IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 10, 2019, accepted October 26, 2019, date of publication November 7, 2019, date of current version November 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952168

CMRS: A Classifier Matrix Recognition System for Traffic Management and Analysis in a Smart City Environment

FAYEZ ALQAHTANI[1], ZAFER AL-MAKHADMEH[1], OMAR SAID[2,3], AND AMR TOLBA[1,2]

[1]Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[2]Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-Kom 32511, Egypt
[3]College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia

Corresponding author: Zafer Al-Makhadmeh (zalmakhadmee@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University through the Research Group under Grant RG-1439-088.

ABSTRACT The application of the Internet of Things (IoT) in a smart city improves its efficiency in terms of communication and installation costs by scaling geographical distance through intelligent devices and digital information. Different applications in a smart city, including health care, road safety, industry and home automation, rely on the IoT. Considering the significance of the IoT in smart city road applications, this manuscript introduces a classifier matrix recognition system (CMRS) for improving real-time traffic optimization. This classifier matrix system performs an independent and matching analysis of the real-time traffic images and constructs a decision factor for deriving its conclusion. The conclusion is served as responding notifications through the connected IoT systems for the users employing roadside-communication–assisted applications. CMRS exploits the advantages of block classification and matrix operations for improving the correlation accuracy and similarity index. The experimental results indicate that the proposed CMRS improves correlation accuracy with a high similarity index and less processing time and dissimilarity rate.

INDEX TERMS Block classification, image recognition, IoT, ITS, smart city, traffic management.

I. INTRODUCTION

The rapid development of the IoT is resulting in its services being used in different real-world applications. This communication platform is capable of connecting various elements, including humans, computers, machines, tiny sensors and other objects [1], [2]. More specifically, the connected devices are heterogeneous in nature, which facilitates pervasive access to resources, applications and services [3], [4]. The intelligence of the communicating devices in this paradigm provides anywhere and anytime access to resources the end user requests [5]. Because the interconnection of this platform incorporates heterogeneous devices, it supports various communication technologies ranging from device-to-device (D2D) to distributed cloud computing [6]. The independent operations of the devices are assisted via a wireless communication medium in which the protocol and technology are adopted appropriately. Distributed and concurrent access, mobility support, scalability and flexibility, interoperability and real-time responses are some of the benefits the IoT provides for end users. Real-time applications of the IoT include health care, industrial automation, smart homes, roadside-driving– assisted communication, traffic management and smart cities [7].

A smart city environment exploits the complete development of the IoT and other reliable technologies such as Intelligent Transportation Systems (ITS) to improve end-user assistance and communication comfort [8], [9]. Real-time automated traffic management is a recognizable smart city application that improves road-safety and driving cautiousness for end users [10], [11]. Automated traffic management and congestion avoidance on roads are feasible through sensing and activating devices that are connected through the internet [12], [13]. The sensing device can monitor and capture instances on the roadside and transmit them through IoT-enabled devices to the processing center [14], [15]. With the help of advanced processing and analyzing systems, the input is processed, and the response is activated to the

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Chen.

users. The user's equipment that is connected to the communication environment receives the activated output as notification or assistance (as in Google maps) [16], [17]. The inputs of these sensors are usually signals; the advanced features of monitoring devices help share images with the processing center. Images are extracted from a dynamic motion-concentric environment, from which the device grabs notable changes/events as input by the device [18], [19]. The radio-unit that is connected with the device acts as a port for transmitting the sensed input to a processing center that is located far away. Such applications of the IoT assist in constructing a reliable and well-connected service infrastructure in a smart city environment [20], [21].

Computer vision [22] is an emerging technology in which the machines are programmed to react intelligently by analyzing digital images. The machines activate to the inputs by gaining useful information from the images or videos. This kind of technology is popular in robots, object detection and driverless cars. This technology is integrated with the IoT in smart city environments for analyzing digital image inputs [23]. In particular, in the case of traffic management, the image inputs from the sensing/monitoring units are analyzed through sophisticated techniques. Processing centers employ computer vision for processing and activating the digital images. Computer vision analyzes the input image/video by extracting its features, classifying elements and processing binary and digital features [24]. The processing features of this field inherit the advantages of IoT and other computing devices for augmenting the reliability of robust smart city solutions. Computer vision is employed in digital image processing, surveillance system actuation, road-traffic monitoring, medical image diagnosis and robotic systems [25]. The contributions of the paper are as follows.

- We design a classifier matrix recognition system for analyzing the roadside traffic images fetched from surveillance systems.
- We design a level-based image analysis method for reducing errors in processing and performing a correlation-based recognition for accurate road-traffic information retrieval.
- We perform a comparative study of the proposed method with the existing image processing techniques using different metrics.

The organization of the manuscript is as follows. In Section 2, the works related to image processing, the IoT in traffic management and its applications are presented with a detailed description. The introduced CMRS is discussed in Section 3, with its different levels and functions. Section 4 describes the comparative study along with the different metrics and previous methods considered. Section 5 concludes with the findings and a summary.

II. RELATED WORKS

Lee and Kim [26] employed a convolution neural network (CNN) for detecting roadside traffic signs. The two-dimensional texture and shape of the detected image are processed using CNN to identify its edges. The boundary of the image is detected by projecting the stored image plane into the input image plane. This traffic detection method achieves high accuracy irrespective of the mobile platform. A cooperative edge computing framework was designed by Long *et al.* [27] for video analysis in IoT-assisted multimedia systems. This framework employs both greedy and heuristic solutions for addressing suboptimal image matching and quantization problems. The winner determination problem in this framework mitigates the subgroup image matching problems due to resource starvation. These jointly help improve the accuracy of human detection.

Li *et al.* [28] discussed the analysis of multispectral and panchromatic images with the help of a fusion process in a multi-sensor IoT environment. This fusion method helps mitigate the distortions in the image due to different spectrum ranges acting in the same image. Superpixel segmentation and spectral weight distribution methods are adopted to control the errors in analyzing degraded images. Unlike the work in [28], Rui *et al.* [29] introduced the concept of deep learning for analyzing multispectral image analysis. In an IoT environment, the analyzing device is trained with the processed image using deep learning. The multispectral fusion method relies on the output of deep learning for reducing detection reliability, power-concentric analysis and degradation of the images.

Arinaldi *et al.* [30] exploited the computer vision technique for identifying and classifying vehicles from the observed traffic videos. The computer vision technique relies on a recurrent convolution neural network for estimating the properties of the road and vehicle density. The accuracy of detection is improved through quantitative and qualitative analysis for the vehicle count, its type and classification. Wang *et al.* [31] introduced adaptive boosting support vector machines (AB-SVM) for identifying events by analyzing large-scale traffic data. AB-SVM is a hybrid automatic incident classification method in which the spatiotemporal form of the input signals is analyzed for detection. Communication filters, association-based comparison and SVM for training data are successfully employed for reducing noise, classifying events and real-time data coverage in this method. However, this method fails to achieve better accuracy in correlating large datasets.

Zhang [32] projected a geometric discriminative feature fusion (GDFF) method for analyzing traffic images. This method uses the input data to train the CNN to extract image-related features. RGB histograms and discriminative features of the images are classified using the extracted features. The images are segmented and are represented using a region connected graph to identify the connection between the features and the training sets. This method is adaptive to different topologies with less feature variances. The less feature variances create an impact over the processing time of the image because the connectivity of the graph is less. Lu *et al.* [33] presents visual attention model-based traffic

signal identification and categorizing. The attention model categorizes the signal by differentiating the image into multiple candidate regions to identify small targets. This detection model is useful in improving the accuracy of the traffic signal image with fewer distortions.

Bai *et al.* [34] introduced a novel vehicle identification and categorization method using deformable part-based models. In this model, the tagged vehicle images are used to form the training dataset. The traffic input is then correlated with the trained dataset to identify the event in a precise manner. Conventional image processing techniques such as crop and scaling are applied for the correlating image to match the processed data, and the vehicle is categorized using SVM. This method is better at improving the accuracy and sensitivity of the analysis. Hu *et al.* [35] introduced a histogram of multiscale block local binary pattern (HMBLBP) for exploiting the texture features for distinguishing traffic density. In this model, the image is split into different cells from which the region of interest (ROI) is identified for estimating the pixels in a temporal manner. Similar to the work introduced in [31] and [32], this model also employs SVM for distinguishing the traffic patterns. This method achieves a better true positive rate and distinguishing accuracy. The true positive is stabilized for limited and stored images, which reduces the rate of similarity index.

Multi-class SVM is introduced by Ruiqi *et al.* [36] for recognizing traffic patterns from the sensed roadside images. SVM classifies the road congestion by considering the density of the vehicles. The quadratic discriminant analysis is performed over the classified data for pattern matching and identifying the appropriate event. This method aids reliable road planning and traffic management quality in smart cities. Ahn *et al.* [37] introduced an image distribution method over IoT devices that addresses high-resolution image stitching issues. The load of the images is distributed to the cloud, from which the homographic information is fetched for exchange. A feature point dependent extraction method augments distribution by reducing the processing time in image stitching. Furthermore, this method achieves less complexity and supports better image generation. Jiang *et al.* [38] presented fuzzy-logic–based quality of service optimization for handling visual information in IoT environments. This optimization follows fuzzy correlation for analyzing visual information. The fuzzy process resolves the information, and then defuzzification integrates the features of the fuzzy process into a single solution to provide better quality of service.

This survey discusses information distribution, processing and analysis for detecting events concerning roadside communication. Processing a visual image and correlating it with the stored information help in detecting and classifying events [31], [34], [35]. The analysis methods follow learning concepts [26], [31], [32] for distinguishing the images in a reliable manner. Unlike these methods, this article exploits the conventional image preprocessing technique for image analysis and event recognition to observe the changes from smaller to complex levels of analysis. Block-based analysis followed by correlating feature extraction and recurrent analysis are the novel schemes for event recognition in this proposed method. The simplest form of image validation that improves accuracy and reduces error is surveyed in [35].

III. CLASSIFIER MATRIX RECOGNITION SYSTEM

CMRS is designed to analyze real-time roadside images and to report the traffic conditions as notifications. The analysis of the current roadside live images is performed as a part of monitoring and streamlining traffic in smart city environments. The notification services in the end-user application assist better navigation, object detection and traffic forecasting. Therefore, the advantages of live image analysis are manifold for commercial, individual and other telecommunication services. The prominent part of this analysis is the non-similarity of the images with respect to time-domain and age-domain.

In a time-domain, a single live image varies with its density of elements at different time instances (similar to road traffic observed at different frequencies during a day). In an age-domain, the elements and structure of the live image vary with time (prolonged time). CMRS is designed to estimate and detect the type of live image using a comparative analysis of the stored image. This analysis represents the correlation of different images over time. Correlation is performed between stored and processed input images. The difference in images is estimated by verifying the structural block of the live image through matrix resolution and recurrent analysis. Therefore, the process of CMRS is divided into four levels: block conversion, matrix resolution, and recurrent analysis and recognition. Figure 1 represents the schematic view of the levels in processing live images transmitted through IoT devices.

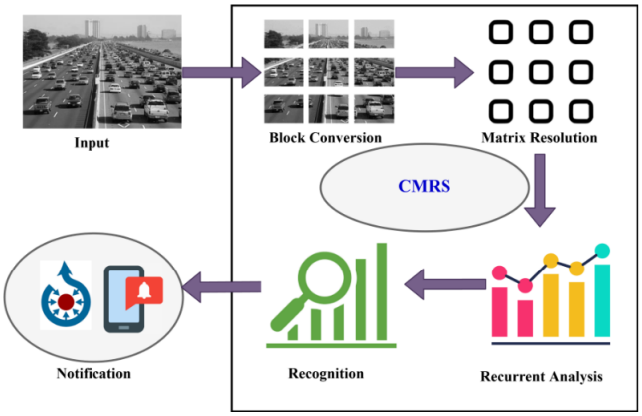

FIGURE 1. Schematic view of CMRS.

A. LEVEL 1: BLOCK CONVERSION

The input from the roadside monitoring device is received as an image that is validated for further analysis. Block conversion is performed for improving the accuracy of image processing and reference matching. At a block conversion level, the image is spilt into individual blocks that are useful in processing. The word ''useful'' refers to the simplest way of image analysis by extracting features and information from

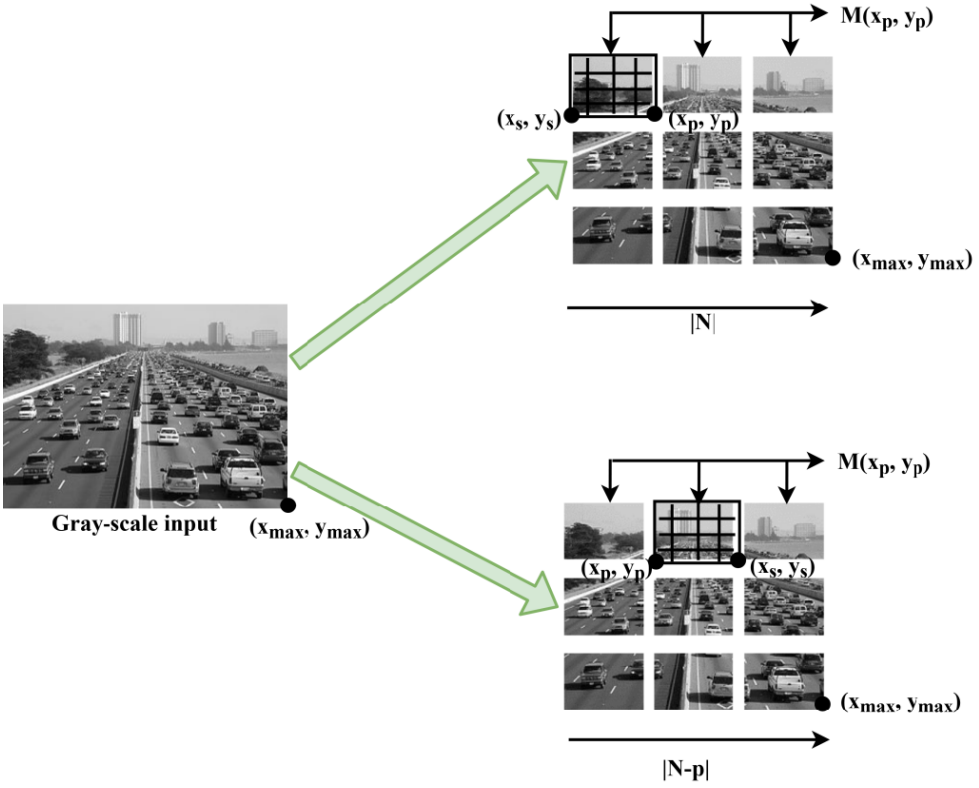

FIGURE 2. Initial and final pixel position determination.

the input. The usefulness is determined based on the accuracy in processing the image and verifying it at the time of correlation. Block-by-block processing prevents unnecessary errors in image handling and pattern estimation. Unnecessary errors include bit tampering, transmission and broken errors. The reasons for these errors include the wireless medium, transmitter-receiver malfunctioning, connectivity issues and malicious users. A computer vision visualizes images as a binary input of 0's and 1's. Extraction of overlapping images as blocks is the fundamental measure in block conversion. The RGB images are converted to gray scale for ease of analysis and processing.

The image consists of two layers: background and foreground. The block conversion involves the segmentation of a complete gray-scale image into N-blocks. The even block sizes are discarded because the input image size and texture vary with time and age-domain. Let (x_p, y_p) denote the pixel positions of an image, the maximum dimension of which is (x_{max}, y_{max}) region. The converted image is divided into $(x_{max}, y_{max})/N$ blocks for which the initial and final positions of the (x_p, y_p) are to be determined. With respect to (x_p, y_p), the mean pixel range is estimated using Equation (1).

$$M\left(x_p, y_p\right) = \frac{1}{2}\sum_{i=1}^{N} |\gamma\left(x_p, y_p\right) - \gamma(x_N, y_N) \tag{1}$$

where, $M(x, y)$ is the mean pixel range, $\gamma(x_N, y_N)$, is the concentration of pixels' (1's and 0's) in (x_p, y_p) positions, and (x_N, y_N) represents the pixel concentration in N^{th} position. The dividing points $(x_s \; and \; y_s)$ are estimated as

$$\left.\begin{aligned} x_s &= \frac{\sum\limits_{(x_N, y_N)} M(x_p, y_p)}{N} \\ y_s &= \frac{\sum\limits_{(x_N, y_N)} M(x_p, y_p)}{|N - p|} \end{aligned}\right\} \tag{2}$$

Equation 2 frames the required points (x_s, y_s) for the block segregation. Now, this pair $(x_s, y_s) \in x_{max}, y_{max} \forall (x_p, y_p)$ is to be determined as the initial/final positions of the block. Let β represent $\max\left\{\left(x_p, y_p\right)\right\} \forall p = 1, 2, \ldots, N$, then *if* $M\left(x_p, y_p\right) \geq \beta$, *then* (x_s, y_s) is the initial position, and *if* $M\left(x_p, y_p\right) < \beta$, *then* (x_s, y_s) is the final position of the block. Therefore, the block positions are (x_s, y_s) and $\left(x_p, y_p\right)$ if the $M\left(x_p, y_p\right) \geq \beta$. On the other hand, the $\left(x_p, y_p\right)$ and (x_s, y_s) pair represents the final position if $M\left(x_p, y_p\right) < \beta$. The initial and final positions of the pixels are fixed as determined using Equation (2). This is illustrated in Figure 2.

B. LEVEL 2: MATRIX RESOLUTION

In this level, the extracted blocks are converted as an $N \times N$ matrix of all the pixels from 1 to N. The blocks that are segregated as per (x_s, y_s) and $\left(x_p, y_p\right)$ pairs are denoted as a covariance matrix of size $N \times N$. The covariance matrix

element $c_i, i = 1, 2, \ldots, N$ is computed as:

$$e_i = \frac{1}{N-1} \sum_{j=1}^{N} \left[(\alpha_j - p_j)^T \right] \tag{3}$$

where α_j is the mean of the block between (x_s, y_s) and (x_p, y_p). The covariance matrix elements must satisfy the condition of $e_i e_i^T = 1$, i.e., the matrix for $e_i \forall (x_s, y_s)$ to (x_{max}, y_{max}) of all N must achieve a maximum value of 1. This maximum value of 1 implies that the multiplicative covariance matrix of the input and stored image is 1. This covariant is a combination of the normal and transposed matrices of the images.

Resolving the matrix into a simplex form is preceded by analyzing the similarity between the extracted blocks. In this scenario, the mapping of the stored image is mandatory. The covariant matrix of the stored image is estimated by identifying its elements (e_{si}) as:

$$e_{si} = \frac{1}{N \times N} \left[\sum_{x=0}^{x_{max}} \sum_{y=0}^{y_{max}} (x, y)^2 \right] - \sum_{i=1}^{N} \alpha_{\frac{si}{N}} \tag{4}$$

where $\alpha_{\frac{si}{N}}$ is the mean of the stored image blocks from $i = 1$ to N^{th} pixel. The matrix of both the elements is equated in their simplest form as $e_i e_i^T = 1$. The expanded matrix is represented as:

$$\begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1N} \\ e_{21} & e_{22} & & e_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ e_{p1} & e_{p2} & \ldots & e_{pN} \end{pmatrix} \begin{pmatrix} e_{s11} & e_{s21} & \ldots & e_{spN} \\ e_{s12} & e_{s22} & & e_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ e_{s1N} & e_{s2N} & \ldots & e_{spN} \end{pmatrix} = 1 \tag{5}$$

The transpose of the e_{si} elements is represented in Equation (5). This matrix achieves a unit value provided $e_{pN}(\sum_{i=1}^{N} e_{spi}) = 1$. Considering the two individual elements e_{spN} and e_{si}, the covariance matrix results in unity if these conditions are satisfied.

Condition 1: The range of pixels $p = 1$ to N if it is found to be the same in both the stored and the input image, and then the dissimilarity (Δd) is zero.

Analysis 1: Consider two images (input and stored) that are framed for covariance estimation as represented in Equation (5). The simplest verification in the condition is done as follows for the two elements e_{PN} and e_{sPN} as using ∇d as:

$$\nabla d = \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} e_{ij} - \alpha_i^2 \right] - \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} e_{ij}.e_{sij} - (\alpha_i.\alpha_{si}) \right] \tag{6}$$

The estimated $\nabla d \in$ the set where either α_j or $\alpha_{si} \neq 0$. This indicates that the mean is appropriately estimated where no tampered/missing/pixels are found in both the (x_s, p_s) and the (x_p, y_p) pair. Therefore, the difference between $e_{ij} - \alpha_i^2$ tends to zero; if α_i or $\alpha_{si} = 0$, then $\nabla d = 0$ for any N. Contrarily, if either of e_{ij} or $e_{ij} \neq 0$, then $\nabla d \neq 0$. If $\nabla d < 0$, then the mean of the input image is estimated incorrectly for all $e_{ij} < e_{sij}$. This means the image consists of tampered/overlapping pixels from which the information gain is less. The information from these pixels are not completely discarded if it is analyzed using the least features extracted from the blocks (see the next subsection).

Condition 2: This conditional analysis is different from Condition 1. The conversion results in a dissimilar number of blocks, i.e., N and $|N - p|$.

Analysis 2: The dissimilar number of blocks between the input and stored image results in non-uniform covariance matrix resolution. In such a case, the input image is scaled to satisfy Equation (5) with a scaling function. The scaling-factor–based matrix is represented as:

$$\left.\begin{array}{c} e_i.S_f = e_{si} = 1 \\ (i.e.)\left[e_i.S_f\right]\left[e_{si}\right]^T = 1 \\ (or) \\ \nabla d = \left(e_i.S_f\right) - \left(e_i S_f\right).e_{si} \end{array}\right\} \tag{7}$$

In Equation (7), the scaling factor elements are represented in which the transpose of the stored image covariance matrix is considered for unity. The dissimilarity here is estimated as an element of the scaling factor matrix for balancing the erroneous pixels.

The matrix s_f is the scaling factor that is a similar representative of the e_i matrix size. It consists of elements $[s_{f}11, s_{f}12, \ldots, s_{fNp,}]$ that reflect the concentration of the pixels in the blocks N or $|N - p|$. This means that the factors of $0'$s and $1'$s over each other are represented in this matrix. An example matrix is represented as below.

$$\begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1N} \\ e_{21} & e_{22} & & e_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ e_{p1} & e_{p2} & \ldots & e_{pN} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & \ldots & s_{f1N} \\ 0 & s_{f2N} & & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix} \times \begin{pmatrix} e_{s11} & e_{s21} & \ldots & e_{sp1} \\ e_{s12} & e_{s22} & & e_{sp2} \\ \vdots & \vdots & \cdots & \vdots \\ e_{s1N} & e_{s2N} & \ldots & e_{spN} \end{pmatrix} = 1 \tag{8}$$

Using Equation (8) as a reference, the dissimilarly is estimated as:

$$\nabla d = \begin{cases} \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} (e_{ij} - s_{fij}) \right] \\ - \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} (e_{ij} * s_{fij} - e_{sij}) \right], \; if \; e_{ij} > e_{sij} \\ \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} (e_{ij} + s_{fij}) \right] \\ - \left[\frac{1}{N \times N} \sum_{i=1}^{p} \sum_{j=1}^{p} (e_{ij} * s_{fij} - e_{sij}) \right], \; if \; e_{ij} < e_{sij} \end{cases} \tag{9}$$

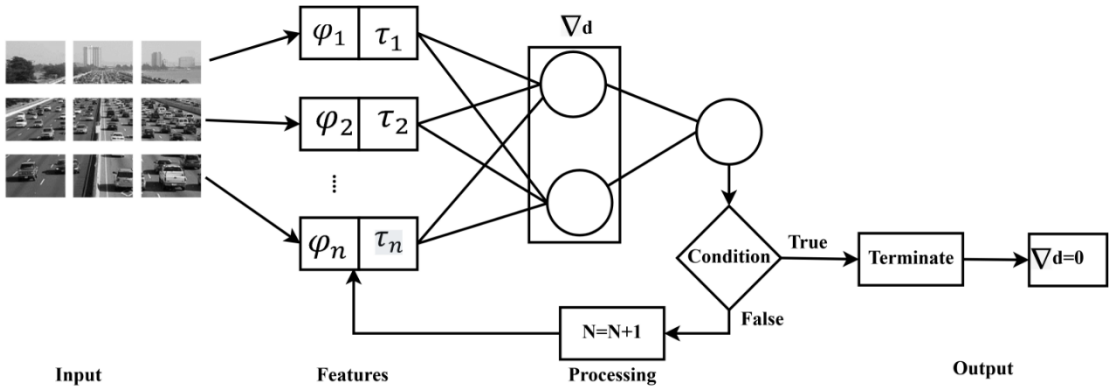

FIGURE 3. Recurrent analysis.

The dissimilarity is considered for both overlapping pixels ($e_{ij} > e_{sij}$) missing bits $e_{ij} < e_{sij}$ using Equation (9). The ∇d value is nevertheless zero; rather, it is very small, so it can be discarded.

Discarding/accepting ∇d relies on the useful information extracted from the image features. If the features are useful through a series of block evaluations, then ∇d from Condition 2 is discarded else it decreases the similarity of the image. The process of feature extraction and recurrent analysis is presented in the next subsection.

C. LEVEL 3: RECURRENT ANALYSIS

In a recurrent analysis process, the converted block is analyzed for the features extracted to gain possible information. For this analysis, machine learning that accepts the matrix resolution is incorporated. Unlike the conventional machine learning approaches, the recurrent analysis is halted if useful information is gained. The occurrence of the useful information is verified if $\nabla d = 0$ at any step of the recursive analysis process. The recurrence analysis follows two features, entropy and correlation, that are computed as in Equation (10):

$$\left.\begin{array}{c} entropy\,(\varphi) = -\sum_{i=1}^{p}\sum_{j=1}^{p}\rho(x_i, y_i)\forall(i,j) \in (x_{max}, y_{max}) \\ correlation\,(\delta) = \sum_{i=1}^{p}\sum_{j=1}^{p}(i,j)\,\rho\,(x_i, y_i) - \alpha_i, \alpha_j) \end{array}\right\} \tag{10}$$

For simple, complex-free image input, entropy is less, and correlation is high. Therefore, an image block with less entropy and high correlation compared to the shored image is said to have $\nabla d = 0$. The number of blocks of the image is said to vary depending on its size, texture and composition. The process of recurrent analysis is illustrated in Figure 3.

The block is analyzed for the features extracted from the converted image. For each feature, ∇d is analyzed in coherence to the condition of $min\,(\varphi)$ and $max(\delta)$ as a comparison with the stored image. If this condition, i.e., $\varphi \geq \varphi_s$ and $\delta \geq \delta_s$, is achieved, $\nabla d = 0$ else the block consists of dissimilarity as computed using Equation (6) for the same pixels. If the number of blocks is different, then ∇d is estimated as in Equation (9). Here, φ_s and δ_s denote the entropy and correlation of the stored image. If these conditions are not satisfied, the analysis is carried out for the next block. Therefore, the recurrent process is valid until the N blocks of the image. The $\nabla d \neq 0$ blocks are categorized for future reference as training sets. Similarly, the valid block information is reflected in the processing image at the time of storage for further analysis. The ∇d observed in each of the recurrent analysis step is estimated.

$$\left.\begin{array}{c} \nabla d_1 = e_{pN} - e_{pN}.e_{spN} \\ \nabla d_2 = \left(e_{pN} - \nabla d_1^2\right) - \left(e_{pN}.e_{pN}\right) + \left(\nabla d_1.\alpha_{s1}\right) \\ \vdots \\ \nabla d_N = \left(e_{pN} - \nabla d_{N-1}^2\right) - \left(e_{pN}.e_{pN}\right) + \left(\nabla d_{N-1}.\alpha_{sN-1}\right) \end{array}\right\} \tag{11}$$

In Equation (11), ∇d is estimated for the condition where the input and stored image have the same $p = 1$ to N. The ∇d for Condition 2 is also estimated in the same manner. The condition specified in Figure 3 represents the termination/continuity of the analysis process. Similarly, if the condition is satisfied in any of the rounds (between 1 and N), then the process is terminated. The extracted feature is used to detect the accuracy and similarity improvement of the images. The recurrent analysis helps improve the accuracy rate by reducing ∇d errors. Figure 4 presents the true positive rate corresponding to the analysis probability of the features. One feature denotes the consideration of φ (indicated in red), and two features represents both φ and δ (indicated in green).

The true positive rate of the images are estimated as the possibility of extracting useful information from analyzing The true positive rate of the images is estimated as the possibility of extracting useful information from analyzing the features φ and δ. In this analysis, the rate of analysis of a single feature results in less information extraction compared to the analysis of two features. Useful information is obtained if the features satisfy the given condition. Table 1 presents iterates and the true positive rate of the examined features.

The missing condition checks and true positive rate (TPR) denote that $\varphi \geq \varphi_s$ is not satisfied, and hence, the block is

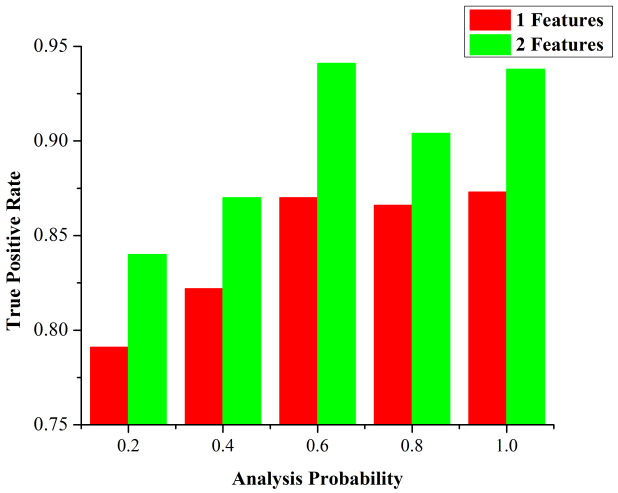

FIGURE 4. True positive rate analysis.

TABLE 1. Iterates and true positive rate.

Iterates	$\varphi \geq \varphi_s$	TPR	$\varphi \geq \varphi_s$ *or* $\delta \geq \delta_s$	TPR
50	41	0.713	38	0.827
100	69	0.744	78	0.84
150	103	0.756	139	0.87
200	147	0.743	176	0.836
250	169	0.86	197	0.904
300	174	0.84	206	0.941
350	-	0	87	0.614
400	59	0.83	138	0.938
450	-	0	106	0.743

discarded. Instead, verification of both the conditions generates a positive response with less TPR.

D. LEVEL 4: RECOGNITION

In the recognition process, the images are compared to deliver which event is detected and identified. The recognition is the mapping of image features through sparse random projection and coordinate matching. The pixel's positions (initial and final) of the processed image (blocks) are mapped with the extract positions of the stored image. The deviations and features are estimated for the block in reference to the stored image. If the block requires s_f, then ∇d is estimated for the block. The rate of variation must be less than ∇d compared to the stored image. The rate of variance (r_v) is estimated as:

$$r_v\,(e_i, e_{si}) = ||N - e_{si}.\nabla d_N|| \tag{12}$$

In Equation 12, the rate of variance is estimated for each of the N-blocks. The variable ∇d_N represented in Equation (12) is different from ∇d computed in Equations (6) and (9). Therefore, the following is adopted in recognizing the processed image with respect to the condition in Table 2.

TABLE 2. Condition and analysis for recognizing images.

Condition	Analysis	Tagging
$r_v.(e_i, e_{si}) \leq \nabla d$	$P = 1$ *to* N is the same	Events are the same
$r_v.(e_i, e_{si}) > \nabla d$	$P = 1$ *to* N is the same	Events are the same with more density
$r_v.(e_i, e_{si}) \leq \nabla d$	N and $\|N - p\|$	Events are the same identified with error
$r_v.(e_i, e_{si}) > \nabla d$	N and $\|N - p\|$	Dissimilar events

Table 2 is used to identify the type of event as correlated with the existing stored information. The condition and its analysis determine the type of event and the labeling (identification) of the event is termed as tagging. All the analysis conditions are based on the rate of variance between the dissimilarity and its difference between two successive blocks. The conditions are modeled between rate of variance and difference in dissimilarity for different analysis of pixel and block. The difference in pixel positions and blocks are the analysis conditions for the detecting the event.

The tagging is the name of the event labeled for the stored image. If the detected image is the same as the stored label, it is augmented to the processed image. This process is the same for any number of input images and its blocks.

IV. REAL-TIME SETUP

Figure 5 represents the real-time setup of a traffic monitoring system. The setup consists of three functional layers: monitoring, access and processing layers.

Monitoring Layer: This layer is responsible for fetching real-time inputs for analysis. Monitoring and surveillance systems (cameras) and speed sensors are fixed at road intersections and along the road path in a smart city environment. The cameras are connected through the internet for transmitting the images to the processing centers, which are located far away. The purpose of speed sensors is to detect the speed of the vehicles passing by the road/lane. The monitoring camera is programmed to capture live images of the road segment at periodic intervals. These images are useful in determining the congestion/traffic information of the roads.

Access Layer: The access layer consists of communication and distributed storage components. Gateways, access points and the cloud platform are the major components in this layer. The role of this layer is to enable communication between the monitoring and processing layers. It simply exchanges the information in a to-and-fro manner between the layers. Heterogeneous communication and different wireless technologies are accessed from this layer.

Processing Layer: The processing layer usually consists of a centralized control center. In the case of traffic management, the processing center is the central control authority (building) that monitors the roadside events through live images and recordings. This control center employs diverse techniques for analyzing and processing the received input from the

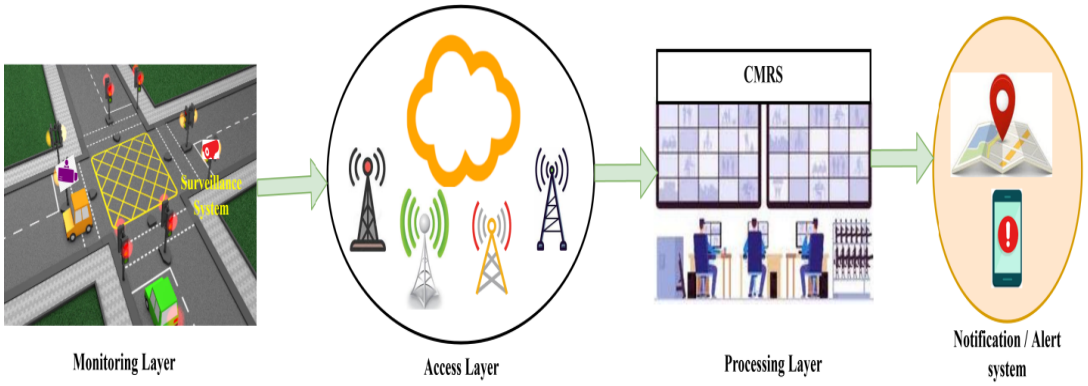

FIGURE 5. Real-time setup representations.

TABLE 3. Simulation setup.

Simulation Setup	Value
Processor	Intel i3
Clock Speed	1.8GHz
Images/sec	8
Stored Data Size	2.56 Gb
Stored Images	1751
Blocks	2-9
Maximum Iterates	1000
Physical Memory	2x4Gb

monitoring layer through the access layer. After processing, the received video/image input is stored for future reference with the event tag in the layer's local storage. The processed images are helpful in assisting driving users through notification or are integrated into the navigation system for alerts.

V. RESULTS AND DISCUSSION

The performance of the introduced CMRS is verified through simulations using an openCV tool [39], in which random traffic images are used for analysis. The images of different dimensions and pixels are collected from open sources and are verified through the level-based analysis procedure. In Table 3, the simulation setup is presented.

For a brief analysis, the outcome of CMRS is compared with the existing AB-SVM, GDFF and HMBLBP for the metrics' accuracy, processing time, dissimilarity rate and similarity index, respectively. These methods are considered for comparison to perform processing with a correlation analysis. Therefore, in this analysis, the similar methods are compared because the proposed method follows the same process. The images used in this method are processed using the methods specified in [31], [32] and [35] for achieving a unanimous analysis. The experimental results are verified for the impact of iterates and blocks classified in CMRS. The metrics' accuracy and processing time are measured with respect to the iterates, and dissimilarity rate and mean accuracy are assessed with respect to blocks.

A. IMPACT OF VARYING ITERATES

The varying number of iterates increases the possibility of arriving at an accurate solution by validating the conditions until termination.

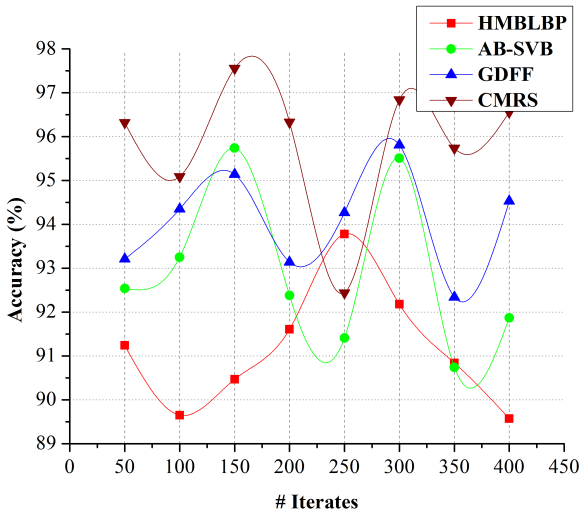

FIGURE 6. Accuracy versus iterates.

1) ANALYSIS OF ACCURACY

In Figure 6, the accuracy observed in the proposed CMRS is compared with the existing methods. In the matrix resolution level, the requirement of s_f for validating the covariance matrix is determined. The dissimilarity factor is estimated based on α_i and α_{si} for both the input and stored images. This pre-estimates the accuracy level by distinguishing N and $|N-1|$ blocks at each covariance estimation. The detained/error block is finely analyzed in the recurrent analysis level. In this level, ∇d_N is estimated until $\varphi \geq \varphi_s$ and $\delta \geq \delta_s$ conditions are satisfied. If at most one condition is satisfied, the block is considered to provide useful information, and it is discarded from the ∇d set. As the number of iterates increases, $(e_{pN} - \nabla d_N^2)$ and $(\nabla d_N.\alpha_s)$ metrics are estimated for both φ and δ to analyze the possibility of a

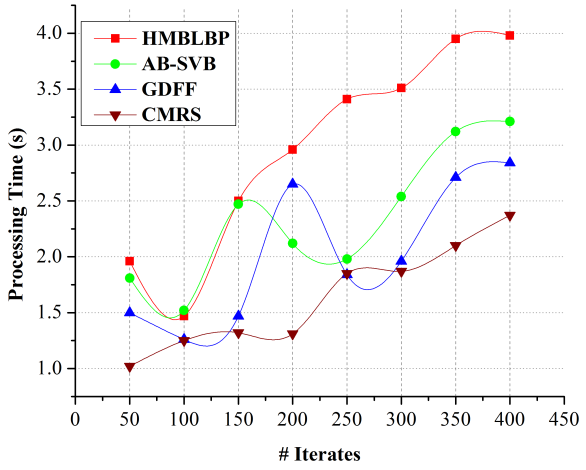

FIGURE 7. Processing time versus iterates.

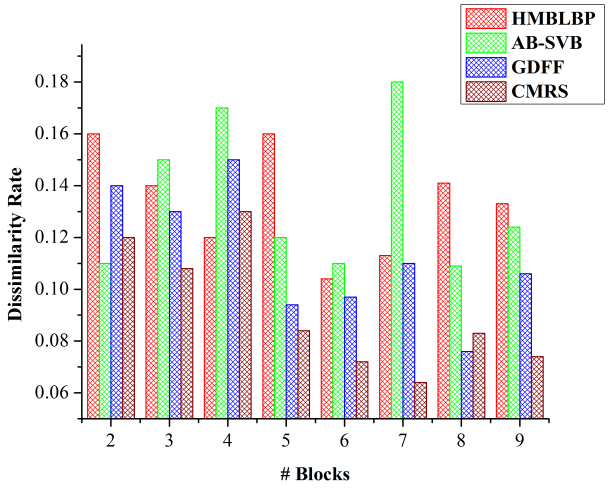

FIGURE 8. Dissimilarity rate versus blocks.

useful information block. Therefore, the varying iterate finds the possible solution for reducing ∇d, thereby improving the accuracy of the image.

2) ANALYSIS OF PROCESSING TIME

Figure 7 presents a comparative analysis of processing time with respect to the varying iterates. As iterates increase, processing time increases. The proposed CMRS employs recurrent analysis for gaining useful information by reducing ∇d over the input block. Unlike the conventional recurrent analysis, this method terminates if either of $\varphi \geq \varphi_s$ and $\delta \geq \delta_s$ conditions are satisfied. The next block is estimated only if both the conditions fail. Therefore, a complete execution of the recurrent analysis is not required for CMRS. Furthermore, the matrix resolution level predetermines the blocks based on covariance matching with the stored image.

Thus, the process of recurrent analysis does not cover the entire blocks; rather, the blocks that contain ∇d (per Condition 1 or Condition 2) are alone induced in this process to refine ∇d_N and other eligible blocks. Similarly, the proposed CMRS pursues block-based analysis rather than operating on an entire image, thereby reducing the processing time.

B. IMPACT OF VARYING BLOCKS

The varying number of blocks decreases the chances of the dissimilarity rate, and hence, the mean accuracy rate is retained. Although it increases the processing time of an image, it reduces the complexity of analysis by generating blocks with appropriate pixel points.

1) DISSIMILARITY ANALYSIS

The dissimilarity rate of the proposed CMRS is comparatively less (see Figure 8) with respect to the varying blocks. Increased blocks reduce the complexity of image analysis and the chances of errors in covariance formation. The covariance matrix is formed for all the segregated N blocks, and ∇d is estimated in both $p = 1$ *to* N and $|N - p|$ cases. The dissimilarity rate identified in the matrix resolution is reduced using a recurrent analysis process on the basis of flexible features. To reduce the complexity in processing, the optimal blocks are directly used for image recognition. On the other hand, ∇d_N is estimated for the $|N - p|$ condition for the φ and δ to reduce the dissimilarity. Therefore, ∇d identified (as in Equations (6) and (9)) is reduced to ∇d_N (as in Equation (11)). This helps reduce dissimilarity by analyzing the blocks through different iterates until the terminating condition is met.

2) MEAN ACCURACY ANALYSIS

The rate of processing multiple blocks of an image recurrently decreases the chances of dissimilarity. The dissimilarity is reduced in the matrix resolution layer that is further refined by estimating ∇d_N using recurrent analysis. Although these features help reduce the dissimilarity rate, the block conversion process is responsible for reducing computation errors in image analysis.

It identifies the exact position of the blocks with respect to the pixels preventing overlapping and tampered bits causing errors. Therefore, CMRS focuses on reducing the dissimilarity in all the first three levels abruptly. Furthermore, the segregated blocks are less complex in determining ∇d and ∇d_N independently.

The segregation of a bigger image into simple blocks of less size reduces the computations on it. The dissimilarity and difference in dissimilarity is assessed in successive blocks on the basis of pixel positions. The consecutive position of the pixels is used for estimating the rate of variance for all the N-blocks. This kind of estimation reduces hefty computation over the entire block, preventing additional time for block processing. As the computation time decreases, the rate of block processing increases, reducing the errors in pixel analysis.

Therefore, the accuracy of the analyzed blocks is high. To handle erroneous blocks, S_f is introduced to consent with

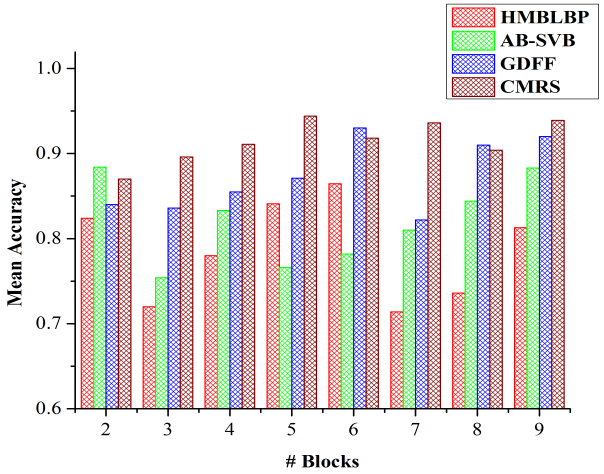

FIGURE 9. Mean accuracy versus blocks.

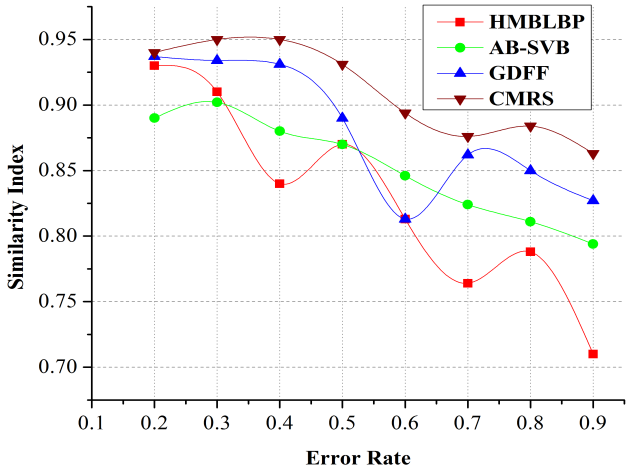

FIGURE 10. Similarity index versus error rate.

$|N - p|$ blocks at the time of covariance estimation. In erroneous block identification, the concentration of 0's and 1's is considered to obtain useful information through φ and δ extraction. However, the ∇d_N evaluation facilitates the final reduction of dissimilarity, thereby improving the accuracy rate (Figure 9).

3) SIMILARITY INDEX ANALYSIS

In Figure 10, the similarity index is compared between the existing methods and the proposed CMRS with respect to the error rate. In general, if the error rate increases, the similarity index decreases. The error rate is estimated as in Equation (12) for the addressed blocks at the time of recognition.

The mismatching blocks are considered as errors provided $r_v(e_i, e_{si}) > \nabla d$ is satisfied. Hence, the error rate is deprived of the estimated similarity index, causing degradation. In CMRS, the possible cases of ∇d and ∇d_N are classified and refined through the level 2 and level 3 processes for the input image to reduce dissimilarity. This positive feature augments the rate of the similarity index at the time of matching (x_p, y_p) with the stored image pixel positions with respect to $M(x_p, y_p)$ to prevent overlapping.

The missing pixel positions are consented by S_f elements, and thereby, at most, either φ or δ is extracted that satisfies the recurrent analysis condition. Thus, the error rate has less impact over the similarity index of the image in CMRS. In Table 4, the comparative analysis values of the existing and proposed methods are presented.

TABLE 4. Comparative analysis values.

Metric	AB-SVM	GDFF	HMBLBP	CMRS
Accuracy (%)	89.57	91.87	94.53	96.56
Processing Time (s)	3.98	3.21	2.84	2.371
Dissimilarity Rate	0.133	0.124	0.106	0.074
Mean Accuracy	0.813	0.883	0.92	0.939
Similarity Index	0.71	0.794	0.827	0.863

VI. CONCLUSION

This manuscript discusses a classifier matrix recognition system for processing sensed traffic images for improving the reliability of road and navigation applications. This recognition system splits the image into blocks and then analyzes its covariance matrix for precise matching with the stored images. The functions of the system are classified into four levels to extract useful information from the processed images. The useful information helps determine the traffic scenario and assists end users through notification/guided navigation. The proposed recognition system is consistent in identifying and matching the similarity between input and stored images by improving accuracy by 4.7% and the similarity index by 8.6% and by reducing processing time by 27.693% and the dissimilarity rate by 4.7%.

REFERENCES

[1] A. H. Alavi, P. Jiao, W. G. Buttlar, and N. Lajnef, "Internet of Things-enabled smart cities: State-of-the-art and future trends," *Measurement*, vol. 129, pp. 589–606, Dec. 2018.

[2] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 696–707, Apr. 2018.

[3] B. Jedari, F. Xia, H. Chen, S. K. Das, A. Tolba, and AL-M. Zafer, "A social-based watchdog system to detect selfish nodes in opportunistic mobile networks," *Future Gener. Comput. Syst.*, vol. 92, no. 3, pp. 777–788, Mar. 2019.

[4] A. Tolba and A. Altameem, "A three-tier architecture for securing IoV communications using vehicular dependencies," *IEEE Access*, vol. 7, pp. 61331–61341, 2019.

[5] D. Li, Z. Cai, L. Deng, and X. Yao, "IoT complex communication architecture for smart cities based on soft computing models," *Soft Comput.*, vol. 23, no. 8, pp. 2799–2812, Apr. 2019.

[6] Z. Al-Makhadmeh and A. Tolba, "An intelligence-based recurrent learning scheme for optimal channel allocation and selection in device-to-device communications," in *Circuits, Systems, and Signal Processing*. New York, NY, USA: Springer, 2019, doi: 10.1007/s00034-019-01056-7.

[7] O. Said and A. Tolba, "Design and performance evaluation of mixed multicast architecture for Internet of Things environment," *J. Supercomput.*, vol. 74, no. 7, pp. 3295–3328, Jul. 2018.

[8] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Jun. 2019.

[9] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "Traj compressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2019.2910591.

[10] S. K. A. Hossain, M. A. Rahman, and M. A. Hossain, "Edge computing framework for enabling situation awareness in IoT based smart city," *J. Parallel Distrib. Comput.*, vol. 122, pp. 226–237, Dec. 2018.

[11] A. Tolba, "Content accessibility preference approach for improving service optimality in Internet of vehicles," *Comput. Netw.*, vol. 152, pp. 78–86, Apr. 2019.

[12] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017.

[13] Z. Ning, P. Dong, and X. Wang, "When deep reinforcement learning meets 5G vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2937079.

[14] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, and Y. Wang, "TripImputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3292–3304, Oct. 2018.

[15] Z. Ning, Y. Feng, M. Collotta, X. Kong, X. Wang, L. Guo, X. Hu, and B. Hu, "Deep learning in edge of vehicles: Exploring trirelationship for data transmission," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5737–5746, Oct. 2019, doi: 10.1109/TII.2019.2929740.

[16] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework," *Future Gener. Comput. Syst.*, vol. 83, pp. 619–628, Jun. 2018.

[17] Z. Ning, P. Dong, X. Wang, J. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, Oct. 2019, Art. no. 60, doi: 10.1145/3317572.

[18] A. Alarifi, A. Tolba, Z. Al-Makhadmeh, and W. Said, "A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks," in *The Journal of Supercomputing*. New York, NY, USA: Springer, 2018, doi: 10.1007/s11227-018-2398-2.

[19] Z. Ning, J. Huang, X. Wang, J. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling," *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep./Oct. 2019, doi: 10.1109/MNET.2019.1800309.

[20] R. Kunst, L. Avila, E. Pignaton, S. Bampi, and J. Rochol, "Improving network resources allocation in smart cities video surveillance," *Comput. Netw.*, vol. 134, pp. 228–244, Apr. 2018.

[21] A. Tolba and E. Elashkar, "Soft computing approaches based bookmark selection and clustering techniques for social tagging systems," *Cluster Comput.*, vol. 22, no. 2, pp. 3183–3189, Mar. 2018, doi: 10.1007/s10586-018-2014-5.

[22] L. Lopez-Fuentes, J. Van de Weijer, M. González-Hidalgo, H. Skinnemoen, and A. D. Bagdanov, "Review on computer vision techniques in emergency situations," *Multimedia Tools Appl.*, vol. 77, no. 13, pp. 17069–17107, Jul. 2017.

[23] K. I. Kiy, "A new method of global image analysis and its application in understanding road scenes," *Pattern Recognit. Image Anal.*, vol. 28, no. 3, pp. 483–495, Jul. 2018.

[24] Z. Zhang, C. Trivedi, and X. Liu, "Automated detection of grade-crossing-trespassing near misses based on computer vision analysis of surveillance video data," *Saf. Sci.*, vol. 110, pp. 276–285, Dec. 2018.

[25] J. Zhu, K. Sun, S. Jia, Q. Li, X. Hou, W. Lin, B. Liu, and G. Qiu, "Urban traffic density estimation based on ultrahigh-resolution uav video and deep neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4968–4981, Dec. 2018.

[26] H. S. Lee and K. Kim, "Simultaneous traffic sign detection and boundary estimation using convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1652–1663, May 2018.

[27] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[28] H. Li, S. Liu, Q. Duan, and W. Li, "Application of multi-sensor image fusion of Internet of Things in image processing," *IEEE Access*, vol. 6, pp. 50776–50787, 2018.

[29] H. Rui, Z. Yunhao, T. Shiming, Y. Yang, and Y. Wenhai, "Fault point detection of IOT using multi-spectral image fusion based on deep learning," *J. Vis. Commun. Image Represent.*, vol. 64, Oct. 2019, Art. no. 102600.

[30] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," *Procedia Comput. Sci.*, vol. 144, pp. 259–268, Jan. 2018.

[31] L.-L. Wang, H. Y. T. Ngan, and N. H. C. Yung, "Automatic incident classification for large-scale traffic data by adaptive boosting SVM," *Inf. Sci.*, vol. 467, pp. 59–73, Oct. 2018.

[32] H. Zhang, "Geometric discriminative deep features for traffic image analysis," *J. Vis. Commun. Image Represent.*, vol. 57, pp. 163–171, Nov. 2018.

[33] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Comput. Vis. Media*, vol. 4, no. 3, pp. 253–266, Sep. 2018.

[34] S. Bai, Z. Liu, and C. Yao, "Classify vehicles in traffic scene images with deformable part-based models," *Mach. Vis. Appl.*, vol. 29, no. 3, pp. 393–403, Apr. 2017.

[35] H. Hu, Z. Gao, Y. Sheng, C. Zhang, and R. Zheng, "Traffic density recognition based on image global texture feature," *Int. J. Intell. Transp. Syst. Res.*, vol. 17, no. 3, pp. 171–180, Sep. 2019.

[36] L. Ruiqi, Z. Xian, Z. Luo, and L. Lin, "Research on the intelligent judgment of traffic congestion in intelligent traffic based on pattern recognition technology," in *Cluster Computing*. New York, NY, USA: Springer, 2018, doi: 10.1007/s10586-017-1684-8.

[37] H. Ahn, J. H. Lee, and H. J. Cho, "Research of panoramic image generation using IoT device with camera for cloud computing environment," *Wireless Pers. Commun.*, vol. 105, no. 2, pp. 619–634, Dec. 2018.

[38] X. Jiang, H. Ding, H. Shi, and C. Li, "Novel QoS optimization paradigm for IoT systems with fuzzy logic and visual information mining integration," in *Neural Computing and Applications Journal*. London, U.K.: Springer, Jul. 2019, doi: 10.1007/s00521-019-04020-3.

[39] R. Laganière, *OpenCV Computer Vision Application Programming Cookbook*, 2nd ed. Birmingham, U.K.: Packt, 2014.

FAYEZ ALQAHTANI is currently an Associate Professor with the King Saud University (KSU). In 2004, he was appointed as the Director of the Computer Division at the Deanship of Student Affairs. He is also the Head of the Computer Science Department and a member of a number of academic and professional associations, such as the Association for Computing Machinery, the Australian Computer Society, and the Association for Information Systems. He has conducted research projects in several areas of information and communication technology, such as Web 2.0, information security, enterprise architecture, software process improvement, the Internet of Things, and fog computing. He has participated in several academic events.

ZAFER AL-MAKHADMEH received the M.Sc. and Ph.D. degrees from the Department of Computer Engineering, Faculty of Information and Computer Engineering, Kharkov National Technical University of Ukraine, in 1998 and 2001, respectively. He is currently an Associate Professor with the Computer Science Department, Community College, King Saud University, Saudi Arabia. His main research interests include cloud computing, image processing, computer vision, and intelligent systems.

OMAR SAID received the Ph.D. degree from Menoufia University, Egypt. He is currently an Associate Professor with the Department of Mathematics and Computer Science, Menoufia University. He has authored many articles at international journals and conferences. His research areas are network management, the Internet of Things, the Internet protocols, routing, multimedia communication, and wireless communication.

AMR TOLBA received the M.Sc. and Ph.D. degrees from the Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently an Associate Professor with the Faculty of Science, Menoufia University, Egypt. He is on leave from Menoufia University to the Computer Science Department, Community College, King Saud University (KSU), Saudi Arabia. He has authored or coauthored over 50 scientific articles in top ranked (ISI) international journals and conference proceedings. His main research interests include socially aware networks, vehicular ad hoc networks, the Internet of Things, intelligent systems, and cloud computing.