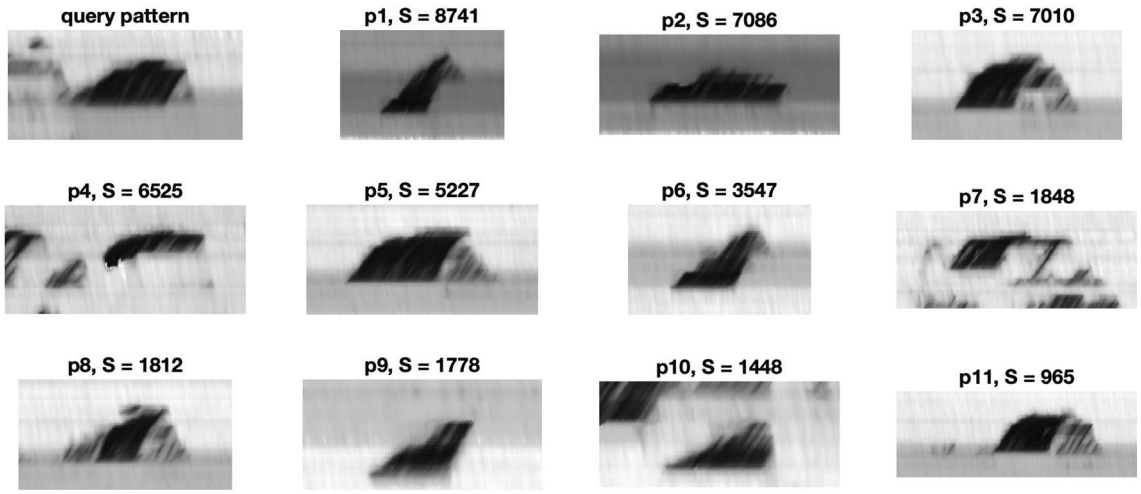


Fig. 10 Another retrieval result of the homogeneous congestion in Fig. 8) with different parameter $\theta_s = 2$

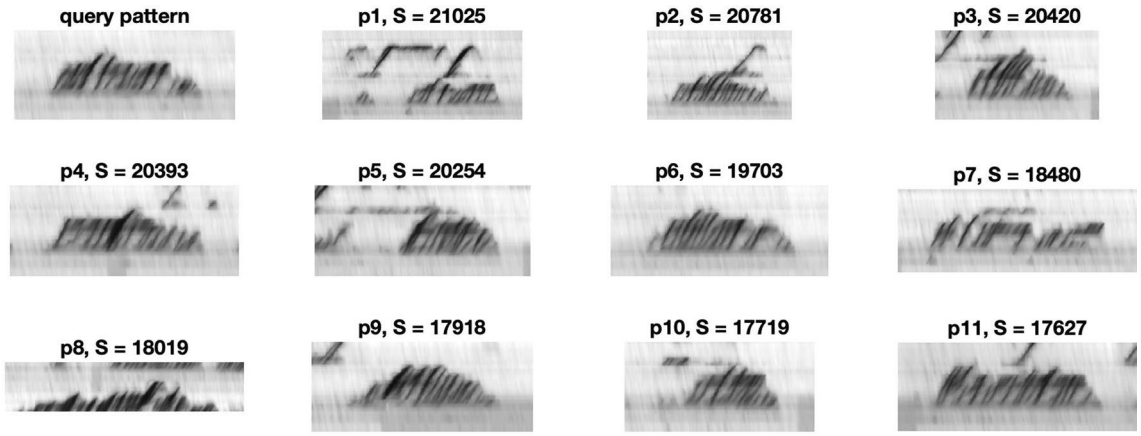


Fig. 11 Another retrieval result of the stop-and-go congestion in Fig. 7 by increasing θ_w to 3

components of the corresponding relation graph for this pattern are a bottleneck node and a homogeneity node. Overall, the obtained patterns do represent the main phenomenon. Note that the shapes of homogeneous areas in obtained patterns are not necessarily identical to those in the query pattern because the chosen attribute includes only sizes.

Complex Congested Traffic Retrieval

Figure 9 illustrates an attempt to retrieve large-scale congestion patterns. The query pattern consists of various types of traffic jams, including disturbances that occur fairly frequently, multiple bottleneck activation, and a homogeneous congested area. Many obtained patterns can cope with these complications in the input pattern, meaning they have different bottlenecks that cause dense stop-and-go traffic. Some of them show homogeneous regions. Regarding the overall structure, several patterns (for instance, p1, p2 or p3) represent two clusters of disturbances that are potentially due to the activation of two primary bottlenecks.

Large-scale complex congestion patterns are rare in the dataset. For example, among the 778 patterns, only 29 patterns (around 3.5%) have congestion areas that are larger than 50,000 [km × min]. These patterns typically span large areas and coincide with morning or evening peak traffic hours, limiting their occurrence to at most twice a day. In contrast, smaller and more isolated congestion patterns are observed more frequently.

Impacts of Control Parameters

In this subsection, we analyse how control parameters change similarity scores and, hence, alter the ranks of

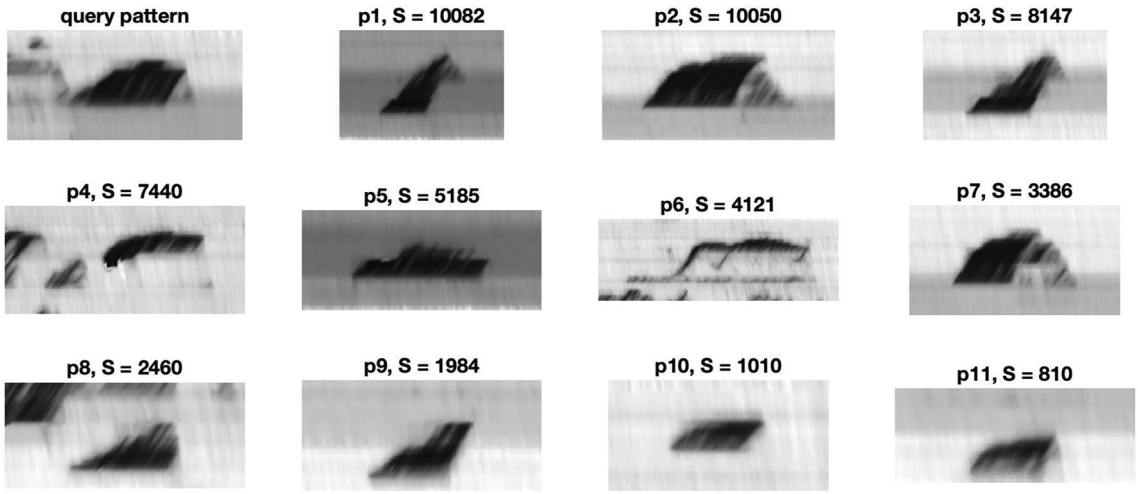


Fig. 12 Another retrieval result of the homogeneous congestion in Fig. 8 by increasing the unmatch penalty θ_d to 2

obtained patterns. This is relevant for customizing retrieval results.

Size Penalty θ_s

θ_s penalises size differences between two matched nodes. Therefore, it regulates the importance of searching for nodes of similar types. To demonstrate this, we modify $\theta_s = 2$ for the retrieval in Fig. 8. This modification enforces a stricter condition on the sizes of matching nodes. The corresponding result is shown in Fig. 10. Overall, the similarity scores of returned patterns decrease. The order of patterns consists of various changes such as the promoting of p1 (from the 2nd to the 1st place) and p5 (from the 1st to the 5th). In addition, new patterns are also moved forward, such as p10.

Weight Penalty θ_w

θ_w controls the frequency of a component's appearances. This is mostly relevant to disturbances in stop-and-go traffic patterns. By increasing or decreasing this parameter, the outcomes are adjusted to be against or in favour of the differences in the frequencies of disturbances. Figure 11 demonstrates the impact of increasing θ_w on the same search made in Fig. 7. Even though there is not much (overall) difference compared to the previous result, this new result shows several changes in the order. The new ranking promotes those patterns with more similar numbers of disturbances as in the example pattern. The overall similarity scores are smaller due to the stricter condition of occurrence frequencies.

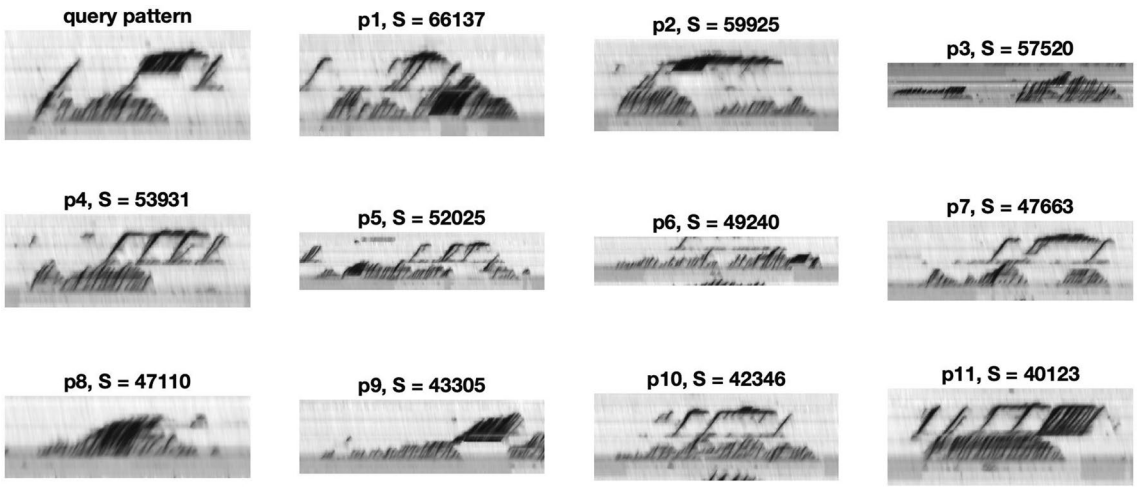
Unmatch Penalty θ_d

There may be unmatched nodes from two relation graphs. How much this decreases the similarity score is regulated by θ_d. By lowering this parameter, users opt for finding the completion of the components in the query pattern, and at the same time tolerate the existence of extra components in the target patterns. Similarly, increasing θ_d aims for the compact of target patterns concerning the given pattern. An example of the effect of increasing θ_d is shown in Fig. 12, which is a modified retrieval of the one in Fig. 8. Since θ_d has a higher value, those patterns with a more compact representation of homogeneous regions, less other extra regions, are advanced in the ranking list.

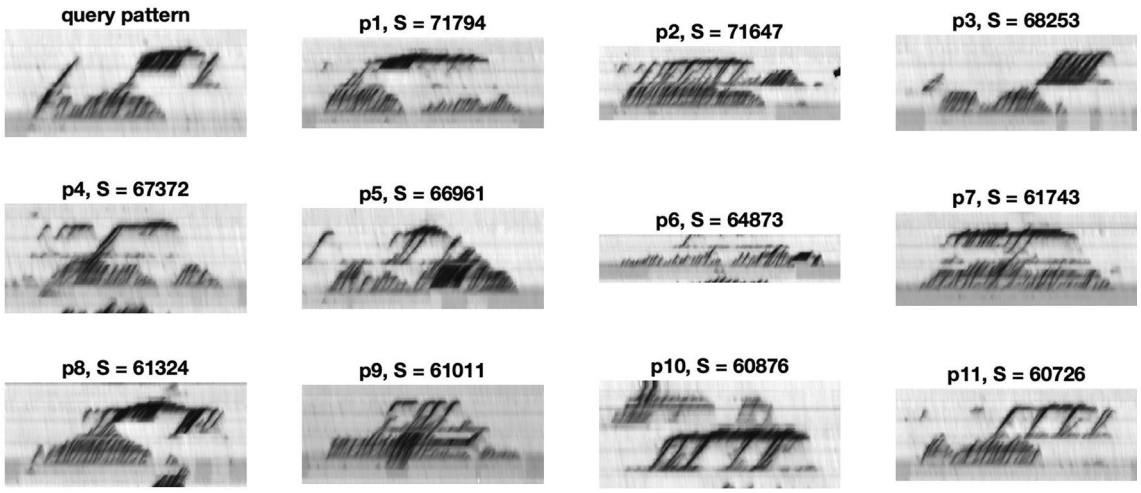
Structural Integrity θ_g

The parameters θ_g, θ_i are designed to promote the matching of pattern structures. A demonstration of their use is illustrated in Fig. 13. Figure 13a shows an example in which both similarities of pairs of matched nodes and their subsequent nodes are relatively important. On the other hand, by setting $\theta_g = 0.7$, the importance of having the same structure becomes higher while that of node similarities is reduced. The obtained patterns in Fig. 13b demonstrate the effect of this change. Differences between components of obtained patterns and those in query patterns are more tolerant. As a result, some good similar patterns are advanced to the top list, e.g. p1, p3, p4, p8. Note that, increasing θ_i leads to low importance levels of node features. This, therefore, can result in patterns that are quite different from the query example despite sharing a common structure.

We have shown that users can customize the data retrieval by tuning the control parameters. To better



(a) $(\theta_s, \theta_g, \theta_d, \theta_w, \theta_i) = (1, 0, 1, 1, 1)$



(b) $(\theta_s, \theta_g, \theta_d, \theta_w, \theta_i) = (1, 0.7, 1, 1, 1)$

Fig. 13 The effect of the structural integrity

explain how to choose these parameters, we provide a generic guidance here. We recommend fixing the size penalty $\theta_s = 1$ unless there are strict requirements on the size of each congested area. If users want to specifically study traffic disturbances, then the weight penalty can be set as $\theta_w = 2$ or 3, otherwise 1. If users are interested in complex large-scale congestion patterns containing all features, choose the structural integrity θ_g between 0.7 and 1, otherwise between 0.1 and 0.3. The unmatch penalty θ_d depends on how much data we have. If the congestion dataset is small and there are not too many similar patterns, choosing a lower value can give more retrieved patterns.

Time Complexity

The processing time in the proposed method is spent mainly on relation-graph construction and graph-similarity measurement. Regarding the former, relation graphs of all congestion patterns in the database are pre-processed and registered in advance. Hence, when retrieving data, only the example pattern needs to be parsed. The processing time depends (almost) linearly on the size of the

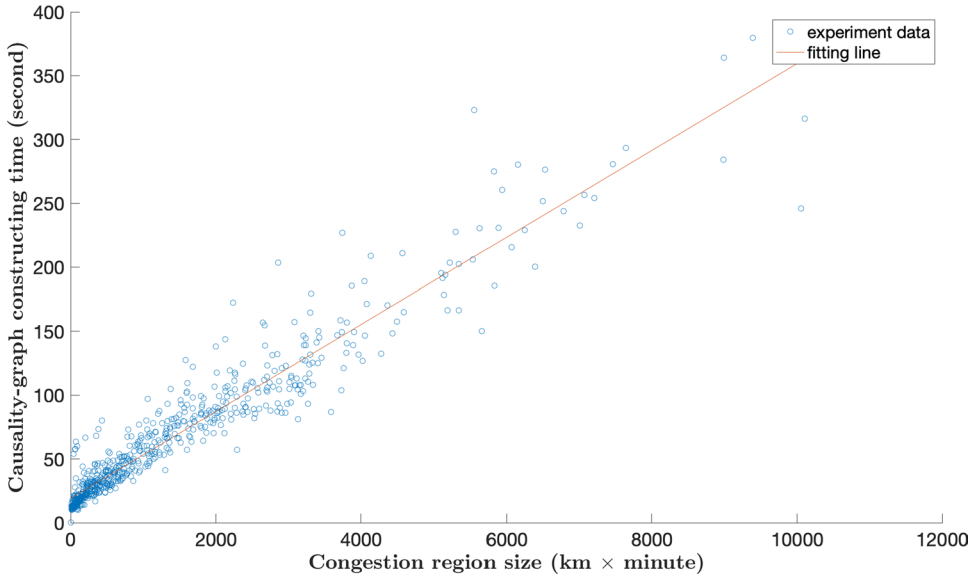


Fig. 14 The constructing time of causality-graphs of all the patterns in the experiment data

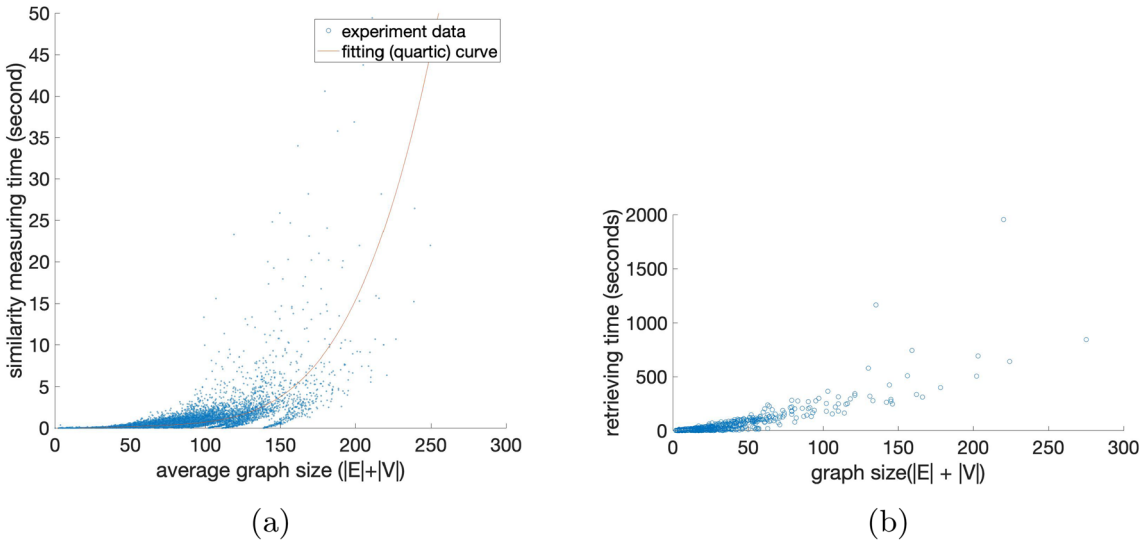


Fig. 15 Computation time of measuring the similarity between two relation-graphs: **a** single pair measurement, **b** retrieval time from the whole data (of 778 congestion patterns)

corresponding congestion region (or pattern) as shown in Fig. 14. In addition, the majority of patterns have sizes of approximately under 1000 (km × minutes) and take around 60 s to build their relation graphs.

Figure 15 illustrates the computation of relation-graph similarity. This includes times for matching every single pair as shown in Fig. 15a and the total retrieving time in the experiment dataset shown in Fig. 15b. It can be expected that the time complexity of the proposed matching method for relation graphs is polynomial w.r.t graph size (measured in the total number of nodes and edges). From a close examination of Fig. 15b, it takes less than one minute to retrieve similar patterns for a pattern of up to 30 nodes plus edges in its relation graph. However, the waiting time can be long for large-scale patterns or a collection of numerous patterns. Therefore, to scale up the proposed method to larger datasets, further improvements are necessary. One approach is to narrow down the search space by some quick pre-processing. For example, as suggested by Fig. 15a, when retrieving small-scale patterns, a (computationally) fast filter can be applied to keep only patterns with small numbers of nodes in their relation graphs.

Conclusion and Perspectives

This paper presents a novel methodological framework for interpretable pattern representation and customizable pattern retrieval of highway traffic congestion data. We demonstrate the efficacy and efficiency of this framework on a large-scale traffic database that spans the entire Dutch freeway road network over several years. Our experiments reveal that the methodology allows for retrieving interpretable patterns tailored to users' customized similarity measurements, by adjusting five control parameters to specify desired pattern types based on a query example. The scalability of this retrieval system is supported by time complexity analysis, enabling expansion to larger datasets. Most importantly, the case study highlights the success of our method in retrieving complex patterns across various scenarios, showing that integrating domain knowledge and causal relation graphs with pattern recognition techniques greatly enhances effectiveness.

Our proposed methodology is closely related to other feature extraction methods, particularly those based on deep learning, yet it stands apart in several ways. Deep learning methods are adept at precisely recognizing fundamental traffic primitives, such as congestion bottlenecks and disturbances, but they require a large, well-annotated dataset for training. In contrast, our training-free framework offers a more feasible option when only raw congestion patterns are available, particularly in the early stages of data collection. As data accumulates, leveraging the expanded dataset to train deep neural networks represents a promising avenue for future research. Additionally, our method has advantages in explainability and customizability for data retrieval. However, as explainable AI emerges, deep-learning-based search approaches might provide equally satisfactory results in terms of customizable information retrieval in the future.

Future research directions to enhance the proposed method include addressing computational time concerns. One approach involves initiating a rough classification of input patterns, effectively narrowing the search space to a single class. Another option is a two-step approach that combines generic feature methods with the proposed approach, using fast-to-compute Euclidean distances to measure similarities between patterns quickly. Additionally, integrating more relevant characteristics into the relation graph, especially traffic demand data that drives the evolution of traffic congestion, could further improve the explainability of pattern representation and refine retrieval outcomes. For example, users can study the diversity of congestion patterns under similar demand settings. This can give insights into effective traffic management.

The proposed system is a powerful toolkit to expedite many studies using traffic congestion data. Here, we give one closely relevant topic: generative AI for knowledge-guided rare congestion pattern generation. The proposed unified representation can be interpreted as a knowledge graph. For those rare congestion patterns based on a similarity measurement defined by users, using generative AI to create similar patterns in batches may ultimately break the barrier of macroscopic traffic data acquisition.

Acknowledgements The authors acknowledge the Dutch National Data Warehouse of Traffic Information (NDW) and the European Union's Horizon 2020 Research and Innovation program (Grant Agreement No. 688082) for sponsoring this study.

Author Contributions T.T.N proposed the overall conceptualization and the methodological framework, wrote most of the algorithm code, developed the software and the web application, prepared figures from Figure 3, and wrote the main manuscript text. G.L. proposed the concept of customizable data retrieval, wrote the Python API to the original code, prepared and revised figures 1-2, and wrote and revised the manuscript text. S.C.C. provided supervision and resources and revised the manuscript. H.v.L. provided supervision and resources and revised the manuscript.

Funding This research is funded by the European Union's Horizon 2020 Research and Innovation program (Grant Agreement No. 688082) with the Dutch National Data Warehouse of Traffic Information (NDW).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interst The authors declare no competing interests.

Ethical Approval Not applicable.

References

Andrews Sobral LO, Schnitman L, De Souza F (2013) Highway traffic congestion classification using holistic properties. In: 10th IASTED international conference on signal processing, pattern recognition and applications

Bay H, Ess A, Tuytelaars T et al (2008) Speeded-up robust features (surf). Comput Vis Image Underst 110(3):346–359

Blondel VD, Gajardo A, Heymans M et al (2004) A measure of similarity between graph vertices: applications to synonym extraction and web searching. SIAM Rev 46(4):647–666

Boquet G, Morell A, Serrano J et al (2020) A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection. Transp Res Part C Emerg Technol 115:102622

Bunke H (1997) On a relation between graph edit distance and maximum common subgraph. Pattern Recognit Lett 18(8):689–694

Calvert SC, Van Den Broek TA, van Noort M (2011) Modelling cooperative driving in congestion shockwaves on a freeway network. In: 2011 14th international IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp 614–619

Calvert S, Taale H, Snelder M et al (2018) Improving traffic management through consideration of uncertainty and stochastics in traffic flow. Case Stud Transp Policy 6(1):81–93

Chan TF, Vese LA (2001) Active contours without edges. IEEE Trans Image Process 10(2):266–277

Chen C, Skabardonis A, Varaiya P (2004) Systematic identification of freeway bottlenecks. Transp Res Rec J Transp Res Board 1867:46–52

Conte D, Foggia P, Sansone C et al (2004) Thirty years of graph matching in pattern recognition. Int J Pattern Recognit Artif Intell 18(03):265–298

Cootes TF, Taylor CJ, Cooper DH et al (1995) Active shape models-their training and application. Comput Vis Image Underst 61(1):38–59

Datta R, Joshi D, Li J et al (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv (Csur) 40(2):1–60

Emmert-Streib F, Dehmer M, Shi Y (2016) Fifty years of graph matching, network alignment and network comparison. Inf Sci 346:180–197

Ermagun A, Levinson D (2019) Spatiotemporal short-term traffic forecasting using the network weight matrix and systematic detrending. Transp Res Part C Emerg Technol 104:38–52

Fafoutellis P, Vlahogianni EI (2023) Unlocking the full potential of deep learning in traffic forecasting through road network representations: a critical review. Data Sci Transp 5(3):23

Foggia P, Percannella G, Vento M (2014) Graph matching and learning in pattern recognition in the last 10 years. Int J Pattern Recognit Artif Intell 28(01):1450001

Gao X, Xiao B, Tao D et al (2010) A survey of graph edit distance. Pattern Anal Appl 13(1):113–129

Gärtner T, Flach P, Wrobel S (2003) On graph kernels: hardness results and efficient alternatives. In: Learning theory and kernel machines. Springer, pp 129–143

Haralick RM, Shanmugam K, Dinstein IH (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 6:610–621

Helbing D, Treiber M, Kesting A et al (2009) Theoretical vs empirical classification and prediction of congested traffic states. Eur Phys J B Condens Matter Complex Syst 69(4):583–598

Kim J, Mahmassani HS, Vovsha P et al (2013) Scenario-based approach to analysis of travel time reliability with traffic simulation models. Transp Res Rec 2391(1):56–68

Krishnakumari P, Nguyen T, Heydenrijk-Ottens L et al (2017) Traffic congestion pattern classification using multiclass active shape models. Transp Res Rec 2645(1):94–103

Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res Logist Q 2(1–2):83–97

Li Y, Yu R, Shahabi C et al (2017) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926

Li G, Knoop VL, Van Lint H (2021) Multistep traffic forecasting by dynamic graph convolution: interpretations of real-time spatial correlations. Transp Res Part C Emerg Technol 128:103185

Lopez C, Leclercq L, Krishnakumari P et al (2017) Revealing the day-to-day regularity of urban congestion patterns with 3d speed maps. Sci Rep 7(1):14029

Ma X, Dai Z, He Z et al (2017) Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17(4):818

Nguyen HN, Krishnakumari P, Vu HL et al (2016) Traffic congestion pattern classification using multi-class svm. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp 1059–1064

Nguyen TT, Krishnakumari P, Calvert SC et al (2019) Feature extraction and clustering analysis of highway congestion. Transp Res Part C Emerg Technol 100:238–258

Nguyen TT, Calvert SC, Vu HL et al (2021) An automated detection framework for multiple highway bottleneck activations. IEEE Trans Intell Transp Syst 23(6):5678–5692

Riesen K (2015) Structural pattern recognition with graph edit distance. In: Advances in computer vision and pattern recognition. Springer

Schreiter T, van Lint H, Treiber M et al (2010) Two fast implementations of the adaptive smoothing method used in highway traffic state estimation. In: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp 1202–1208

Treiber M, Helbing D (2002) Reconstructing the spatio-temporal traffic dynamics from stationary detector data. Coop Transp Dyn 1(3):1–3

van de Weg GS, Vu HL, Hegyi A et al (2018) A hierarchical control framework for coordination of intersection signal timings in all traffic regimes. IEEE Trans Intell Transp Syst 20(5):1815–1827

Van Lint J, Hoogendoorn S, van Zuylen HJ (2005) Accurate freeway travel time prediction with state-space neural networks under missing data. Transp Res Part C Emerg Technol 13(5–6):347–369

van Lint H, Nguyen TT, Krishnakumari P et al (2020) Estimating the safety effects of congestion warning systems using carriageway aggregate data. Transp Res Rec 2674(11):278–288

Vlahogianni EI, Karlaftis MG, Golias JC (2005) Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp Res Part C Emerg Technol 13(3):211–234

Wang Y, Papageorgiou M, Messmer A (2006) Renaissance—a unified macroscopic model-based approach to real-time freeway network traffic surveillance. Transp Res Part C Emerg Technol 14(3):190–212

Wang C, Quddus MA, Ison SG (2009) Impact of traffic congestion on road accidents: a spatial analysis of the m25 motorway in England. Accident Anal Prev 41(4):798–808

Wieczorek J, Fernández-Moctezuma RJ, Bertini RL (2010) Techniques for validating an automatic bottleneck detection tool using archived freeway sensor data. Transp Res Rec 2160(1):87–95

Yin X, Wu G, Wei J et al (2021) Deep learning on traffic prediction: methods, analysis, and future directions. IEEE Trans Intell Transp Syst 23(6):4927–4943

Zager LA, Verghese GC (2008) Graph similarity scoring and matching. Appl Math Lett 21(1):86–94

Zhao T, Nie YM, Zhang Y (2014) Extended spectral envelope method for detecting and analyzing traffic oscillations. Transp Res Part B Methodol 61:1–16

Zheng Z, Ahn S, Chen D et al (2011) Applications of wavelet transform for analysis of freeway traffic: bottlenecks, transient traffic, and traffic oscillations. Transp Res Part B Methodol 45(2):372–384

Zhou W, Li H, Tian Q (2017) Recent advance in content-based image retrieval: a literature survey. arXiv preprint arXiv:1706.06064