Efficient Feature Extraction for Highway Traffic Density Classification

Conference Paper · November 2015

DOI: 10.1109/IranianMVIP.2015.7397494

3 authors, including:

Parvin Ahmadi
Sharif University of Technology

Iman Gholampour
Sharif University of Technology

Efficient Feature Extraction for Highway Traffic Density Classification

Mina Abasi Dinani, Parvin Ahmadi
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran

Iman Gholampour
Electronics Research Institute
Sharif University of Technology
Tehran, Iran

***Abstract*—Traffic density estimation is one of the most challenging problems in Intelligent Transportation Systems. In this paper, we estimate the traffic flow density based on classification. Various new efficient features are introduced for distinguishing between different traffic states, including number of key-points, edges of difference-image and moving edges. These features describe the traffic flow without any need to individual vehicles detection and tracking. We experiment our proposed approach on a standard database and some real videos from Tehran roads. The results show high accuracy performance of our method, even in changes of environmental conditions (e.g., lighting), by using efficient features. Duo to low computational cost, our proposed approach for traffic density estimation is applicable in real time applications.**

Keywords—Feature extraction; Intelligent Transportation System; Support Vector Machine; traffic density classification

I. Introduction

Traffic density estimation is one of the challenging problems in intelligent traffic control systems. Road traffic density estimation provides important information for road planning, intelligent road routing, road traffic control, vehicular network traffic scheduling, routing and dissemination. Many vehicle drivers choose their path based on the traffic density information. Thus, providing real time traffic density information with least delay is very important.

Nowadays, in many traffic control centers, traffic density estimation is performed by human operators. Each operator is responsible for monitoring almost large numbers of cameras. Fig. 1, that shows the Tehran traffic control center, illustrates this issue. Due to increasing number of traffic cameras and limited number of operators, the estimation of traffic flow density information is done with considerable delay. Therefore, developing an intelligent system for traffic density estimation in traffic control centers seems necessary.

Fig.1. Tehran traffic control center

Recently, many researchers have employed machine vision techniques for traffic density estimation. Different methods have been developed for this purpose, which are based on:

- Optical flow
- Detection and tracking
- Dynamic texture
- Counting vehicles
- Supervised classification

The approaches that use optical flow for traffic density estimation, obtain traffic flow information directly without any need to detect individual moving vehicles. Firstly, using an appropriate algorithm for optical flow extraction from video, the distribution of traffic flow vectors is calculated. This distribution provides a general description of traffic flow. Then, changes of traffic states can be detected using different ways to compute the difference between two probabilistic distributions, like Kullback-Leibler (KL) divergence [1]. The problem of these methods is that extraction of traffic flow information in traffic scenes is hard duo to different weather, illumination and environmental conditions.

Most of approaches for traffic information extraction are a combination of detection and tracking (e.g. the one proposed in [2]). Generally, these methods consist of three main steps: 1. moving vehicles detection; 2. vehicles tracking; and 3. using trajectory information to describe the traffic states [3]. For example in [4], firstly the vehicles are detected. For this purpose, a combination of background subtraction and motion extraction is employed. Motion information is extracted based on optical flow. After vehicle detection, they are tracked between frames using the tracking methods like Kalman filter. Then, vehicles trajectories are described by multinomial curves. These curves are then utilized for grouping the videos. The major drawback of these methods is related to detection and tracking. Changes of environmental conditions (e.g., lighting), shadow, occlusion and image blurring decrease the detection performance. Moreover, the performance of most existing tracking approaches is decreased when there are large numbers of objects in the scene which usually occurs in highways and roads [3].

Traffic patterns could be classified based on their dynamics in spatio-temporal region (x, y, t) rather than analyzing the individual vehicles. As an example, the work in [5] proposed that traffic flow, which is the movement of individual cars between two points and their interactions, can be modeled directly as a dynamic texture [6]. In reference [5] approach, dynamic texture is defined as a spatio-temporal autoregressive stochastic process. Then for grouping a traffic video, the distance between these stochastic processes is calculated. Despite high accuracy of this method, large computational load to fit the model makes this approach impractical for real time traffic applications [3].

In some researches, traffic density estimation is performed based on counting the number of vehicles. In [7] and [8] some features are used to train Support Vector Machine (SVM) to detect the existence of a vehicle in a special region of image, called virtual loop detector. This information about number of passing vehicles over all video frames is used to understand the traffic density of the road.

Another approach to traffic density estimation is to consider this problem as a problem of supervised classification. In training phase, one or more classifiers are trained using sample labeled videos from different traffic states. Traffic density states are usually classified into three categories: light, medium and heavy traffic states. Various features can be utilized for describing different categories of input videos. After training the classifiers, in test phase, the new traffic scenes are classified to one of the learned classes. Fig. 2 shows the block diagram of this approach. Various features and classifiers are used in literatures for this method. In [9] video motion vectors, image energy, edge and texture are extracted as traffic descriptor features. As classifier, Gaussian Mixture Hidden Markov Model is trained to classify six traffic density states. In [10] edge and texture features are extracted. These features are then applied to train four Hidden Markov Models (HMM). Each HMM is trained for one of the traffic density states: Empty, Low, High, and Full. The most likely HMM determines the most likely traffic state.

In this paper, we estimate the traffic flow density based on classification. Various efficient features are introduced for distinguishing between different traffic states. These features describe the traffic flow without any need to individual vehicles detection and tracking. We experiment our proposed approach on a standard database and some real videos from Tehran roads. The results show high accuracy performance of our method, even in changes of environmental conditions (e.g., lighting), by using efficient features. Duo to low computational cost, our proposed approach for traffic density estimation is applicable in real time applications.

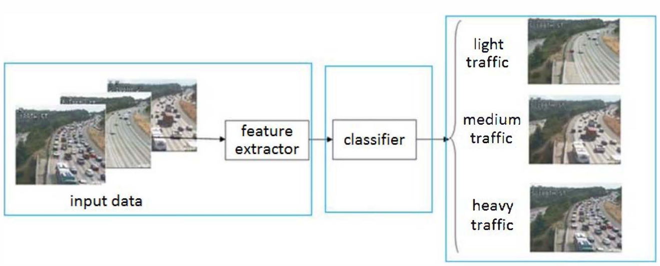

Fig.2. Block diagram of a traffic density system based on classification

II. OUR APPROACH

We utilize SVM as classifier. Three traffic states are considered: light, medium and heavy. The most critical step in classification problems is choosing efficient features with high distinguishing capability. Different features are introduced and employed for traffic states classification in this section. We categorize our proposed features to be used for traffic density estimation, in two groups of static and dynamic features.

A. Static Features

In the problem of traffic density estimation, the larger number of vehicles in an area shows the denser traffic. This property can be described using static features which are first calculated from individual frames and then averaged in a given time duration of video. In this paper, texture [11] and edge histogram [12] are extracted from one frame and used as static features for traffic density estimation. In addition, the number of key-points is proposed as a new static feature.

a) Edge histogram

The number of edges in an image may be regarded as a measure of how crowded the image is [8]. Modified MPEG-7 method of calculating edge histogram gives a good representation for the edges in an image. The first step of calculating MPEG-7 edge histogram description is reducing the image grey levels to 128 levels that act as a noise removal procedure. The image is then broken into predefined number of blocks (here 16 blocks). Each block which completely lies inside the ROI (here the entire road surface) is used to calculate the edge histogram. Blocks are labeled with five edge types of 0, 45, 90, 135 degrees and non-directional edge. Edge histogram contains 16×5=80 bins. Every bin becomes one element of 80-dimensional edge feature vector. For more details regarding the MPEG-7 edge descriptor histogram refer to [12].

b) Texture

As the number of vehicles reduces in a road, the texture of the scene becomes more similar to the road surface texture. To obtain the texture of a scene we calculate the first-order and second-order statistics. With 6 first-order statistics and 5 second-order statistics, an 11-dimensional feature vector is obtained.

1. First-order statistics

First-order texture statistics are calculated based on histogram of gray levels, as described in [11]. These statistics includes: overall mean, standard deviation, skew, R-inverse variance, uniformity, Entropy.

2. Second-order statistic

Second-order statistics are calculated using Gray level Co-Occurrence Matrix (GLCM) [11]. GLCM is calculated based on position operators. We consider position operators with one pixel length and four different directions: 0, 45, 90, 135 degrees. The symmetric GLCM and its corresponding statistical descriptors are calculated for each frame with each position operator. Its descriptors are then averaged over the four angles to give a rotationally-invariant texture feature description [8].The descriptors are: angular second moment (energy), Entropy, inverse derivative moment, contrast and correlation

c) Number of key-points

Key-points can be detected by algorithms such as Scale-Invariant Feature Transform (SIFT) [13], Speed Ups Robust Features (SURF) [14] and Features from Accelerated Segment Test (FAST) [15]. In these algorithms key-points are selected at distinctive locations such as blobs, corners and T-junctions. Usually key-points are used for detecting and tracking objects in video frames [14]. In this paper, we employ the key-points for traffic density detection.

Consider two images with different level of details. The one with higher level of details have more corners, T-junctions and blobs. Therefore SIFT or SURF detector detects more key-points in this image than the other one. This fact is shown in Fig. 3. In this figure each row contains three different images. The number of key-points detected by SURF algorithm in light, medium and heavy traffic state is 209, 261and 421, respectively.

As it is obvious in Fig. 3, when traffic density increases, the level of details in frames increases too. Therefore, the number of key-points could be used to distinguish between different classes of traffic. For this purpose, we divide the frame image into some number of blocks. The number of key-points in each block shows the local density in a part of road. The number of blocks depends on the size of ROI. By considering the optimum number of 64 blocks, obtained experimentally, a 64-dimensional feature vector of the number of key-points is obtained.

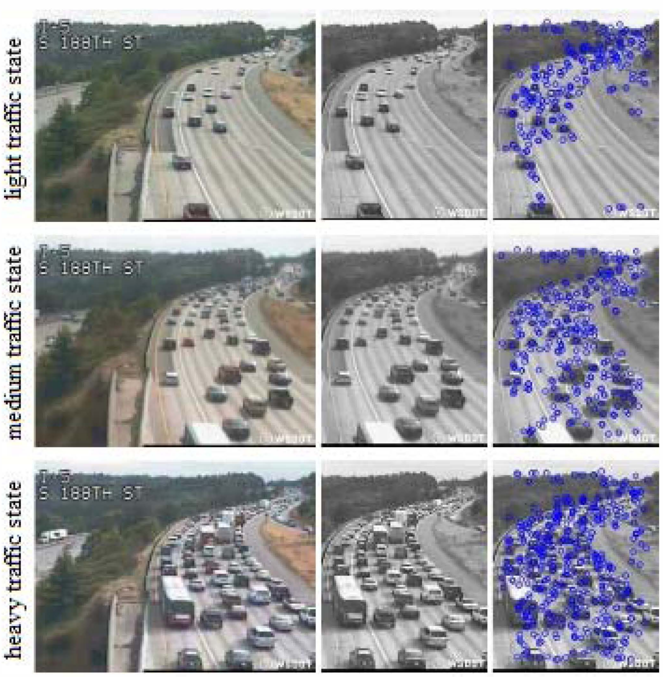

Fig. 3. Key-points detection in different traffic states, (a) traffic image, (b) ROI and (c) detected SURF key-points in ROI

B. Dynamic Features

Dynamic features describe the traffic density based on changes of the video scene in a given time duration. Since the level of traffic density affects the speed of vehicles and changes of the scene is directly related to the vehicles speed, the amount of scene changes from one frame to another can define the level of traffic density.

We propose two approaches to calculate the dynamic features. These methods are explained in Fig. 4. In this figure, f_i is *i*th frame of the video and $F_i^{'}$ shows the "difference-image" of two sequential frames *i* and *i*-1. The distance between f_i and f_{i-1} in a video sequence is not necessarily one frame and should be obtained by experiment to achieve the best result. V_i and $V_i^{'}$ represents static and dynamic feature vectors, respectively. According to Fig. 4, in method (a), static features are calculated from each individual frames. Then, difference of these features in a set of frames is calculated and considered as a dynamic feature. In method (b), the difference of two frames is obtained directly and dynamic features are extracted from this difference-image.

a) Pixel Comparison

The simplest feature that can be obtained from a video is "pixel comparison" [16]. Comparison of corresponding pixels in successive frames shows the variation of illumination or color in them. However, simple comparison of pixels in consecutive frames cannot differentiate between a big change

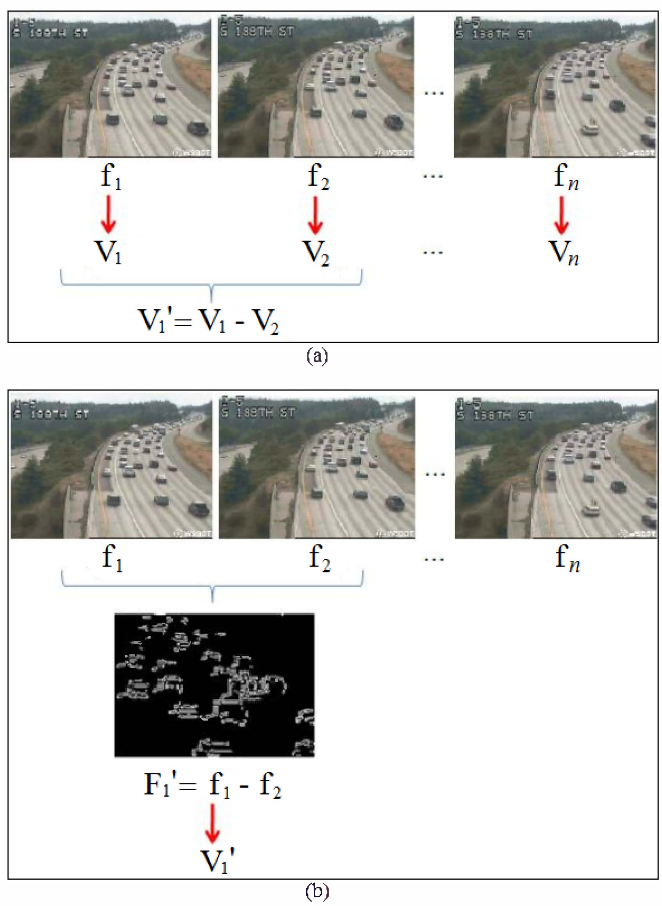

Fig.4. Two methods for calculating dynamic features

in a small area of an image and a small change in a big area of that image. One way to obtain a better feature without this drawback is to consider the pixel differences that are more than an experimental threshold. Equations (1) and (2) show how pixel comparison feature is computed.

$$DP(i, i+1, x, y) = \begin{cases} 1 & if\ |P_i(x,y) - P_{i+1}(x,y)| > T \\ 0 & otherwise \end{cases} \quad (1)$$

$$D(i, i+1) = \frac{\sum_{x=1}^{X}\sum_{y=1}^{Y} DP(i, i+1, x, y)}{XY} \quad (2)$$

First, based on Eq. (1), the difference of color or illumination (P) of two corresponding pixel in (x, y) location of frames is computed and compared with threshold T. Then based on Eq. (2), differences in all frame pixels that are more than T are added up and normalized by the area of frame (here XY).

b) Edges of difference-image

One feature which can describe the traffic state is "edges of difference-image". In order to obtain this feature, Canny operator [17] is applied on difference-image based on Eq. (3). In this equation, Φ shows the Canny operator.

$$DE_i = \Phi(|f_{i-1} - f_i|) \quad (3)$$

Number of edge points in difference-image is related to two parameters: first, number of vehicles in the scene and second, scene changes or vehicles speeds in consecutive frames. In light traffic density, the former is low and the latter is high. Heavy traffic density has opposite situation in which the number of vehicles is high but the scene changes are low. In medium traffic density, however, the number of vehicles is higher than light traffic condition and the scene changes is more than high traffic density. Therefore number of edge points is a distinctive feature to distinguish medium traffic state from two other states. This is illustrated in Fig. 5. In each row of this figure, one of the traffic states is depicted. The number of edge points extracted by Canny operator in light, medium and heavy traffic state is 704, 2142 and 1577, respectively.

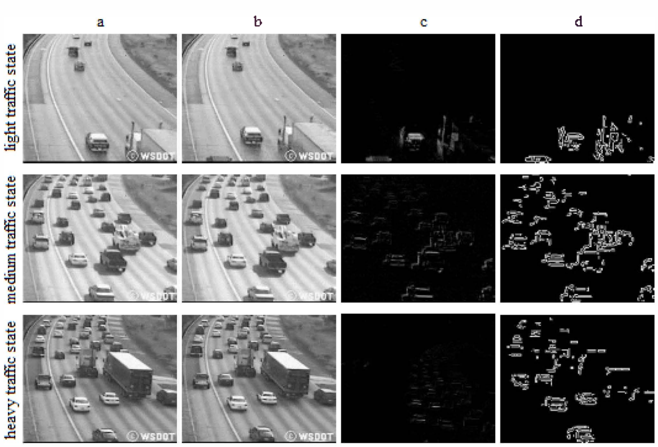

Fig.5. Edge points extraction in different traffic states, (a) previous frame, (b) current frame, (c) difference-image and (d) edges of difference-image

c) Moving edges

"Moving edges" is the next dynamic feature that is calculated based on edges of difference-image. Moving edges are edges which move through sequential frames and are effective in changes of the scene. We investigate the capability of this feature to distinguish between different traffic states.

Suppose that edges of current frame are shown by $E_i = \{e_1 \ldots e_k\}$. E_i is the set of all edge points extracted by Canny operator from the i^{th} frame. $ME_i = \{m_1 \ldots m_l\}$ is the set of l moving edges points that $l \leq k$ and $ME_i \subseteq E_i$. Moving edges could be either on edges or inside of moving objects. Moving edge points occurred by frames changes are obtained via choosing all pixels of E_i which are placed in the small distance of T_{change} from DE_i points (obtained via Eq. (3)) as follows:

$$ME_i^{change} = \{e \in E_i \mid \min_{x \in DE_i} \|e - x\| \leq T_{change}\} \quad (4)$$

Moreover, moving edges of previous frame can be used to obtain moving edges which are temporarily still. These temporarily still moving edges are calculated as follows:

$$ME_i^{still} = \{e \in E_i \mid e \notin E_b, \min_{x \in ME_{i-1}} \|e - x\| \leq T_{still}\} \quad (5)$$

where E_b shows the edge points of background.

Final moving edges are the union of moving edges in the frame and temporarily still edges [18]:

$$ME_i = ME_i^{change} \cup ME_i^{still} \quad (6)$$

d) Discrete Cosine Transform (DCT) coefficients histogram of difference-image

The "DCT coefficients histogram of difference-image" is another feature that we calculate. Discrete cosine transform is a method of transforming a signal to its frequency components [19]. For a two dimensional $m \times n$ signal s such as an image, discrete cosine transform S is calculated based on Eq. (7).

$$S(u,v) = \frac{2}{\sqrt{nm}} C(u)C(v) \sum_{y=0}^{m-1}\sum_{x=0}^{n-1} s(x,y) \cos\frac{(2x+1)u\pi}{2n} \cos\frac{(2y+1)v\pi}{2m}, \quad (7)$$
$$u = 0,\ldots,n, v = 0,\ldots,m$$

Where:

$$C(u) = \begin{cases} 1/\sqrt{2} & u = 0 \\ 1 & otherwise \end{cases}$$

For more information about this feature refer to [19]

e) Changes of color histogram

Since the color of vehicles are random, color histogram information does not use in many traffic applications. Nevertheless, the "changes of color histogram" in consecutive frames of traffic video can be used as a criterion of scene changes. To calculate this feature, the 256 dimensional RGB color space is reduced to 64 dimensions and then the color histogram is obtained. After that, Bhattacharyya distance is used to calculate changes of color histogram. Bhattacharyya distance is calculated based on Eq. (8) [20].

$$d(H_1,H_2)=\sqrt{1-\frac{1}{\sqrt{H_1H_2N^2}}\sum_i\sqrt{H_1(i).H_2(i)}} \quad (8)$$

where H1 and H2 are two histograms and N is the number of histogram bins.

As mentioned before, changes of color histogram is a distinctive feature to distinguish between different traffic states. This is illustrated in Fig. 6. In each row of this figure, one of the traffic states is depicted. Bhattacharyya distance of color histograms in light, medium and heavy traffic state is 0.1159, 0.1756 and 0.2549, respectively.

III. EXPERIMENTAL RESULTS

To evaluate our proposed approach using different introduced features, the classification accuracy should be tested over publicly available dataset. For the traffic density classification, an appropriate dataset has been provided by the University of San Diego, statistical visual computing lab [21]. This dataset consists of some video sequences of daytime highway traffic in Seattle, Washington. Three classes of traffic densities (light, medium and heavy) in different weather conditions (rainy, overcast and clear) are covered. Some example of this video frames are shown in Fig. 7. Every video has 42 to 52 frames of size 320×240 pixels. These videos are manually labeled based on three classes. Table I summarized some information about this dataset.

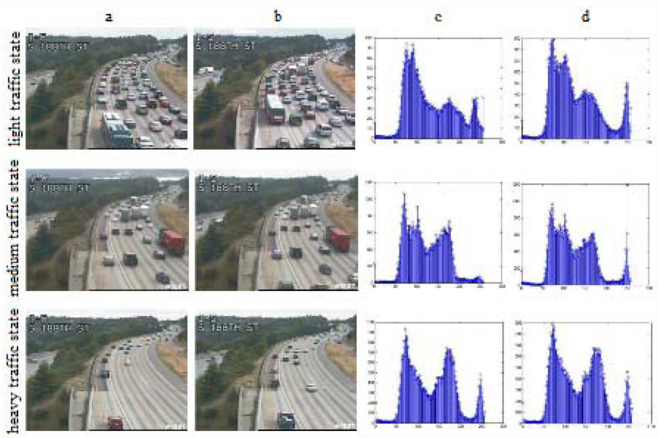

Fig.6. Changes of color histogram, (a) previous frame, (b) current frame, (c) color histogram of current frame and (d) color histogram of previous frame

Fig. 7. Some example of video frames from University of San Diego dataset, first row: different traffic sates, second row: different weather conditions

TABLE I. Properties of the University of San Diego dataset

Total number of videos	134
Total number of frames	6989
Number of videos for training	105 (78.35%)
Number of video for testing	29 (21.65%)
Clear	27 (20.93%)
Rainy	18 (13.95%)
Overcast	89 (68.99%)

Simulations were performed on PC with Intel Core™ i3-2100 CPU 3.10 GHz. Table II shows the classification accuracy and the time taken for feature extraction on the University of San Diego dataset. As it can be seen in this table, the highest classification accuracy is obtained equal to 82.7% by using “number of key-points”, “edges of difference-image” or “DCT coefficients histogram of difference-image”, which the first one is faster.

There are two other works that performed traffic density classification on the University of San Diego dataset. These are the works in [3] and [5] which both of them used dynamic texture [6] as the feature and SVM as the classifier. The classification accuracy of 93.3% and 95.3% were achieved in [5] and [3], respectively. Although, the classification accuracy of these works is more than ours, duo to the high computational cost of calculating dynamic texture, their approach cannot be employed in real time applications.

We also experimented our proposed approach on some real videos from Tehran roads. The video file was captured at 25 frames per second, with a frame size of 360×288 pixels. The experiments were conducted in two cases: (i) lighting condition is suitable in both training and test videos; (ii) lighting condition is suitable for training videos but not suitable for test videos. Fig. 8 shows some example frames of these two cases. The results of classification accuracy and time taken for features extraction are reported in Tables III. According to table III, the highest classification accuracy in suitable training and test lighting condition is obtained equal to 98.9% by using “edges of difference-image”. This feature has also high robustness against the changes of environmental conditions (e.g., lighting). Moreover, in comparison with other features, the time needed to extract this feature is very low.

TABLE III. Classification results using different features on the University of San Diego dataset

Type	Feature	Classification Accuracy (%)	Time (ms)
Static	Edge histogram	79.3	618
	Texture	71.4	1840
	Number of key-points	82.7	371
Dynamic	Pixel Comparison	65.6	500
	Edges of difference-image	82.7	549
	Moving edges	68.9	552
	DCT coefficients histogram of difference-image	82.7	2562
	Changes of color histogram	58.6	1833

Fig. 8. Some example of video frames from Tehran roads, first row: suitable training and test lighting condition, second row: suitable training and not suitable test lighting condition

TABLE IIIII. Classification results using different features on Tehran roads

Type	Feature	Classification Accuracy (%) (suitable training and test)	Classification Accuracy (%) (suitable training and not suitable test)	Time (ms)
Static	Edge histogram	98.4	75.8	2362
	Texture	94	79.2	28935
	Number of key-points	97.3	69.3	353
Dynamic	Pixel Comparison	94	63	609
	Edges of difference-image	98.9	80	658
	Moving edges	88.6	56	802
	DCT coefficients histogram of difference-image	88.7	85.8	8616
	Changes of color histogram	78.2	71.3	85526

It can be concluded that the best feature for traffic density classification based on our approach is “edges of difference-image”, as it leads to the highest classification accuracy on both datasets with high robustness against the changes of environmental conditions and low extraction time.

IV. Conclusion

In this paper, we presented an approach to classify traffic videos based on feature extraction and support vector machine. For this purpose, a variety of new efficient features were proposed for distinguishing between different traffic states. The proposed approach gives a holistic view of traffic density without any need to vehicles detection and tracking. Experimental results on two traffic datasets showed that our proposed approach classifies accurately the traffic states by using efficient features and it is robust to changes of environmental conditions.

References

[1] J. Lee and A. C. Bovik, "Estimation and analysis of urban traffic flow," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 1157-1160.
[2] Y.-K. Jung, K.-W. Lee, and Y.-S. Ho, "Content-based event retrieval using semantic scene interpretation for automated traffic surveillance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 2, pp. 151-163, 2001.
[3] K. G. Derpanis and R. P. Wildes, "Classification of traffic video based on a spatiotemporal orientation analysis," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, 2011, pp. 606-613.
[4] B. Maurin, O. Masoud, and N. P. Papanikolopoulos, "Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds," *Robotics & Automation Magazine, IEEE*, vol. 12, pp. 29-36, 2005.
[5] A. B. Chan and N. Vasconcelos, "Classification and retrieval of traffic video using auto-regressive stochastic processes," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, 2005, pp. 771-776.
[6] http://www.vision.jhu.edu/dynamicTextures.htm.
[7] Z. Li, E. Tan, J. Chen, and T. Wassantachat, "On traffic density estimation with a boosted SVM classifier," in *Digital Image Computing: Techniques and Applications (DICTA), 2008*, 2008, pp. 117-123.
[8] T. Wassantachat, Z. Li, J. Chen, Y. Wang, and E. Tan, "Traffic Density Estimation with On-line SVM Classifier," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, 2009, pp. 13-18.
[9] X. Li and F. M. Porikli, "A hidden Markov model framework for traffic event detection using video features," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 2004, pp. 2901-2904.
[10] E. Tan and J. Chen, "Vehicular traffic density estimation via statistical methods with automated state learning," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007, pp. 164-169.
[11] R. C. G. a. R. E. Woods, "Digital Image Processing Introduction," 2002.
[12] C. S. Won, D. K. Park, and S.-J. Park, "Efficient use of MPEG-7 edge histogram descriptor," *Etri Journal*, vol. 24, pp. 23-30, 2002.
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91-110, 2004.
[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, pp. 346-359, 2008.
[15] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*, ed: Springer, 2006, pp. 430-443.
[16] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, vol. 16, pp. 477-500, 2001.
[17] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 679-698, 1986.
[18] C. Kim and J.-N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, pp. 122-129, 2002.
[19] A. B. Watson, "Image compression using the discrete cosine transform," *Mathematica journal*, vol. 4, p. 81, 1994.
[20] S. Dubuisson, "The computation of the Bhattacharyya distance between histograms without histograms," in *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on*, 2010, pp. 373-378.
[21] http://www.svcl.ucsd.edu/projects/traffic/.