

Traffic Congestion Pattern Classification Using Multiclass Active Shape Models

Panchamy Krishnakumari, Tin Nguyen, Léonie Heydenrijk-Ottens,
Hai L. Vu, and Hans van Lint

Identifying and classifying traffic and congestion patterns are essential parts of modern traffic management underpinned by the emerging intelligent transport systems. This paper explores the potential of using a combination of image processing methods to identify and classify regions of congestion within spatiotemporal traffic (speed, flow) contour maps. The underlying idea is to use these regions as (archetype) shapes that in many combinations can make up a wide variety of larger-scale traffic patterns. In this paper, use of a so-called statistical shape model is proposed as a low-dimensional representation of the archetype shape, and an active shape model algorithm coupled with linear classification is developed to classify the patterns of interest. Application of the proposed method is demonstrated with a preliminary set of speed contour maps reconstructed from loop detector data in the Netherlands. The results show that the extended active shape model can be used as a multiclass classifier. In particular, 70% of the traffic patterns in the test data were correctly classified with use of only two archetype shapes and simple logistic classifiers. The results point to the importance of use of expert knowledge by means of (a priori) manual classification of the training examples. This work opens many research directions, including semiautomated searches through traffic databases, automatic detection, and classification of new traffic patterns.

In research, in education, and in practice, spatiotemporal contour maps of speed, density, and flow provide an intuitive means to identify, study, explain, and illustrate (longitudinal) traffic flow phenomena on the basis of either real traffic data or data from traffic simulation models. These phenomena include homogeneous congestion (HC) patterns at bottlenecks, reduced flows related to blockages, wide moving jams (WMJs) that propagate over large distances against the direction of traffic flow, and high-density platoons of heavy vehicles that form moving bottlenecks. With contour maps these phenomena become visible, which helps scientists to formulate hypotheses and derive theories and models to describe the underlying dynamics. Figure 1, for example, shows speed contour plots on the A20 freeway in the Netherlands between Rotterdam and Gouda (Figure 1a), the A13 between Rotterdam and The Hague (Figure 1b),

and the A16 east of Rotterdam (Figure 1c), which were used in a study of traffic dynamics related to severe accidents.

Constructing smooth contour maps from (sensor) data is relatively straightforward with the adaptive smoothing method introduced by Treiber and Helbing (1) and further refined by others (2, 3). Technically, contour maps represent matrices of traffic variables (densities, speeds, flows) on consecutive cells Δx along a route for consecutive time periods Δt , where neither Δx nor Δt must be of constant size or duration, respectively. When mapped onto an underlying grid $\{x_i, t_j\}$ with $i = 1, \dots, N$ and $j = 1, \dots, M$, traffic contour plots can be understood as images with a color mapping from the traffic variable of interest to whatever color coding provides the required visual representation. This image representation opens a wide array of possibilities for traffic scientists and engineers, such as deriving distributions of wave speeds from raw traffic data, without making any prior assumptions (4). These wave speeds and patterns can then be used for calibration and validation of traffic flow models (5).

This paper explores a different application perspective of image processing techniques within the traffic domain, that is, the classification and identification of different traffic patterns. Classifying congestion patterns has a twofold application—for off-line analysis and for real-time predictions. For off-line analysis, it can be used to find days and routes in the historical database with similar congestion patterns. These data can be used by traffic managers to compare two incidents to gain insight for better traffic control. The classification of partial patterns along with metadata such as incident location and severity can also be used for short-term predictions.

Classifying congestion patterns is not a new idea. The ASDA/FOTO method of Kerner et al. is a well-known, patented, theory-laden approach (6), and many machine learning alternatives are available (7–10). Whereas the latter studies focused on class labels that indicate level of service (e.g., light, medium, and heavy congestion), this study's aim, like that of Kerner et al., is to classify entire spatiotemporal congestion patterns. In contrast to that of Kerner et al., this study does not use an elaborate set of expert rules but instead uses flexible and data-driven methods. A supervised learning method was used in an earlier study to classify such patterns with a multiclass support vector machine (11). That study derived an equal-size feature vector for all small and larger traffic patterns identified in traffic contour maps. In this paper, (geometrical) shapes instead of feature vectors are used. The method used comprises several image processing techniques to break down larger-scale traffic patterns into smaller regions. The underlying idea is that these regions constitute base (archetype) shapes that in many combinations can make up a wide variety of larger-scale traffic patterns. With robust identification methods for such archetype shapes, it is possible to dissect, identify, and classify complex traffic patterns automatically. Despite

P. Krishnakumari and H. van Lint, Delft University of Technology, Stevinweg 1, 2600 GA Delft, Netherlands. T. Nguyen, Swinburne University of Technology, Melbourne, Victoria, Australia. Current affiliation for T. Nguyen: Delft University of Technology, Stevinweg 1, 2600 GA Delft, Netherlands. L. Heydenrijk-Ottens, CGI Nederland B.V., George Hintzenweg 89, 3068 AX Rotterdam, Netherlands. H. L. Vu, Monash University, 23 College Walk, 3800 Clayton, Victoria, Australia. Corresponding author: P. Krishnakumari, p.k.krishnakumari@tudelft.nl.

Transportation Research Record: Journal of the Transportation Research Board, No. 2645, 2017, pp. 94–103.
<http://dx.doi.org/10.3141/2645-11>

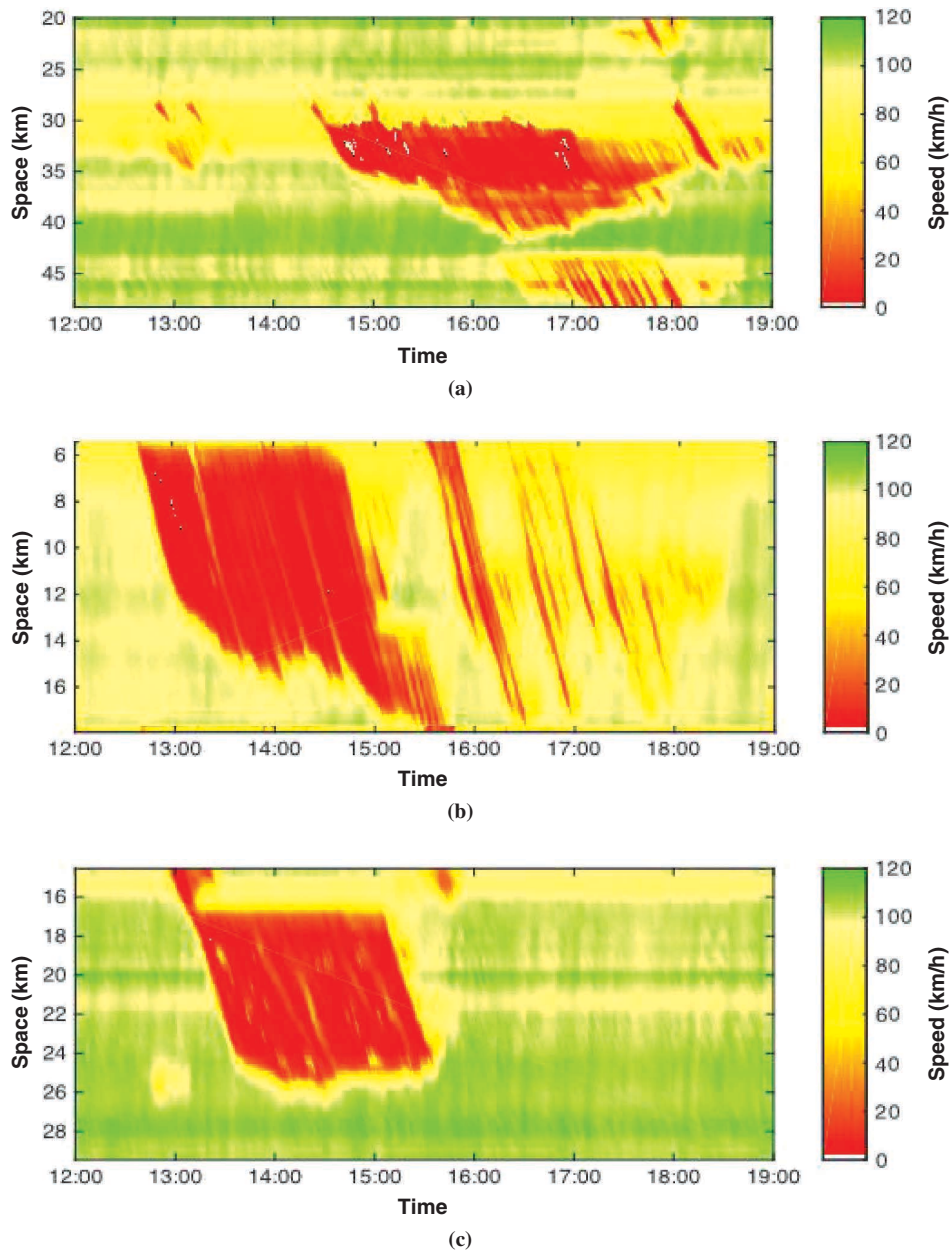


FIGURE 1 Three severe incidents with similar characteristics regarding incident, spatiotemporal extent of queue, and vehicle loss hours: (a) A20, June 9, 2015; (b) A13, March 24, 2015; and (c) A16, January 9, 2011.

the approach's simplicity, the classification results are encouraging, with 70% accuracy attained with just two archetype shapes and simple logistic classifiers and without resorting to the use of additional information (e.g., flow), as in the method of Kerner et al. The applications for this technique are numerous and range from traffic database searching and indexing to traffic state estimation and prediction.

The rest of the paper is organized as follows. The next section gives an overview of the approaches in the multiclass classifier using active shape models (ASMs). Then the experimental setup is described, along with the validation method. The results of the method are presented, and a synthesis of the findings is then given. The paper closes

with preliminary conclusions and an outlook on further improvement of the method.

METHOD

In this work, an approach that is fundamentally different from the state of the art in traffic pattern classification is introduced and developed. Instead of the use of local features to identify characteristics of traffic patterns, (archetype) shapes are used to classify patterns to capture the global structure of these patterns. This method works at a higher abstraction level than feature-based methods. The

shape-based methods are usually used for shape recognition and fitting, mainly in image recognition. This is the first paper, to the authors' knowledge, that introduces shape-based classification in the traffic domain for multiclass classification. The overall method is outlined in Figure 2a.

The idea is to extract contours from speed contour maps and use these as the basis for building the shape model, the base shape classifier (SC) model, and the pattern classifier model. All these components are explained below; a full mathematical explanation is beyond the scope of this paper. The final part of the method explains how the fitting result from ASM has been used for a multiclass prediction process.

Contour Extraction

The basic ingredients of the method are contour maps of detector data generated with the adaptive smoothing method. This method is extensively described elsewhere (1–3). Raw data for 1 day are considered in one contour map, which therefore can contain multiple

congestion patterns for a given day. The individual traffic congestion patterns from each space–time plot are extracted with a naïve contour extraction.

The naïve contour extraction finds the outline of a pattern in an image, as there are multiple patterns in one image (Figure 3a). The first step is to filter out irrelevant information by assuming a speed threshold, v_{thres} , that differentiates between congested and freely flowing traffic. In this paper, $v_{\text{thres}} = 65$. This crude assumption can be relaxed, as discussed later in the paper. This step results in a Boolean mask, as shown in Figure 3b. After thresholding, dilation is used to fill the holes created to provide a smooth mask. This smoothed mask is then used to detect the contours in the image by joining the continuous points along the boundary with similar pixel intensity (12). The detected contours are used to define the boundary to extract each pattern in an image (Figure 3c). Resulting from the naïve contour extraction is a data set of various traffic congestion pattern images.

An additional contour refinement is then performed on the obtained patterns to extract the congestion shape from each pattern. The irrelevant information is filtered out from the naïve extracted pattern image with the same assumption used before; low speed implies congestion, as shown in Figure 3e. A morphological closing strategy that consists of two successive binary transformations, dilation followed by erosion (13), is used to eliminate small and isolated gaps from the relevant regions without destroying the original shape, as shown in Figure 3f. For both transformations, a 3×3 cross-structuring element is used. This binary smoothed mask is then used to detect the contours in the image (12), as shown in Figure 3g.

Manual Classification

The refined contour extraction results in a data set of various traffic congestion pattern images. This data set is manually classified into five classes according to the size of images (space and time extent of traffic jam) and the type of congestion. Table 1 shows the five classes with some examples. This classification is arbitrary—other analysts may come up with more or fewer classes and different criteria. Unsupervised learning also can be used for creating these classes. However, this would require building feature vectors based on the application or using complete black-box methods like deep learning to find all the relevant features in the patterns.

Base Shape Identification and Base Shape Predictor

The manually classified data show that all the patterns in the class are approximately a combination of two base shapes, isolated WMJs (IWMJs) and HC. For example, Table 1 shows that low-frequency WMJ patterns comprise n WMJ shapes, mixed class patterns comprise both HC and IWMJ patterns, and so on. The key distinction in the context of this paper between the two base shapes is that WMJs are stripe-like shapes, whereas HC patterns form triangular shapes. This distinction is backed by theoretical research that investigated a wide range of congestion patterns (1, 14). These works distinguish two main types of congestion based on first-order traffic flow theory, synchronized and WMJs, which includes characteristics similar to those in the base archetypes. Hence, it was decided to start with these two distinct and well-defined base shapes. As more archetypes emerge from data, the proposed method can be scaled to include these base shapes.

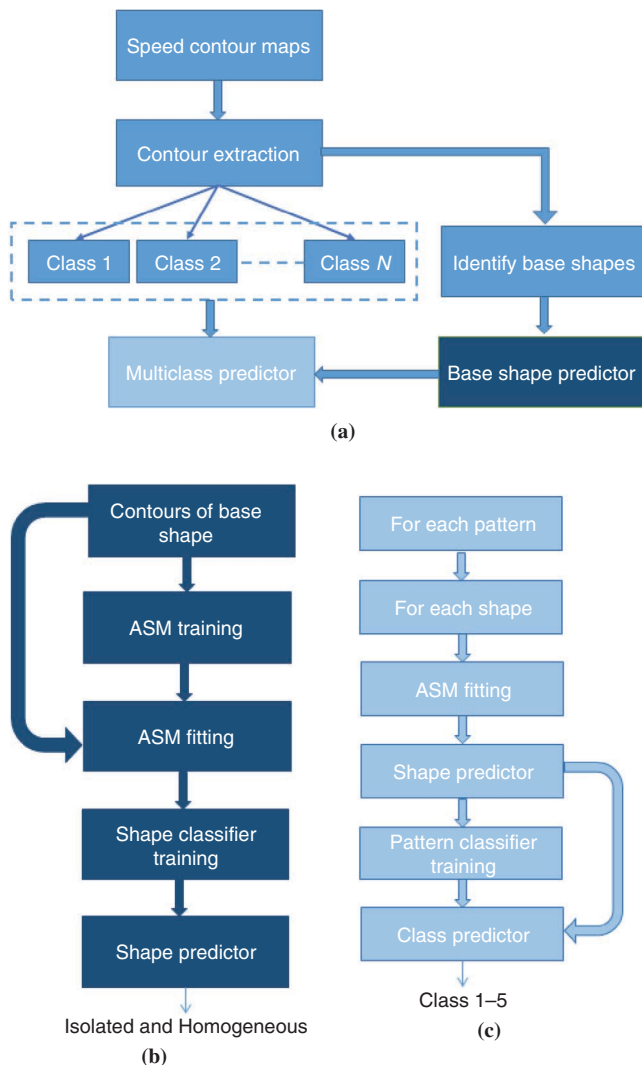


FIGURE 2 Approach: (a) overview, (b) base shape predictor, and (c) multiclass predictor.

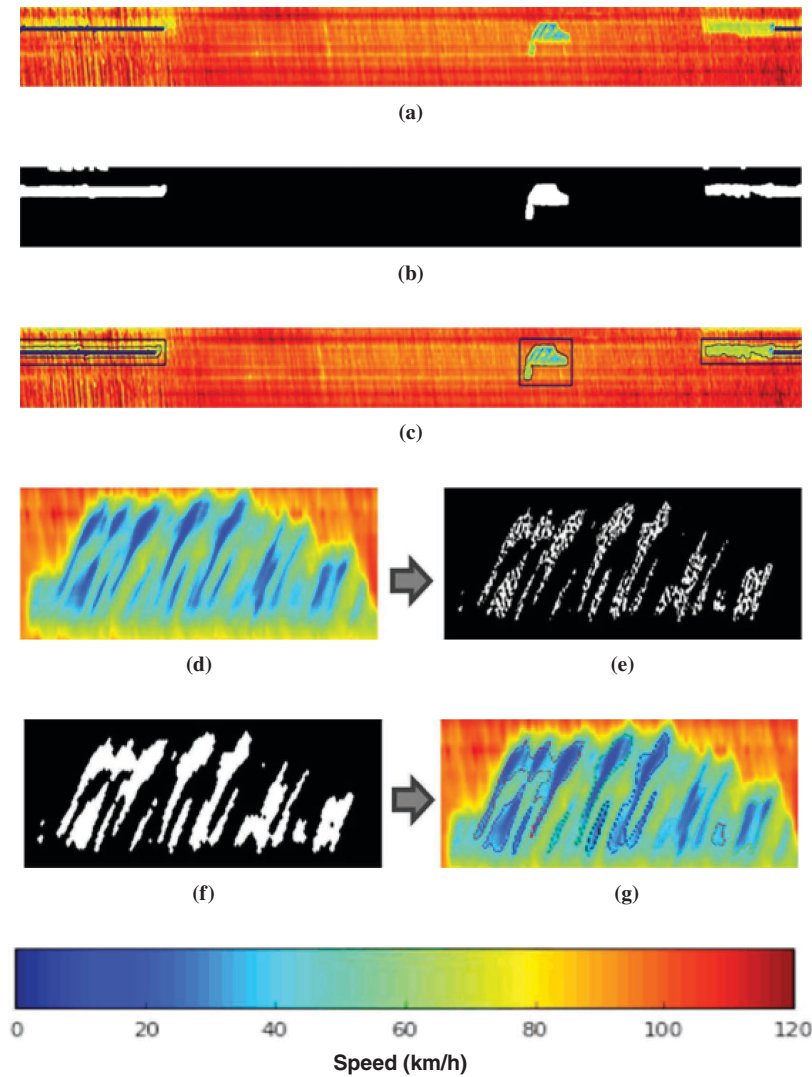


FIGURE 3 Contour extraction, naïve and refined: (a) raw data, multiple patterns; (b) threshold mask; (c) naïve extracted contour with boundary; (d) naïve extracted pattern; (e) binary image; (f) performed closing; and (g) refined contours.

The ASM, given a new observed shape, tries to fit this shape to one of the base shapes. To do this, the originally single-class ASM algorithm was extended to a multiclass ASM through inclusion of a linear classifier that predicts whether a given shape is HC or IWMJ. Figure 2b outlines how to do this step by step. Each step is explained in the following (a full mathematical explanation of the ASM components, including the multiclass extension, is beyond the scope of this paper). The two main phases in ASM are discussed below: constructing a statistical shape model (SSM) for each base shape (model training) and fitting a new shape to this SSM (model fitting).

ASM Training

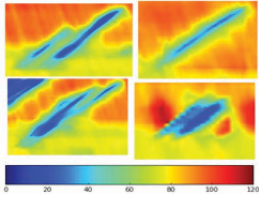
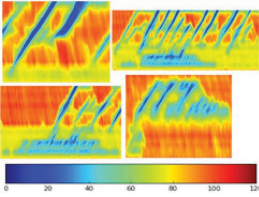
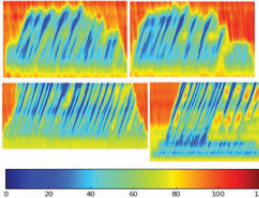
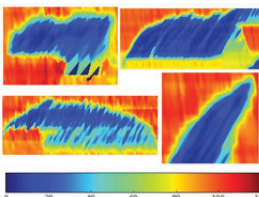
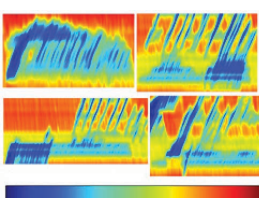
The ASM is a model-based segmentation method introduced by Cootes et al. (15). The method is based on the principle that a shape can be represented by a mean shape and its variations. The mean shape and the variances constitute the SSM, which contains all the parameters that are needed to define that shape. The SSM is used

to find potential instances of the shape model in a new image or contour. Initially, a set of landmarks in the new contour is defined, after which the shape defined by these landmarks is deformed according to the SSM to provide the best fit possible within the SSM. The deformation is based on finding correspondences between the new shape and the shape defined by various SSM components and iterative minimization of a cost function for the fit. The allowed degree of deformation is constrained by the variations defined in the SSM. If the SSM includes large variances, large deformations are possible, and vice versa.

An overview of the method is given below. A more detailed explanation is available elsewhere (15). The steps for building the SSM model (i.e., the training phase) for a base shape are as follows:

Step 1. Align the shapes to the first shape in the data set, and generate a mean shape from the aligned shapes. First, all the shapes must have the same number of landmarks, which is rarely the case. Therefore, a so-called iterative closest point (ICP) method is used

TABLE 1 Class Description for Manually Classified Patterns

Name	Remarks	Examples
Isolated WMJs	Short (1–2 km) high-density traffic jams. Both head and tail of this queue propagate backward with virtually constant speeds (typically –18 km/h). WMJs typically result from large disturbances (e.g., abrupt braking) farther downstream.	
Heterogeneous Congestion 1, low-frequency WMJs	Large-scale light congestion patterns with a few WMJs emitting from the congested area.	
Heterogeneous Congestion 2, high-frequency WMJs	Large-scale light congestion patterns with many WMJs emitting from the congested area.	
Homogeneous congestion	High-density (low-speed) severe congestion regions, typically caused by incidents or other lane blockages. In terms of shape, the downstream front is stationary, whereas the upstream front moves with various shock wave speeds.	
Mixed large-scale pattern	Combination patterns not falling in either of the other categories.	

to register the contours from all classes to a given model contour. In ICP, a point set is transformed to best match the chosen model, where the transformations are revised iteratively until the distance between the point set and the model is minimized (16). The alignment itself is achieved with Procrustes analysis because the standard ASM also uses this method for alignment (17).

Step 2. Realign the shapes to the mean shape and generate a new mean shape from the newly aligned shapes.

Step 3. Repeat Step 2 (update the mean shape) until convergence.

Step 4. Finally, apply principal component analysis (PCA) to compute the eigenvectors and eigenvalues of the aligned shapes. The SSM components are the eigenvectors of the centered shapes in the training data, and the variances are the eigenvalues of these shapes. When PCA is applied to the data, any shape (within the training set) \mathbf{x} (an n -dimensional vector of points) can be approximated with

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (1)$$

where

$\bar{\mathbf{x}}$ = SSM mean shape having point correspondences with \mathbf{x} and the same dimension,

\mathbf{P} = SSM principal components, and

\mathbf{b} = parameters corresponding to the SSM components that deform the shape.

This is the basis for fitting a new set of landmarks to the SSM. The first four components of the IWMJ base class are shown in Figure 4a. This figure explains the shape variations of the class according to the mean shape.

ASM Fitting

Given a new shape \mathbf{Y}' for testing, the shape is registered with the mean shape by using ICP to compute \mathbf{Y} . With \mathbf{Y} and the SSM model

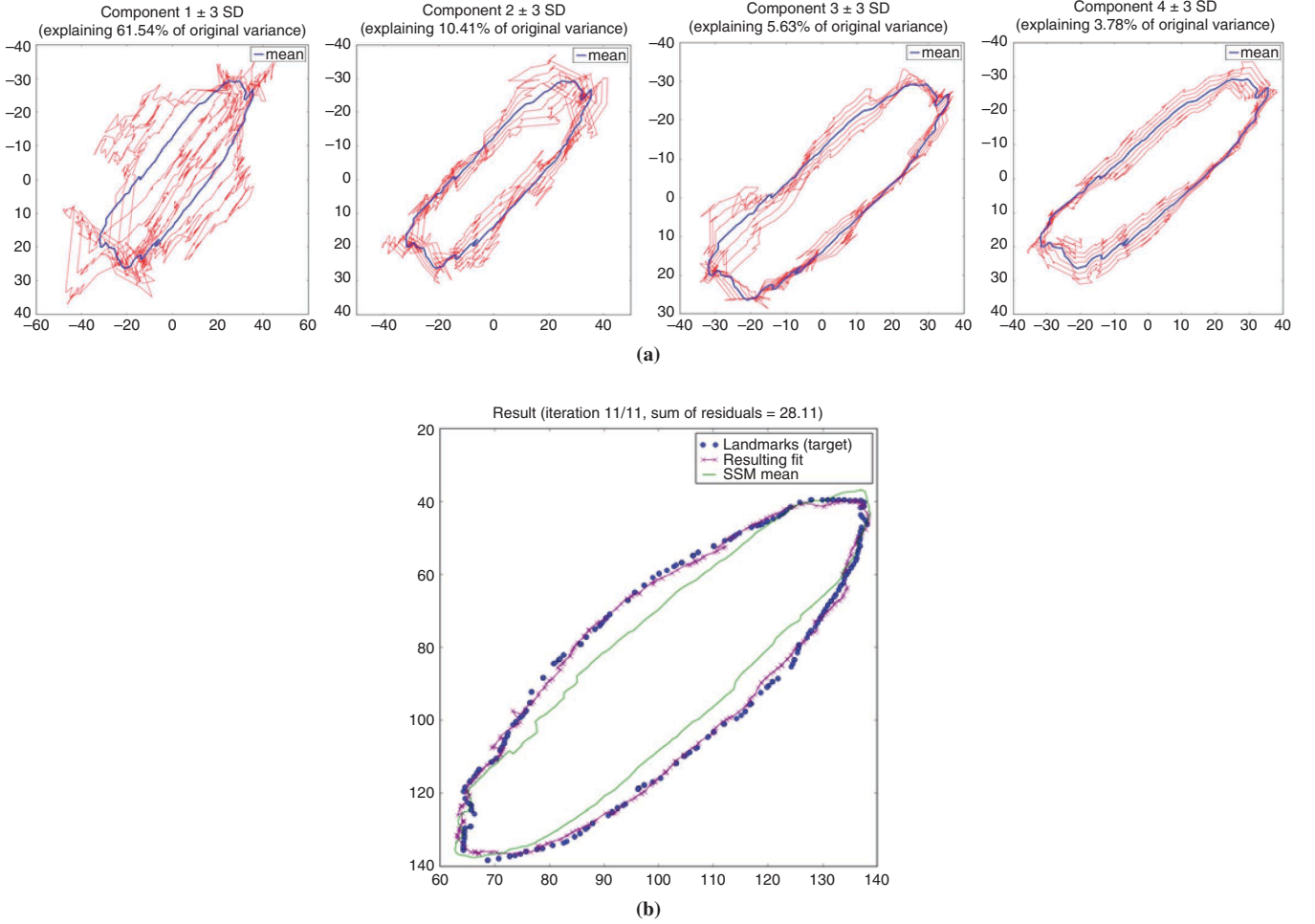


FIGURE 4 IWMJ: (a) first four PCA components of IWMJ shape model and (b) ASM fitting for IWMJ shape after N th iterations ($N = 11$).

of the shape, ASM fitting finds the model points \mathbf{x} that best fit \mathbf{Y} . The steps for ASM fitting are as follows:

Step 1. Initialize the shape parameter \mathbf{b} as 0, implying model points = mean shape, that is, $\mathbf{x} = \bar{\mathbf{x}}$.

Step 2. Generate the model points positions with $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$, where \mathbf{P} is the SSM principal component.

Step 3. Find the pose parameters transform that best aligns the model points \mathbf{x} to the new set of landmarks \mathbf{Y} by using Procrustes analysis (17).

Step 4. Project \mathbf{Y} into the model coordinate frame \mathbf{Y}' by using the inverse transform from Equation 1.

Step 5. Update the shape parameters \mathbf{b} to match \mathbf{Y}' by finding the least squares solution of $\mathbf{A}\mathbf{x}' = \mathbf{B}$, where \mathbf{A} is \mathbf{P} , \mathbf{x}' is \mathbf{b} , and $\mathbf{B} = \mathbf{Y}' - \bar{\mathbf{x}}$.

Step 6. Repeat Steps 2 through 5 until convergence.

An example of an ASM fitting result is shown in Figure 4b. The fitting error metric for the ASM that is used as the convergence criteria is the Euclidean distance between the mean shape and the ASM fitted shape. The ASM stops the iteration when there is no significant difference in the error metric result from the previous iteration and the current iteration. Here, an error difference threshold of 0.0001 was used, which is statistically insignificant with respect to the error rate and the shape surface area.

Designing and Training a Base Shape Classifier

Now, with a working ASM model for both base shapes (IWMJ and HC), a shape classifier is needed that can predict which of these base shapes provides the best representation for a newly found shape in a speed contour plot. First, the new shape is fitted with both base shapes, resulting in fitting error values e_i (error on the IWMJ base shape) and e_h (error on the HC base shape), respectively. These errors are defined as the Euclidean distance between the SSM mean and the fitted shape. Additionally, to increase prediction accuracy, a third metric is computed that is based on additional nonshape properties of IWMJ and HC shapes, respectively. As an example, in this case a metric is based on the spatiotemporal area covered by the shape \mathbf{a} (in meter \times seconds) and the gradient \mathbf{g} (the variation in speed in the given area \mathbf{a}). The compound metric is the ratio \mathbf{g}/\mathbf{a} , which can be understood as the amount of heterogeneity (variation in speed) per unit space \times time, resulting in a three-dimensional feature vector $(e_i, e_h, \mathbf{g}/\mathbf{a})$. To build the classifier, a well-known method in linear classification, logistic regression, is applied (18). It originally was a conditional probability model that measures the likelihood relationship between a specific output and an input by using a logistic function:

$$p(t) = \frac{1}{1 + e^{-t}} \quad (2)$$

where t is a linear combination of input feature vector x and β is the vector of coefficients, which is considered a model characteristic. In preparing training data, isolated and homogeneous shapes are labeled as 0 and 1, respectively. These numbers are supposed to be outputs of logistic function. Coefficient vector β_{sc} is trained to minimize the cost of matching logistic function to training data, which is measured by Euclidean distance.

$$\text{cost} = \sum_{x_i \in \text{training set}} (p(\beta_{sc} \cdot x_i) - y_i)^2 \quad (3)$$

$$x_i = \left(e_i, e_h, \frac{g}{a} \right) \quad (4)$$

$$y_i = \begin{cases} 0 & \text{for isolated shapes} \\ 1 & \text{for homogeneous shapes} \end{cases} \quad (5)$$

The resulting (trained) model is the (base) shape predictor, which classifies a shape as either IWMJ or HC. The shape classifier is used to build feature vectors for training pattern classifiers for multiclass classification.

Using the Base Shape Classifier

Given a new shape, the shape is fitted to the IWMJ and HC SSM as described in the section on ASM fitting to create a description vector

$x_{sc} = (e_i, e_h, g/a)$. This vector is used by the shape classifier model to make the decision based on the following equation:

$$\widehat{\text{shape}} = \begin{cases} \text{homogeneous shape} & \beta_{sc} \cdot x_{sc} > 0 \\ \text{isolated shape} & \text{otherwise} \end{cases} \quad (6)$$

This equation together with the logistic function gives a straightforward explanation for the classifier to make the decision. The new shape is classified as a homogeneous shape if the probability given by the logistic function is greater than 0.5. An overview of the method is shown in Figure 5a.

Multiclass Pattern Classifier and Predictor

The aim of this work is to classify a given traffic (congestion) pattern into one of five predefined classes (Table 1). The final step now is to design, train, and test a classifier that can do this. A schematic overview of the proposed method is shown in Figure 2c.

Since each of these five patterns can be broken down into combinations of two (archetype) base shapes (IWMJ and HC), the first step in classifying a congestion pattern is to break down the pattern into these base shapes, as shown in Figure 5a. For each of the five classes a training set can be built. The output data are the class labels (1 to 5), and the input data are equal to a simple two-dimensional feature vector (n_i, n_h) , where n_i is the number of occurrences of the IWMJ shape in the given pattern and n_h is the number of occurrences of the HC shape in the given pattern.

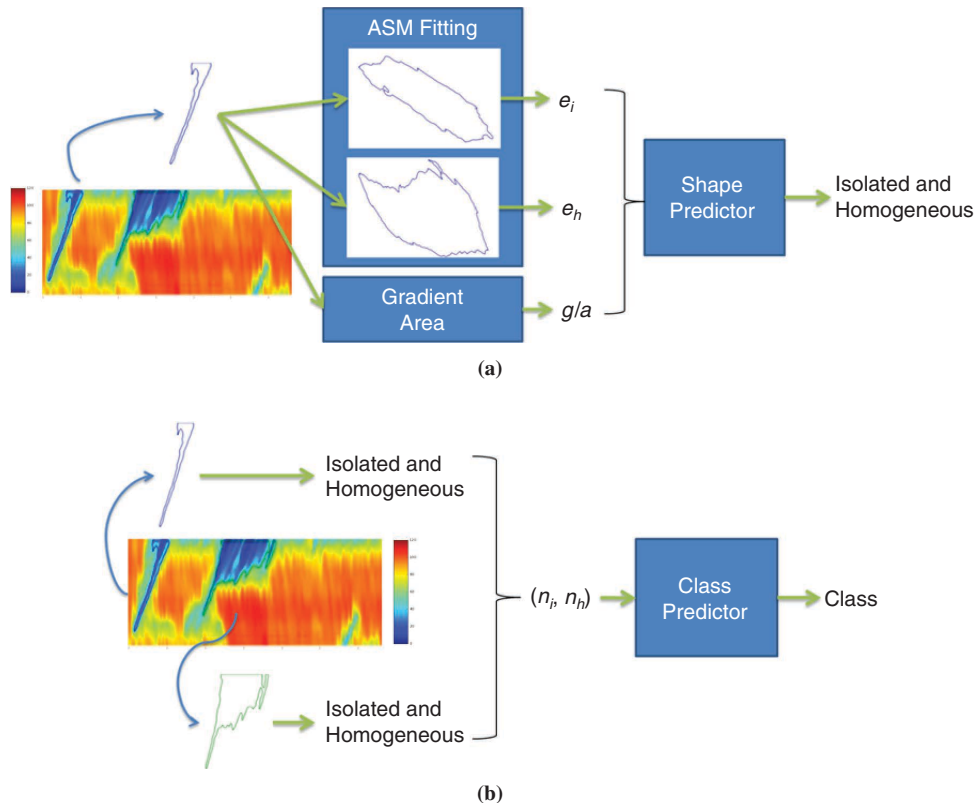


FIGURE 5 Multiclass predictor: (a) base shape prediction given shape and (b) class prediction given new pattern.

The general idea of the multiclass pattern predictor is given in Figure 5b. Multiclass logistic regression is used to implement it. A simple yet efficient method, the so-called one-versus-all (OVA) scheme, was used for training; it constructs one binary logistic regression model for each class and assigns a probability p to the training sample belonging to this class versus the probability that it belongs to any of the other classes.

After the classifier is trained, there are five coefficients, $\beta_{FC1}, \dots, \beta_{FC5}$, corresponding to five binary logistic regression models of five classes. For a given (new) pattern with feature vector $\mathbf{x}_{FC} = (\mathbf{n}_i, \mathbf{n}_h)$, the class label is assigned to it, giving rise to the largest probability, that is,

$$\widehat{\text{class}} = \underset{i}{\operatorname{argmax}} p(\beta_{FCi} \cdot \mathbf{x}_{FC}) \quad (7)$$

EXPERIMENTAL SETUP

The data used in this study come from the National Data Warehouse for Traffic Information (19), a Dutch organization that archives and provides real-time access to traffic data from the Dutch agency of the Ministry of Infrastructure and the Environment (Rijkswaterstaat), the 12 Dutch provinces, two metropolitan regions, and four of the largest cities in the Netherlands (Amsterdam, Rotterdam, Utrecht, and The Hague). The data used for the experiments were collected from two heavily congested roads in the Netherlands:

- Southbound A13 from Den Haag-Zuid to Rotterdam center and
- Eastbound A15 from Havens 5500–5700 to Rotterdam IJsselmonde.

Space–time plots of carriageway speed for these two roads were constructed with all available loop data for the entire month of March 2015. Each space–time plot represents a period of 24 h from 00:01 to 23:59 at a resolution of 30 s and 100 m. Each of the plots contains multiple congestion patterns. After each pattern was identified and extracted separately with naïve contour extraction, there were 140 traffic congestion patterns detected on the first road and 160 patterns on the second road. Because only large-scale patterns with similar space–time ratios are of interest here, 120 patterns were selected to create the classes. The patterns were manually classified into five classes based on spatiotemporal extent and characteristics of traffic congestion, as shown in Table 1. The number of patterns per class is small; the first trial was intended to demonstrate the ideas and learn lessons for larger-scale application. After the classes were identified by experts, the naïve contours were refined to construct better distinctive contours from the patterns. From these refined con-

tours, two base shapes were identified; 35 contours were manually chosen to build the SSM models of both homogeneous and isolated base shapes. These same contours are used to train the shape classifier.

A simple OVA approach was used for the multiclass predictor that reduced the problem of classifying contours among five classes into five feature vectors, where each model discriminated a given class from the other four classes (20). For this OVA approach, there were $N = 5$ binary classifiers, the k th classifier trained with positive examples belonging to class k and negative examples belonging to the other four classes. The classifier that produced the maximum output was considered to be the best fit. Rifkin and Klautau stated that, provided the binary classifiers are tuned well, this OVA approach is extremely powerful and produces results that often are at least as accurate as other, more complex approaches (20). The accuracy of the ASM OVA classifier was measured to judge the overall efficiency of the algorithm with the following formula:

$$\text{accuracy} = \frac{\text{number of correctly classified images}}{\text{total number of images}} \quad (8)$$

A confusion matrix, or contingency table (21), is constructed to investigate which class is behaving poorly, and that class is studied further for a better understanding of the data and to make future recommendations to improve accuracy.

RESULTS AND DISCUSSION

This section presents the results of the proposed method. The OVA ensemble of ASM models achieved an average prediction accuracy of 70%, which is relatively low compared with the state of the art. However, this method represents five classes by using only two archetype shapes, whereas other methods need more degrees of freedom for defining each class. Thus, the presented method can constrain the classification complexity while providing satisfactory accuracy for a preliminary study with such a small data set. A more elaborate confusion matrix is given in the Table 2. The table shows the percentage of correctly and erroneously classified patterns in each class given the ground truth. For example, for the IWMJs class, the ground truth patterns in that class were correctly classified with 81% accuracy, but there were some patterns in other classes that were wrongly classified as IWMJ. The high-frequency WMJs class contains the most patterns that were wrongly classified with 50% accuracy.

Table 2 shows that two classes have low accuracy compared with the other classes, namely, high-frequency WMJ and mixed. It is

TABLE 2 Confusion Matrix for OVA Evaluation

Known	Predicted				
	Isolated	Low Frequency	High Frequency	Homogeneous	Mixed
Isolated	0.81	0.03	0.03	0.13	0.00
Low frequency	0.03	0.76	0.09	0.09	0.03
High frequency	0.00	0.28	0.50	0.16	0.06
Homogeneous	0.21	0.00	0.00	0.74	0.05
Mixed	0.05	0.00	0.22	0.17	0.56

hypothesized that this result reflects the small training data set or the limited representativeness of the samples in that class. The data were further investigated to qualitatively test this hypothesis.

Two of the wrongly classified patterns are shown in Figure 6a; in this case, both were wrongly classified as mixed patterns instead of low-frequency WMJ and HC, respectively. These patterns are actually mixed congestion patterns. The classifier confused these because they have a feature vector similar to that of the mixed class. Successful classification depends on the subjective manual classification process and the degree to which labeling patterns that can be distinguished through OVA ASM are successful. This definition of success was confirmed by an initial analysis of the wrongly classified patterns in which the authors agreed with the classifier's decision on 15% of all wrongly classified patterns rather than with the initial manual classification. To reduce the subjectivity of the manual classification, unsupervised clustering and reinforcement learning can be used as an alternative for building the classes.

A second reason for the low accuracy is the constraints of the assumptions that have been made for extracting the contours, such as thresholding at 65 km/h. These are evident in Figure 6b, where the highlighted shapes consist of combinations of two or more isolated shapes and combinations of isolated and homogeneous shapes, respectively. Finally, speed was used exclusively to distinguish patterns. Identifying and classifying distinct traffic patterns also requires information on the flows, particularly when one is distinguishing between congested patterns. Using just the speed and the naïve single speed thresholding can uniquely identify only a few patterns. To extend the repertoire of the classifiers, more archetypes

are needed to approximate all the unique shapes that are present within the patterns, both speed and flow patterns must be combined, and meta-information (speed limits, geometry, etc.) may be needed to construct a more dynamic thresholding method to extract the shapes from the patterns.

Still, given all these limitations, the simplicity and transparency of the method (a few base shapes + logistic regression) offers great potential for further research and application development.

CONCLUSION AND FUTURE WORK

This paper proposed a method, consisting of contour detection, shape models, and a multiclass OVA classifier, to automatically classify spatiotemporal traffic patterns with network sensor data. The various components were adequate for labeling complex traffic patterns with acceptable accuracy, although the data set used was too small to warrant definite conclusions. With only two archetype shapes and simple logistic classifiers, 70% accuracy of the classification on the test data was achieved. Furthermore, for the 30% of cases in which the classifier could not decide on any of the five designated patterns, the contour shapes were affected by the assumptions and subjective manual labeling was used to construct the training data.

There are many future directions of research that can further improve the accuracy. First, the ensemble scores can be used to directly put a confidence score to each classification. Further sophistication can be reached by ensemble bootstrapping or more modern Bayesian techniques. Second, an iterative manual-automated classi-

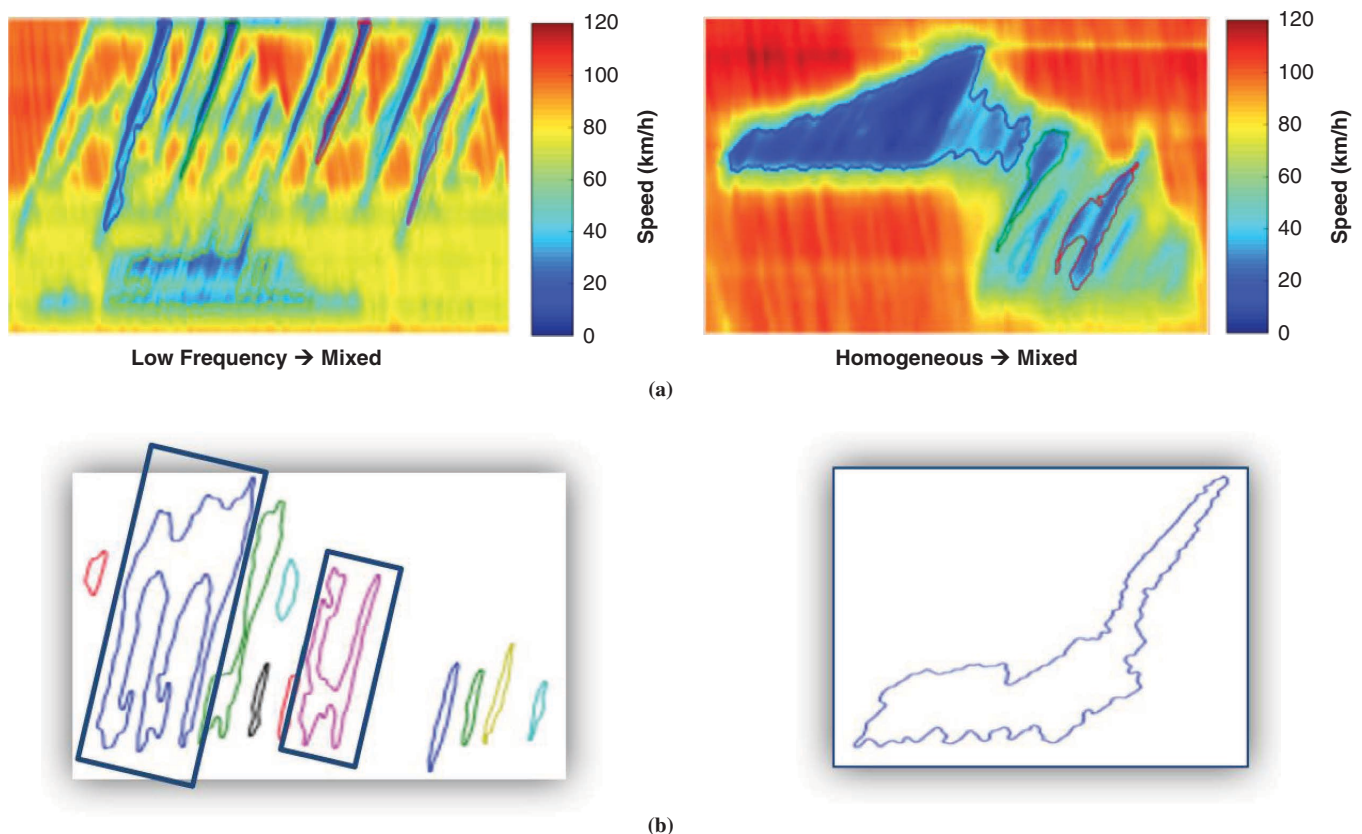


FIGURE 6 Synthesis of wrongly classified patterns: (a) manually classified and (b) potential base shapes.

fication procedure is envisaged in which the manual classification is reevaluated after each training round with the classification scores of the OVA ASM. This procedure could yield splitting or combining classes, hierarchically subdividing classes, or otherwise. Third, metadata (type of date and time, circumstances, topological characteristics, etc.) can be combined to better classify the patterns. Another extension will combine speed plots with flow contour plots for better definition of classes of congestion and for improved accuracy. Finally, the database will be further enriched with more congestion patterns, with a goal of identifying more complex patterns.

ACKNOWLEDGMENTS

The authors acknowledge the Dutch National Data Warehouse for Traffic Information for sponsoring this project. This work received support from the SETA Project, funded by the European Union's Horizon 2020 Research and Innovation Program, and an Australian Research Council fellowship grant. The authors thank the anonymous reviewers for their constructive remarks.

REFERENCES

1. Treiber, M., and D. Helbing. Reconstructing the Spatio-Temporal Traffic Dynamics from Stationary Detector Data. *Cooperative Transportation Dynamics*, Vol. 1, No. 3, 2002, pp. 3.1–3.24.
2. Schreiter, T., H. van Lint, M. Treiber, and S. Hoogendoorn. Two Fast Implementations of the Adaptive Smoothing Method Used in Highway Traffic State Estimation. In *Proceedings of the 13th International Conference on Intelligent Transportation Systems, 2010*, IEEE, New York, 2010, pp. 1202–1208. <https://doi.org/10.1109/ITSC.2010.5625139>.
3. van Lint, J., and S.P. Hoogendoorn. A Robust and Efficient Method for Fusing Heterogeneous Data from Traffic Sensors on Freeways. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 25, No. 8, 2010, pp. 596–612. <https://doi.org/10.1111/j.1467-8667.2009.00617.x>.
4. Schreiter, T., H. van Lint, Y. Yuan, and S. Hoogendoorn. Propagation Wave Speed Estimation of Freeway Traffic with Image Processing Tools. Presented at 89th Annual Meeting of the Transportation Research Board, Washington, D.C., 2010.
5. Treiber, M., and A. Kesting. Validation of Traffic Flow Models with Respect to the Spatiotemporal Evolution of Congested Traffic Patterns. *Transportation Research Part C: Emerging Technologies*, Vol. 21, No. 1, 2012, pp. 31–41. <https://doi.org/10.1016/j.trc.2011.09.002>.
6. Kerner, B.S., H. Rehborn, M. Aleksic, and A. Haug. Recognition and Tracking of Spatial–Temporal Congested Traffic Patterns on Freeways. *Transportation Research Part C: Emerging Technologies*, Vol. 12, No. 5, 2004, pp. 369–400. <https://doi.org/10.1016/j.trc.2004.07.015>.
7. Andrews Sobral, L. O., L. Schnitman, and F. De Souza. Highway Traffic Congestion Classification Using Holistic Properties. Presented at 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, 2013.
8. Jin, X., D. Srinivasan, and R. L. Cheu. Classification of Freeway Traffic Patterns for Incident Detection Using Constructive Probabilistic Neural Networks. *IEEE Transactions on Neural Networks*, Vol. 12, No. 5, 2001, pp. 1173–1187. <https://doi.org/10.1109/72.950145>.
9. Riaz, A., and S.A. Khan. Traffic Congestion Classification Using Motion Vector Statistical Features. In *Proceedings of the 6th International Conference on Machine Vision*, International Society for Optics and Photonics, 2013.
10. Thianniwet, T., S. Phosaard, and W. Pattara-Atikom. Classification of Road Traffic Congestion Levels from Vehicle's Moving Patterns: A Comparison Between Artificial Neural Network and Decision Tree Algorithm. In *Electronic Engineering and Computing Technology* (S. L. Ao and L. Gelman, eds.), Springer, New York, 2010, pp. 261–271. https://doi.org/10.1007/978-90-481-8776-8_23.
11. Nguyen, H.N., P. Krishnakumari, H.L. Vu, and H. van Lint. Traffic Congestion Pattern Classification Using Multi-Class SVM. In *Proceedings of the 19th International Conference on Intelligent Transportation Systems*, IEEE, New York, 2016, pp. 1059–1064.
12. Suzuki, S., and K. Abe. Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision Graphics and Image Processing*, Vol. 30, No. 1, 1985, pp. 32–46. [https://doi.org/10.1016/0734-189X\(85\)90016-7](https://doi.org/10.1016/0734-189X(85)90016-7).
13. Vincent, L. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing*, Vol. 2, No. 2, 1993, pp. 176–201. <https://doi.org/10.1109/83.217222>.
14. Helbing, D., M. Treiber, A. Kesting, and M. Schönhof. Theoretical vs. Empirical Classification and Prediction of Congested Traffic States. *European Physical Journal B*, Vol. 69, No. 4, 2009, pp. 583–598. <https://doi.org/10.1140/epjb/e2009-00140-5>.
15. Cootes, T.F., C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models—Their Training and Application. *Computer Vision and Image Understanding*, Vol. 61, No. 1, 1995, pp. 38–59. <https://doi.org/10.1006/cviu.1995.1004>.
16. Chetverikov, D., D. Svirko, D. Stepanov, and P. Krsek. The Trimmed Iterative Closest Point Algorithm. In *Proceedings of the 16th International Conference on Pattern Recognition*, IEEE, New York, 2002, pp. 545–548. <https://doi.org/10.1109/ICPR.2002.1047997>.
17. Goodall, C. Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society, Series B: Methodological*, Vol. 53, 1991, pp. 285–339.
18. Hosmer, D.W., S. Lemeshow, and R. X. Sturdivant. Introduction to the Logistic Regression Model. In *Applied Logistic Regression*, John Wiley & Sons, New York, 2000, pp. 1–30.
19. National Data Warehouse of Traffic Information. <http://www.ndw.nu/en/>.
20. Rifkin, R., and A. Klautau. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research*, Vol. 5, 2004, pp. 101–141.
21. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognition Letters*, Vol. 27, No. 8, 2006, pp. 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>.

The Standing Committee on Artificial Intelligence and Advanced Computing Applications peer-reviewed this paper.