

# Automatic Moving Vehicle Detection and Classification Based on Artificial Neural Fuzzy Inference System

V. Murugan<sup>1</sup> · V. R. Vijaykumar<sup>2</sup>

Published online: 31 January 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** The main objective of this work is to automatically detect moving vehicles on the road. Adaptive Neuro-Fuzzy Inference System (ANFIS) classifier is adopted in this paper to classify moving vehicles on road. An input traffic video scenes are taken with vertical and horizontal positioned cameras. The proposed system contains six major steps such as preprocessing, vehicle detection, tracking, structural matching, feature extraction and classification. In this proposed method, preprocessing consists of color conversion and noise removal. Vehicle detection is performed by using background subtraction and Otsu thresholding algorithm. Kalman filter is used in the third step to track moving vehicles in successive frames. In the fourth step, Active Shape Modelling method is used to recover the 3D shape of the vehicle in order to find the boundaries of vehicle. In the fifth step, features of the detected vehicles are extracted by Harrish corner detector, log Gabor filter and these features are taken into account to classify the types of vehicle. Finally, ANFIS is proposed to classify the vehicles which is trained by updating the membership function. Experimentation results provides better accuracy rate and low mean error rate when compared with the state of art methods.

**Keywords** Active shape modeling · Adaptive Neuro-Fuzzy Inference System (ANFIS) · Feature classification · Feature extraction · Fuzzy Neural Network (FNN) · Kalman filter · Vehicle detection

---

✉ V. Murugan  
muruganenggtheni@gmail.com

V. R. Vijaykumar  
vr\_vijay@yahoo.com

<sup>1</sup> Department of Electronics Engineering, Madras Institute of Technology, Anna University Chennai, Tamilnadu, India

<sup>2</sup> Department of Electronics and Communication Engineering, Anna University Regional Office, Coimbatore, Tamilnadu, India

## 1 Introduction

Intelligent Transportation System (ITS) [1, 2] plays an important role in constructing a modern transportation to improve the safety of lives in the busy roads and important junctions in the highways. In ITS [1, 2], Visual traffic surveillance using computer vision techniques makes it more sophisticated and it is powerful when compared to other sensors to perform better traffic control. It has attracted significant interest in computer vision community due to better ability to detect and classify vehicles. Several methods has been developed for detection, tracking and traffic parameter estimation in automatic visual traffic surveillance system [3].

The extraction of critical traffic parameters like vehicle speed, traffic flow rate are enabled by the vehicle detection and tracking [5, 6]. The classification of vehicles plays an important role in traffic control centers to identify the type of vehicles on roadways in order to record the vehicular traffic data [4].

The visual traffic surveillance system consists of step by step successful process which must be adaptable to different weather and illumination condition. The major difficulties for the system are cast shadows, vehicle headlights and bad illumination. In such situations, detecting the moving vehicle will be a challenging work.

For the above mentioned situation, the proposed system for moving vehicle detection has been designed. The system consist of six major steps: Preprocessing is used to convert true color input image (RGB) into a gray scale image, noise removal using average filtering. Gaussian Mixture Model (GMM) [8] which is used for vehicle detection by separating a static background with a moving foreground and extracts the vehicle in the image by the use of differential morphological closing operation. The Kalman filter is used for vehicle tracking, which predict the new position of the moving object. Otsu thresholding [9] is used to gray scale image into binary image. Structural Matching uses the Active Shape Modelling (ASM) [10] for robust detection of shape under image intensity variation. Feature Extraction is used for extracting the features in which the Harris Corner detection [11], Scale-Invariant Feature Transform (SIFT) descriptor and Speeded-Up Robust Features (SURF) descriptor [12] are used to detect the interest point and respective affine regions. The edge detection used to extract the shape features and Log-Gabor filter is used to extract the texture feature. Finally Adaptive Neuro-Fuzzy Inference System (ANFIS) [13] classifier is proposed to classify the vehicles (e.g.: truck, car, bus...).

This paper is organized as follows: Sect. 2 describes recent related works about Vehicle detection and tracking system. The proposed methodology, composed of different stages are described in Sect. 3 with neat diagram and necessary formulas, Sect. 4 describes the Simulation results and analysis of the proposed methodology. Finally, Sect. 5 renders the conclusions.

## 2 Related Work

Sharma [7] differential morphology closing profile is proposed for automatic extraction of vehicle from the traffic image. To obtain high detection and quality rate, some additional operation has applied as a part of the system. The automated detection method is compared with other traditional image processing methods and the experimental results shows that the automated system provides better results than the traditional methods.

Ambardekar [14] proposed a traffic surveillance system that works without prior knowledge to perform surveillance functions in real time and explicit camera calibration. In this, the pose of a vehicle in the 3D world is detected by using the optical flow and knowledge of camera parameters. The gradient based matching and contour based matching techniques are used to classify the vehicles in the Classification stage. This traffic surveillance system is applied to real time dataset samples which produces higher efficiency and but however noises in the image samples are not removed so the quality of the input image samples are not correct.

Salarpour [15] proposed a method for tracking the multiple vehicles. The Kalman filter, color feature and distance of the vehicle from one frame to the next are used for tracking vehicle. The tracking method can distinguish the vehicles and the system can apply to multiple moving vehicles. The proposed algorithm can deal with the tracking problem such as appearance, disappearance and occlusion. It can work in clutter scene and the results are suitable. On the other hand, this Kalman filter has some issues such as visual homogeneity condition and its effectiveness relies on the speeds of moving objects.

Ozkurt and Camci [16] present a Neural Network (NN) method for vehicle classification and traffic density calculation to get the useful information for traffic management system. The real traffic videos from the Istanbul Traffic Management Company are obtained to analyze the system and the experimental results shows a promising performance. They are most robust than the feature based algorithms, but it becomes slower.

Zhou et al. [17] presented a novel approach for adaptive background estimation. Subsequently, the image is splitted into several small nonoverlapped blocks. The candidates of the vehicle division be able to be created from the blocks if there is a number of change in gray level among the present image and the background removed image. A low-dimensional feature is generated by introducing Principal Component Analysis (PCA) in the direction of two histograms of each candidate, and a classifier is performed based on the SVM classifier to classify it as a part of a real vehicle or not. Lastly, each and every one classified results are combined, and a rectangle is construct to characterize the shape of each vehicle. The results demonstrate with the purpose of the proposed SVM classifier provides higher performance under varied conditions, which be able to strongly and successfully remove the influence of casting shadows, headlights, or bad illumination. But the SVM classifier is a two class classification algorithm and so it needs to the modification designed for multi class classification.

Bota et al. [18] proposed a new vehicle detection, vehicle tracking and vehicle classification framework depending on stereovision. Identify the objects by means of either a points grouping method (for huge objects) or a density map grouping method (for little objects). Execute initial a rough classification depending on the objects' dimensions and track objects related to the motion model of each class. Extract motion features and execute a refined classification. Class explicit dichotomizers are consequently used in the direction of filter the categorized objects, eliminating incorrect classification. A large database of manually labeled objects is used for detecting motion models, training the classifiers and evaluating the accuracy of the system.

Lai et al. [19] proposed system mainly consists of three steps: vehicle region extraction, vehicle tracking, and detection. The background subtraction is considered as the first step to extract the foreground regions from the highway scene. A number of geometric properties are functional to remove the false regions and shadow removal algorithm is designed for finding more accurate segmentation results. Following vehicle detection, a graph-based vehicle tracking algorithm is proposed for building the association among vehicles identified at various time instants. Lastly, develops a two measures, such as aspect ratio and

compactness in order to categorize vehicles. From the experimentation results, three videos with varied lighting conditions are used in order to demonstrate the efficiency of the proposed system.

Battiato et al. [20] proposed a new approach for vehicle tracking and detection which has been developed with a data-driven approach implemented on real-world data. The major aim of the system is the tracking of the vehicles in the direction of recognize lane changes, gates transits and other behaviors functional designed for traffic study. The discrimination of the vehicles into two classes (cars vs trucks) is also needed in favor of electronic truck-tolling. Together tracking and classification are executed on online by a system made up of two major parts such as tracker and classifier, in which it is automatically adjust the design of the system in the direction of the experimental conditions. The results demonstrated that the proposed data-driven approach performs better when compared to the state-of-the-art algorithms.

Kachach and Cañas [21] proposed a new algorithm with three major steps: vehicle detection, vehicle tracking, and vehicle classification. Moving vehicles are identified by an enhanced Gaussian Mixture Model (GMM) for background elimination. The design consist of new a technique in the direction of resolve the occlusion problem with a combination of two-dimensional closeness tracking method and the Kanade–Lucas–Tomasi feature tracking procedure. The final step classifies the shapes into five major vehicle types: motorcycle, car, van, bus, and truck with three-dimensional templates and an algorithm depending on the histogram of oriented gradients and the SVM classifier. The classifiers are verified using both real and simulated traffic. The experimentation results demonstrated that the GRAM-RTM dataset and a proper real video dataset, proposed work performs better than other classifiers.

Zhang et al. [28] proposed a vehicle detection method making full use of high level details on aerial images. In the training stage, it choose front windshield samples to train a part detector and whole vehicle samples to train a root detector. In the matching stage, first use the root detector to define an entire vehicle obtaining the root response, then the part detector is scanned in the root bounding box to decide a front windshield and get the part response.

Yuan et al. [29] proposed a novel context-aware multichannel feature pyramid. The main contribution of this paper is proposing two context-aware structural descriptors, termed as a context-aware difference sign transform feature and context-aware difference magnitude transform feature. An image has been tiled with a dense grid of the cells, and each cell is described by both local details and context-aware structural descriptors. The context-aware structural descriptors have the ability to capture the context-aware structural information of cells. The proposed context-aware multichannel feature pyramid is able to provide more effective features for vehicle detection.

### 3 Automated Traffic Surveillance System

The proposed traffic surveillance system is depicted in Fig. 1, the vertical and horizontal positioned camera is used with respect to static background and moving objects. The input of the preprocessing stage will have N number of frames for t minutes.

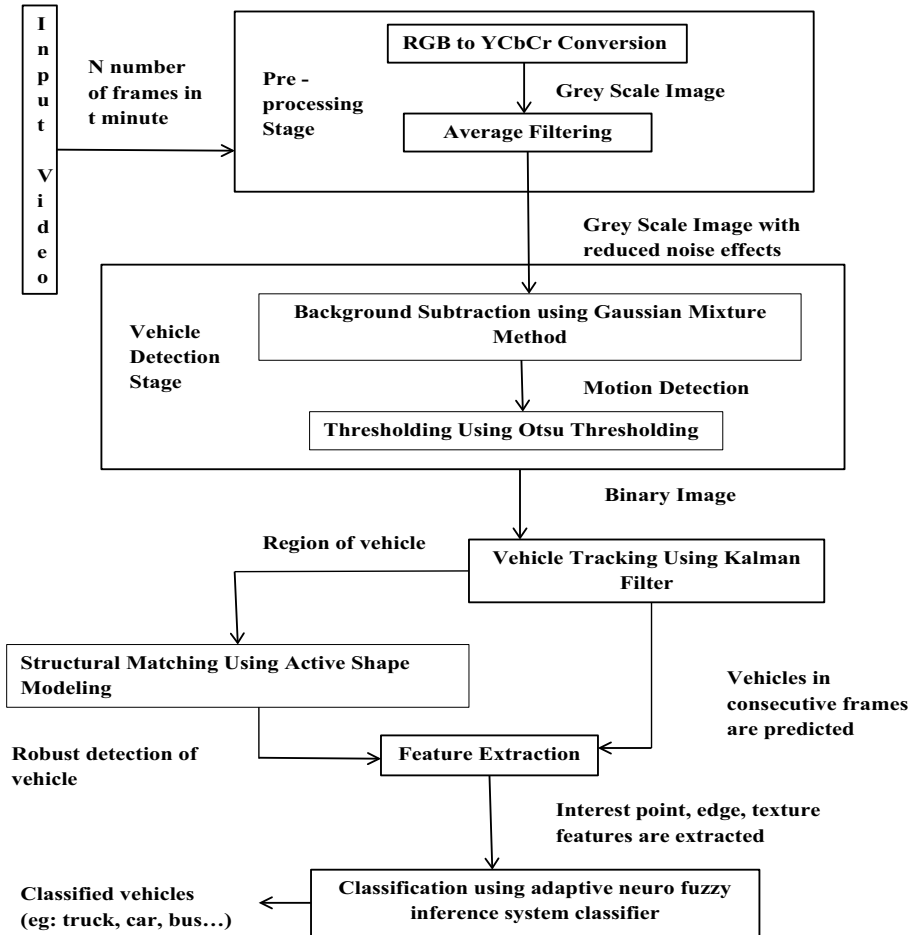


Fig. 1 Proposed traffic surveillance system

### 3.1 Preprocessing

Preprocessing is the initial task which is used to convert RGB true colored video frames to grayscale image by using RGB to YCbCr color space conversion, In this process, the luminance is retained by eliminating the hue and saturation information and it reduces the complexity of the system. The converting of RGB color space to YCbCr color space is shown in the Eq. (1).

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & 0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Averaging filter is applied to the gray scale image  $I_{GS}$  [7] to reduce the false object detection.

Figure 2a shows the real time video for traffic surveillance system, Fig. 2b shows the segmented real time vehicle video.



Fig. 2 a Input video image, b Segmented video image

Figure 3 shows the color conversion of input video from RGB to  $YCbCr$  traffic surveillance system.

Figure 4 shows the noise removed results by using average and Kalman filtering for automatic traffic surveillance system. These filtering methods results are evaluated using the following two metrics. The results are computed and the performance is evaluated based on the parameters like Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE).

### 3.1.1 Peak Signal to Noise Ratio (PSNR)

The PSNR ( $\tau_x$ ) in dB is given as,

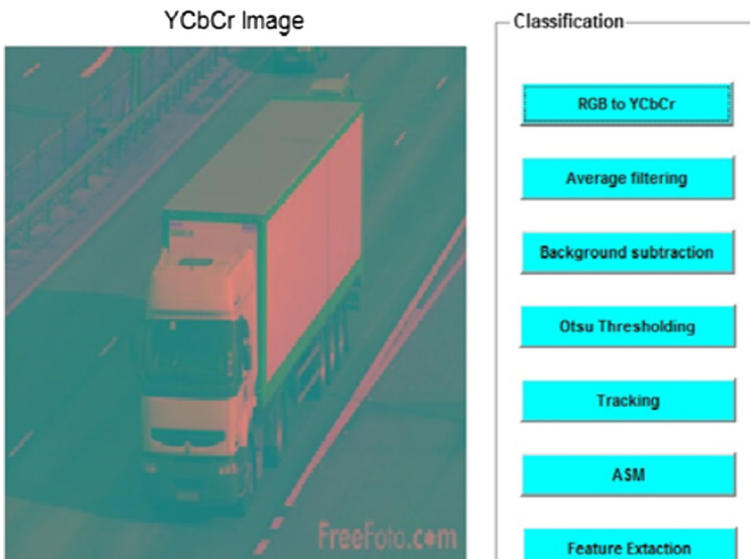


Fig. 3 Color conversion from RGB to  $YCbCr$



**Fig. 4** Noise removal using **a** average filtering, **b** Kalman filtering

$$\tau_x = 10 \log_{10} \frac{R^2}{\mu_x} \quad (2)$$

where  $R$  is the maximum possible value in the corresponding data and  $\mu_x$  is Mean Square Error (MSE).

### 3.1.2 Mean Square Error (MSE)

Mean Square Error (MSE) is defined as

$$\mu_x = \frac{1}{T} \sum_{i=1}^M \sum_{j=1}^N \left( I_x(i,j) - I'_x(i,j) \right)^2 \quad (3)$$

where  $I_x(i,j)$  is the original data,  $I'_x(i,j)$  is the watermarked data, and  $M$  and  $N$  are data height and width such that  $T = M \times N$ . In this work, for video, PSNR is calculated by taking average of PSNR values of all the tracking of corresponding frames of the video. The average PSNR is computed as,

$$\bar{\tau} = \frac{1}{F} \sum_{x=1}^F \tau_x \quad (4)$$

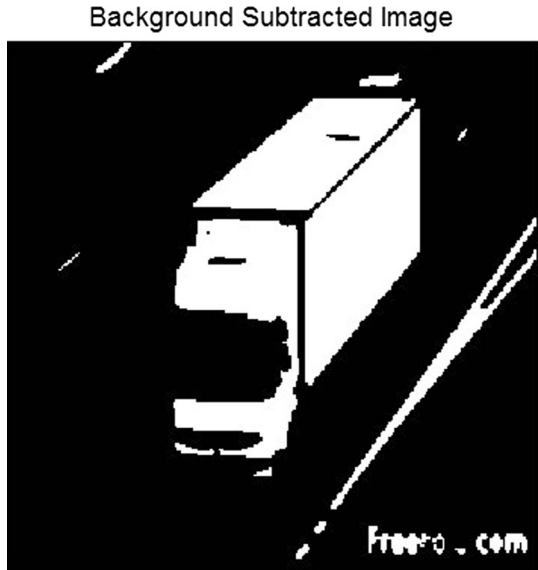
In the Table 1 shows the comparison results of three different filtering methods with two metrics.

## 3.2 Vehicle Detection

The vehicle detection is based on motion detection in the video sequence. The Mixture of Gaussians background subtraction is used for the motion detection and it separates a static

**Table 1** PSNR and MSE results

Metrics	Average filtering	Kalman filtering
PSNR (dB)	38.17	40.23
MSE	0.263	0.2153

**Fig. 5** Background subtraction

background from a static foreground. For each background pixel, the method uses mixture of three Gaussians with respect to road, shadow and vehicle [22, 23]. The GMM estimates the state changes of the corresponding pixel from one frame to the next frame. The current pixel ( $X$ ) probability value is considered and it is given in the following equation,

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \eta(X_t; \mu_{i,t}, SD) \quad (5)$$

where  $K$  is the number of distributions, the  $\eta$  is the Gaussian probability density function and it is given by

$$\eta(X_t, \mu, SD) = \frac{1}{2\pi^{\frac{d}{2}} SD^{1/2}} e^{-\frac{1}{2} \frac{(X_t - \mu)^2}{SD}} \quad (6)$$

whereas, the weight of the  $i$ th Gaussian in the mixture at time  $t$  is

$$\sum_{i=1}^K \omega_{i,t} = 1 \quad (7)$$

The mean of associated component is given by

$$\mu_t = \sum_{i=1}^k \omega_{i,t} \mu_i = 1 \quad (8)$$

The component is treated as matched component, when the value of the pixel is close to a chosen component mean. In order for the matched component, the Standard Deviation (SD) will be greater than the difference between the pixel and mean. The Gaussian weight, Standard deviation and mean are updated for the newly obtained pixel value. The weight is decreased for the non matched component, but the mean and standard deviation will be remains same. The background components are identified by applying the threshold on the



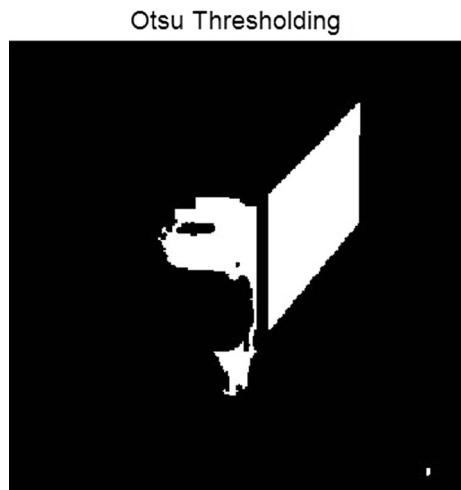
**Fig. 6** GMM results

component weight. Except the background components, the remaining components are determined as foreground. The Morphological operation is done on the background eliminated image for automatic object detection.

Figure 5 shows the background subtracted results and Fig. 6 shows the result of GMM model.

The Differential Morphological Profile (DMP) is generated by using a set of discrete Structuring Elements (SE) and morphological reconstruction operators for the grayscale image [7].

The closing operation is done by reconstruction of the background eliminated image with a structuring element of size  $i$ . The closing by reconstruction is applied to the background eliminated image with a structuring element of size  $i$ . The morphological closing by reconstruction is computed as dilation of the Image with SE, followed by a geodesic (a set of operators) reconstruction by erosion. An iterative procedure is followed in geodesic reconstruction and it is performed until the set of operators are unchanged. The

**Fig. 7** Otsu thresholding results

maximum response point (darker components) is known as multiscale closing characteristics. The vehicles are extracted automatically by using multiscale closing characteristics. To convert the gray scale image into a binary image, Otsu thresholding is applied to the gray scale image and it is illustrated in Fig. 7.

Objects are to be matched in the consecutive frame, to get the meaningful information from the sequence of images in tracking. The Kalman filter is used in this method for tracking the moving vehicle. In this, region of the vehicle is erected first and the center of the vehicle is found based on the vehicle region and the position of the vehicle in the next frame.

---

#### Algorithm 1: Vehicle Segmentation

---

**Input :** Original Gray scale image

**Output:** Threshold ( $t$ )

1. Compute histogram and probabilities of each intensity level
  2. Set up initial  $\omega_i(0)$  and  $\mu_i(0)$
  3. Step through all possible thresholds maximum intensity
    1. Update  $\omega_i$  and  $\mu_i$
    2. Compute  $\sigma_b^2(t)$
  4. Desired threshold corresponds to the maximum  $\sigma_b^2(t)$ .
- 

### 3.3 Vehicle Tracking

The Kalman filter estimates the next time step and updates the position of vehicle from the predicted in the update phase, where it projects forward in time with the current state and error covariance. The posterior estimation of the state and error covariance is predicted in the measurement update phase by incorporating the latest measurements into the system. The center  $C$  of the points  $(x, y)$  composed of two steps, prediction (time update) and correction (measurement update). The priori and area  $A$  variables of the detected vehicle in the image plane are integrated in the Kalman filter. Based on the projection maps from the vehicle vertical and horizontal edges, the variable  $A$  and  $C(x, y)$  are determined and it exposed as vectors.

After the integration of variables  $A$  and  $C(x, y)$ , it will result in the following state ( $m_v$ ) and measurement ( $n_v$ ) vectors.

$$m_v = [x, y, A, v_x, v_y, v_A]^T \quad (9)$$

$$n_v = [x, y, A]^T \quad (10)$$

where  $v_x$  is the velocity in the movement of objects (vehicle) center point in the  $x$  direction, whereas  $v_y$  in  $y$  direction. The rate of change in the object image size is  $v_A$ , where the moving object size will be vary for every frame. In consecutive video frames, the possible appearance of vehicle location and size are predicted in the priori estimate. The Tracking function for consecutive frames utilized the priori information to narrow down the search space for re-detecting a vehicle. Whereas the posteriori estimate is calculated and it is used

as the best estimate for the locating the vehicle and to calculate the size. For multiple vehicles tracking, each and every vehicle in the tracking list is instantiated by Kalman filter. Figure 8 shows that the tracking of a vehicle in the consecutive frames of a video.

### 3.4 Structural Matching Using Active Shape Modeling (ASM)

The region of vehicle is to be tracked in the first frame which will have difficult subjects for 3-D shape recovery, which are highly specular, contain semi transparent areas having large textureless regions. Active Shape Modeling (ASM) is used for the structural matching and it provides a robust shape for recognizing the vehicle class based on recovered 3D shape. Principal Component Analysis (PCA) is used to obtain the statistical properties of class of vehicles and it is used to set 3D poses in Active Shape Model training. The image of 3D poses of vehicle are taken from different poses of body and four wheels of vehicles. Three images of the 3D poses were taken for each vehicle with different position of the body and the positions of the four wheels of vehicles. The training set includes the mean shape, eigen values and eigen vectors [24].

The set of features is pointed automatically and manually along the region of the vehicle. A similar transformation is applied to find the best pose, and some new shape parameter is computed to find best deformation. The estimated shape is the best fit to real images of vehicle and it is achieved by hypothesizing and matching image intensity edges



**Fig. 8** Tracking of real time vehicle in Frame 19, 20 and 21 of Input video

in ASM. The estimated shape for tracking vehicles will be used to classify the types of vehicles [25] in the classification stage and it recovers the pose of a 3D rigid object which is illustrated in Fig. 9.

### 3.5 Feature Extraction

Feature extraction enables to extract a set of feature vectors from a set of detected features. The combination of Harris corner detection, SIFT descriptor and SURF descriptor effectively extract the affine region of the estimated shape. Harris corner detection is used to estimate the Harris motion and it is used to extract the interest corner points on the vehicle. The corner point location is scaled by the SIFT descriptor, where the image was resized with bilinear interpolation to different scales and the SURF feature descriptor is used to recognize the vehicle from the first video frame and match them. Harris corner algorithm is given below as:

1. For each pixel  $(x, y)$  in the image calculate the autocorrelation matrix  $M$  as:

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (11)$$

2. For each pixel of image has Gaussian filtering, get new matrix  $M$ , and discrete two dimensional zero-mean Gaussian function as

$$\text{Gauss} = \exp - (u^2 + v^2)/2\delta^2 \quad (12)$$

3. Calculating the corners measure for each pixel  $(x, y)$  get

$$R = \left\{ I_x^2 I_y^2 - \left( I_x I_y \right)^2 \right\} - k \left\{ I_x^2 + I_y^2 \right\}^2 \quad (13)$$

4. Choose the local maximum point. Harris method considers that the feature points are the pixel value which corresponding with the local maximum interest point.
5. Set the threshold  $T$ , detect the corners points.

The created interest points of the vehicle from a video sequence of frames are extracted from vehicles using SURF descriptor [26]. The SIFT algorithm has four main steps: (1) Scale Space Extrema Detection, (2) Key point Localization, (3) Orientation Assignment and (4) Description Generation. The first stage is to identify location and scales of key points using scale space extrema in the DoG (Difference-of Gaussian) functions with different values of  $\sigma$ . The DoG function is convolved with the image in scale space



**Fig. 9** ASM modelling results

separated by a constant factor  $k$ . In the key point localization step, key point candidates are localized and refined by eliminating the key points where they reject the low contrast points. In the orientation assignment step, the orientation of key point is obtained based on local image gradient. In description generation stage, the local image descriptor is computed for each key point based on image gradient magnitude and orientation at each image sample point in a region centered at key point. The algorithm 2 explains the steps of the SIFT descriptor.

---

**Algorithm 2: SIFT Feature Extraction**


---

**Input:** Original Image ( $L$ )

**Output:** SIFT descriptor values

- Use the keypoints scale to select the Gaussian smoothed image  $L$ , from above
  - Compute gradient magnitude,  $m$   

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$
  - Compute orientation,  $\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$
  - Form an orientation histogram from gradient orientations of sample points
  - Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
  - Some points will be assigned multiple orientations
  - Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position.
- 

SURF (Speed Up Robust Features) algorithm, is base on multi-scale space theory and the feature detector is base on Hessian matrix. Since Hessian matrix has good performance and accuracy. In image  $I$ ,  $x = (x, y)$  is the given point, the Hessian matrix  $H(x, \sigma)$  in  $x$  at scale  $\sigma$ , it can be define as

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{yx}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (14)$$

where  $L_{xx}(\mathbf{x}, \sigma)$  is the convolution result of the second order derivative of Gaussian filter  $\partial^2/\partial x^2(\sigma)$  with the image  $I$  in point  $x$ , and similarly for  $L_{xy}(\mathbf{x}, \sigma)$  and  $L_{yy}(\mathbf{x}, \sigma)$ . SURF creates a “stack” without 2:1 down sampling for higher levels in the pyramid resulting in images of the same resolution. Due to the use of integral images, SURF filters the stack using a box filter approximation of second-order Gaussian partial derivatives. Since integral images allow the computation of rectangular box filters in near constant Time.

To detect the meaningful discontinuities at the edges, which is a set of connected points that lie on the boundary between two regions. The canny edge detection method is used to extract edges of the vehicles. The canny edge detection smoothens the image by blurring the image to remove the noise. The gradients of detection in image are having large magnitudes and they are marked as edges. Two thresholds are applied to determine the potential edges. The edges which are not connected to a certain edge is to be suppressed to determine the final edges. The edges and the occlusion boundary of an object are extracted. To retrieve the image data (mesh of the vehicle) the texture information should be extracted. The Log-Gabor filter method [27] is used to extract the texture feature, which is constructed with arbitrary bandwidth. The image data shape becomes much sharper whereas bandwidth increases. The edge detection results of the image is shown in Fig. 10.

### 3.6 Classification

Adaptive Neuro Fuzzy Inference System (ANFIS) composes of two intelligent techniques, namely Neural Network adaptive capabilities and fuzzy Inference System, which adapts parameters of the Fuzzy inference system using neural networks. The basic structure of the ANFIS is shown in the Fig. 11 and it consists of five layers and to illustrate structure consider two fuzzy if-then rules

Rule 1: *if* ( $x$  is  $A_1$ ) *and* ( $y$  is  $B_1$ ) *then* ( $f_1 = a_1x + b_1y + c_1$ )

Rule 2: *if* ( $x$  is  $A_2$ ) *and* ( $x$  is  $B_2$ ) *then* ( $f_1 = a_2x + b_2y + c_2$ )

where  $x$ ,  $y$  are the inputs,  $A_1$  and  $B_1$  are the fuzzy sets,  $a_i$ ,  $b_i$  and  $c_i$  are the design parameters

In the ANFIS structure, the square represents the adaptive node while the circle represents the fixed nodes. The output of the first layer are the fuzzy membership grade of the inputs are  $x$  and  $y$  and it is given in the following equation

$$O_i^1 = \mu_{A_i}(x), \quad i = 1, 2, \quad O_i^1 = \mu_{B_{i-2}}(x), \quad i = 3, 4, \quad (15)$$

where  $\mu_{A_i}$  and  $\mu_{B_{i-2}}$  can accept any fuzzy membership function.

Typical Membership function is represented as following equation

$$\mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x-r_i}{p_i}\right)^{q_i}} \quad (16)$$

where  $p_i$ ,  $q_i$  and  $r_i$  are the premise parameters of the membership function to control the bell shaped function consequently.

The nodes in the second layer is fixed nodes and they are indicated by M, which is a simple multiples. The output of the layer two is called as the firing strength of the rules and it is represented as in Eq. (17).

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_{i-2}}(x), \quad i = 1, 2 \quad (17)$$

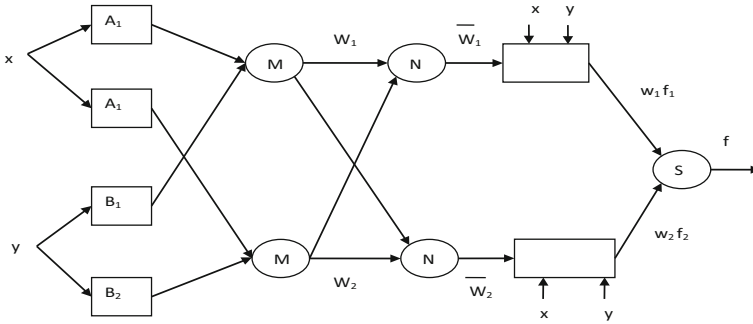
The nodes in the third layer is fixed node and they are indicated by N, which is engaged in normalization function to the early layer outputs and the output of the third layer is called as normalized firing strength and it is given by

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (18)$$

#### Edge result



**Fig. 10** Edge detection results



**Fig. 11** ANFIS structure

The output of the fourth layer is the product of the normalized firing strength and a first order polynomial. The output of the fourth layer is given as,

$$O_i^4 = \bar{w}_i f = \bar{w}_i (a_i x + b_i y + c_i) \tag{19}$$

where  $a_i$ ,  $b_i$  and  $c_i$  are called consequent parameters.

The layer 5 has one fixed node and it is labeled as S, where the summation operation is done for every incoming signal. The overall output of the model is given as follows

$$O_i^5 = \sum_{i=1}^2 \bar{w}_i f_i = \frac{\sum_{i=1}^2 w_i f_i}{w_1 + w_2} \tag{20}$$

### 3.7 ANFIS Training and Testing

A data set (input–output data pairs) is obtained in the ANFIS modelling process and the data-pairs are divided into training and checking data sets. The training data is normalized by using the max–min method and it make each term lies between 0 and 1 in order to make it suitable for the training process. The input vector (set of feature extracted) and the output vector (classes of vehicles) are formed in order to train the data. The information about the feature extracted must be transformed into numerical code and registered on the ANFIS. The initial premise parameters are determined using the training data set for the membership functions by leaving equal space for each of the membership functions. The least square method is used to find the consequent parameters and these parameters are fixed. Then the premise parameters are updated using the gradient decent method at the condition of error tolerance goal is not reached. The process is terminated when the error tolerance is reached. The ANFIS system is now ready to classify the vehicles for given feature vectors. This work uses the ANFIS Editor GUI menu bar to load a FIS training initialization, and then save the trained FIS from this GUI:

1. Load data (training) by selecting the appropriate radio buttons in the Load data portion of the GUI and then clicking Load Data. The loaded data is plotted on the plot region.
2. Generate an initial FIS model or load an initial FIS model using the options in the generate FIS portion of the GUI.
3. View the FIS model structure once an initial FIS has been generated or loaded by clicking the Structure button.

4. Choose the FIS model parameter optimization method back propagation or a mixture of back propagation and least squares (hybrid method). This research uses the default hybrid method as well as back propagation parameter optimization method.
5. Choose the number of training epochs and the training error tolerance. This research uses 10, 30, 50 and 100 as training epochs and 0 as training error tolerance.
6. Train the FIS model by clicking the Train Now button. This training adjusts the membership function parameters and plots the training error plot(s) in the plot region.
7. View the FIS model output versus the training, checking, and testing data output by clicking the Test Now button. This function plots the rest data against the FIS output in the plot region.

## 4 Experimental Results

The performance on the proposed traffic surveillance system has been tested on a video sequence of 3000 frames containing 54 vehicles. The system is able to detect, track and classify the vehicles in an efficient manner. The average velocity of traffic surveillance system calculated by posted speed limits. Table 2 shows the quantitative results for the patch of street under surveillance in the video sequence S1 and the posted limit was 25 mph. The vehicle classes are marked as follows: Car (1), Van (2), Bus (3), and Truck (4). According to the posted speed limits, the average velocity found by the traffic surveillance system for the classes of vehicle.

The training process of ANFIS system with the training data set is shown in Fig. 12. The training error is the difference between the training data output value, and the output of the FIS corresponding to the same training data input value, (the one allied with that training data output value). The MSE of the training data is calculated for each epoch. The proposed system achieves MSE of  $4.8728e-06$  at 39th epoch. The performance of the proposed system against the different classes of vehicles has been calculated by the following measures.

Sensitivity also called the *True Positive Rate (TPR)*, the recall which measures the proportion of positives that are correctly identified as having the condition.

Specificity also called the *True Negative Rate (TNR)* measures the proportion of negatives that are correctly identified as negative condition.

$$\text{Sensitivity} = \text{TPR} = \frac{\text{No. of vehicles detected}}{\text{No. of vehicles appearing in the video frames}} \times 100 \quad (21)$$

**Table 2** Quantitative results for video sequence S1

Vehicle number	Actual class of vehicle	Classified class of vehicle	Average velocity (mph)
1	2	2	29
2	3	3	25.34
3	1	1	29.56
4	1	1	32.07
5	2	4	30.67



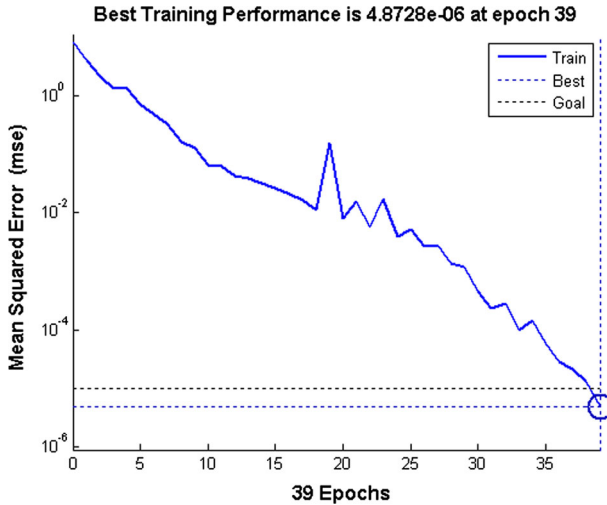


Fig. 12 Training error

$$Specificity = TNR = \frac{No. \ of \ vehicles \ not \ detected}{No. \ of \ vehicles \ appearing \ in \ the \ video \ frames} \times 100 \quad (22)$$

The False Positive Rate (FPR) is calculated as the ratio between the number of negative events wrongly categorized as false positives and the total number of actual negative events.

The False Negative Rate (FNR) is the proportion of positives which yield negative test outcomes with the test, i.e., the conditional probability of a negative test result given that the condition being looked for is present.

$$False \ Positive \ rate \ (FPR) = \frac{Number \ of \ false \ detections}{Number \ of \ vehicles \ detected + Number \ of \ false \ detections} \times 100 \quad (23)$$

$$False \ Negative \ Rate \ (FNR) = \frac{Number \ of \ Vehicles \ missed}{Number \ of \ vehicles \ appearing \ in \ the \ video \ frames} \times 100 \quad (24)$$

The ROC curve is a fundamental tool for diagnostic test evaluation. In a ROC curve the True Positive Rate (Sensitivity) is plotted in function of the False Positive Rate (100-Specificity) for different cut-off points of a parameter. The results are shown in Fig. 13.

The Fig. 13 shows the comparative view of the proposed ANFIS system with gradient and contour based matching techniques [14], ANN [16], aspect ratio and compactness [19], data-driven approach [20], SVM [21] for the ROC analysis and the proposed system achieve better TPR than the other methods. The classification accuracy of the system depends on the number of vehicles correctly classified (true positives + true negatives) and is calculated by the following formula

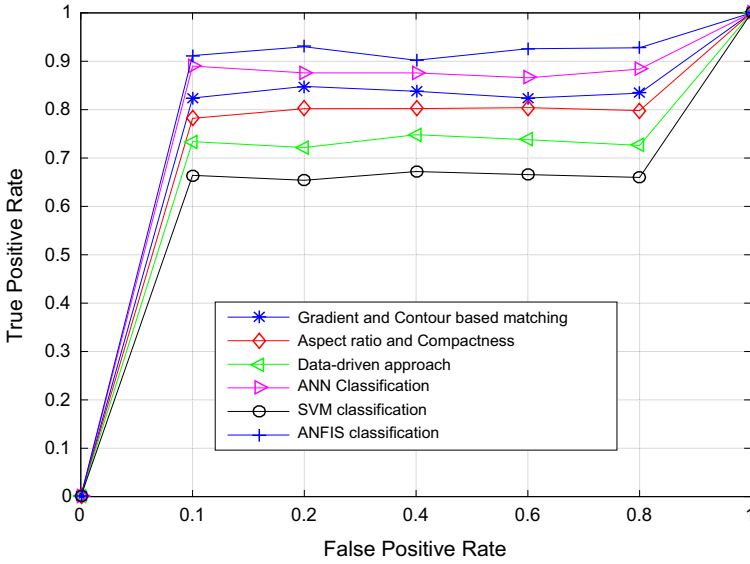


Fig. 13 TPR and FPR over proposed and other classification methods

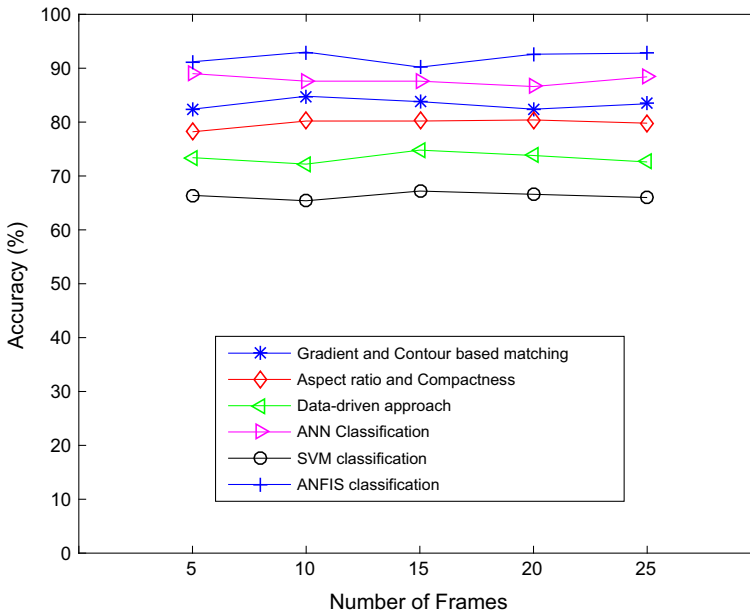


Fig. 14 Comparison of accuracy over proposed and other classification methods

$$Classification\ Accuracy = \frac{Number\ of\ vehicles\ classified\ correctly}{Total\ number\ of\ vehicles} \quad (25)$$

**Table 3** Classification accuracy vs detection methods

No. of frames	Accuracy (%)					
	Gradient and contour based matching techniques [14]	Aspect ratio and compactness [19]	Data-driven approach [20]	ANN [16]	SVM [21]	ANFIS
5	83.8	80.4	74.6	86.6	67.2	91.4
10	85	81	74	88.6	67.6	91
15	82.8	79.4	73.2	86.6	65.6	93
20	82.2	78.8	72.2	88	67.4	91.2
25	83.2	79.2	74.6	88	65.4	92.6

**Table 4** Error rate vs detection methods

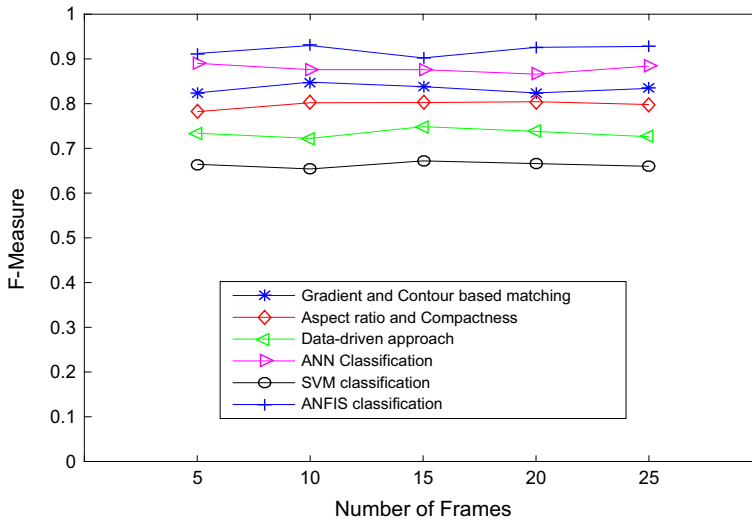
No. of frames	Error rate (%)					
	Gradient and contour based matching techniques [14]	Aspect ratio and compactness [19]	Data-driven approach [20]	ANN [16]	SVM [21]	ANFIS
5	16.2	19.6	25.4	13.4	32.8	8.6
10	15	19	26	11.4	32.4	9
15	17.2	20.6	26.8	13.4	34.4	7
20	17.8	21.2	27.8	12	32.6	8.8
25	16.8	20.8	25.4	12	34.6	7.4

The classification accuracy of the proposed ANFIS system is compared with the with gradient and contour based matching techniques [14], ANN [16], aspect ratio and compactness [19], data-driven approach [20], SVM [21] and it has achieved 92.56% accuracy of crossing vehicle in the video sequence. The accuracy results shown in the Fig. 14 and the values of accuracy and error rate are tabulated in Tables 3 and 4 respectively.

The f-measure of the proposed ANFIS system is compared with the with gradient and contour based matching techniques [14], ANN [16], aspect ratio and compactness [19], data-driven approach [20], SVM [21] and it has achieved 92.56% accuracy of crossing vehicle in the video sequence. The f-measure results shown in the Fig. 15 and the values are tabulated in Table 5.

## 5 Conclusion

This paper deals with the traffic surveillance system. The proposed system consists of four major steps such as background subtraction, vehicle detecting and tracking using structural matching, feature extraction and classification. The Preprocessing technique is used to convert true colour to grey scale image. Later preprocessing technique, the foreground object is detected by use of Gaussian Mixture Model and morphological operation is done to extract the detected moving vehicle. After extracting the vehicle in current frame, Kalman filter is used in this method to predict the position of moving vehicles in next frame and it is used to track the multiple vehicles in single frame. Kalman filter predicts the moving vehicles in successive frames. ASM is used to recover the 3D shape of the vehicles



**Fig. 15** Comparison of F-measure over proposed and other classification methods

**Table 5** F-measure versus detection methods

No. of frames	Gradient and contour based matching techniques [14]	Aspect ratio and compactness [19]	Data-driven approach [20]	ANN [16]	SVM [21]	ANFIS
5	0.83737	0.80385	0.74664	0.86573	0.67129	0.91444
10	0.84927	0.81002	0.74069	0.88619	0.67474	0.91021
15	0.82742	0.79423	0.73145	0.8664	0.65584	0.93
20	0.82136	0.78828	0.7224	0.88013	0.67358	0.91137
25	0.83086	0.79184	0.74507	0.87935	0.65251	0.92589

and feature vectors of vehicles are extracted. ANFIS classifies the detected vehicles based on the feature vectors of detected vehicles. The experimental results shown that the proposed system provides better accuracy when compared to conventional neural networks. The proposed system is used to classify the multiple classes of vehicles and it can be used for traffic surveillance system.

## References

1. Dimitrakopoulos, G., & Demestichas, P. (2010). Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1), 77–84.
2. Joseph, A. D., Beresford, A. R., Bacon, J., Cottingham, D. N., Davies, J. J., Jones, B. D., et al. (2006). Intelligent transportation systems. *IEEE Pervasive Computing*, 5(4), 63–67.

3. Yu, S. H., Hsieh, J. W., Chen, Y. S., & Hu, W. F. (2003, June). An automatic traffic surveillance system for vehicle tracking and classification. In *Scandinavian conference on image analysis* (pp. 379–386). Berlin: Springer.
4. Hsieh, J. W., Yu, S. H., Chen, Y. S., & Hu, W. F. (2006). Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Transactions on Intelligent Transportation Systems*, 7(2), 175–187.
5. Ozkurt, C., & Camci, F. (2009). Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Mathematical and Computational Applications*, 14(3), 187–196.
6. Motamed, C. (2006). Motion detection and tracking using belief indicators for an automatic visual-surveillance system. *Image and Vision Computing*, 24(11), 1192–1201.
7. Sharma, B., Katiyar, V. K., Gupta, A. K., & Singh, A. (2014). The automated vehicle detection of highway traffic images by differential morphological profile. *Journal of Transportation Technologies*, 4(02), 150–156.
8. Huang, Z. K., & Chau, K. W. (2008). A new image thresholding method based on Gaussian mixture model. *Applied Mathematics and Computation*, 205(2), 899–907.
9. Lang, X., Zhu, F., Hao, Y., & Ou, J. (2008). Integral image based fast algorithm for two-dimensional Otsu thresholding. In *Congress on image and signal processing, 2008 (CISP'08)* (Vol. 3, pp. 677–681).
10. Kroon, D. J. *Active shape model and active appearance model*. <http://www.mathworks.com/matlabcentral/fileexchange/26706>.
11. Chen, B. F., & Cai, Z. X. (2005). Harris corner detection based on theory of scale-space. *Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology)*, 36(5), 751–754.
12. Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615–1630.
13. Mohandes, M., Rehman, S., & Rahman, S. M. (2011). Estimation of wind speed profile using adaptive neuro-fuzzy inference system (ANFIS). *Applied Energy*, 88(11), 4024–4032.
14. Ambardekar, A. A. (2007). *Efficient vehicle tracking and classification for an automated traffic surveillance system*. University of Nevada, Reno. Master of Science, pp. 1–76.
15. Salarpour, A., Salarpour, A., Fathi, M., & Dezfoulian, M. (2011). Vehicle tracking using Kalman filter and features. *Signal & Image Processing*, 2(2), 1–8.
16. Ozkurt, C., & Camci, F. (2009). Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Mathematical and Computational Applications*, 14(3), 187–196.
17. Zhou, J., Gao, D., & Zhang, D. (2007). Moving vehicle detection for automatic traffic monitoring. *IEEE Transactions on Vehicular Technology*, 56(1), 51–59.
18. Bota, S., Nedevschi, S., & Konig, M. (2009). A framework for object detection, tracking and classification in urban traffic scenarios using stereovision. In *IEEE 5th international conference on intelligent computer communication and processing, 2009 (ICCP 2009)* (pp. 153–156).
19. Lai, J. C., Huang, S. S., & Tseng, C. C. (2010). Image-based vehicle tracking and classification on the highway. *International Conference on Green Circuits and Systems (ICGCS), 2010*, 666–670.
20. Battiato, S., Farinella, G. M., Furnari, A., Puglisi, G., Snijders, A., & Spiekstra, J. (2015). An integrated system for vehicle tracking and classification. *Expert Systems with Applications*, 42(21), 7263–7275.
21. Kachach, R., & Cañas, J. M. (2016). Hybrid three-dimensional and support vector machine approach for automatic vehicle tracking and classification using a single camera. *Journal of Electronic Imaging*, 25(3). <https://doi.org/10.1117/1.JEI.25.3.033021>.
22. Bouwmans, T., El Baf, F., & Vachon, B. (2008). Background modeling using mixture of gaussians for foreground detection—A survey. *Recent Patents on Computer Science*, 1(3), 219–237.
23. Santosh, D. H. H., Venkatesh, P., Poornesh, P., Rao, L. N., & Kumar, N. A. (2013). Tracking multiple moving objects using gaussian mixture model. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(2), 114–119.
24. Liu, N., & Lovell, B. C. (2005, December). Hand gesture extraction by active shape models. In *Digital image computing: Techniques and applications, 2005 (DICTA'05)* (pp. 1–10).
25. Leotta, M. J., & Mundy, J. L. (2009). Predicting high resolution image edges with a generic, adaptive, 3-D vehicle model. In *IEEE conference on computer vision and pattern recognition, 2009 (CVPR 2009)* (pp. 1311–1318).
26. Anitha, J. J., & Deepa, S. M. (2014). Tracking and recognition of objects using SURF descriptor and Harris corner detection. *International Journal of Current Engineering and Technology*, 4(2), 775–778.
27. Garg, V., & Raheja, N. (2002). Image denoising using curvelet transformation using log gabour filter. *International journal of Recent Technology and Engineering (IJRTE)*, 2(1), 137–142.

28. Yuan, X., Cao, X., Hao, X., Chen, H., & Wei, X. (2017). Vehicle detection by a context-aware multichannel feature pyramid. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7), 1348–1357.
29. Zhang, J., Tao, C., & Zou, Z. (2017). An on-road vehicle detection method for high-resolution aerial images based on local and global structure learning. *IEEE Geoscience and Remote Sensing Letters*, 14(8), 1198–1202.



**V. Murugan** currently working as a Faculty Member in the Department of Electronics Engineering, Madras Institute of Technology (MIT) Campus, Anna University, Chennai, India. He is having seven years of teaching experience in the field of signal and image processing for communications and guided four Postgraduate and nine Undergraduate students. He has published one international journal and presented papers in four international and two national conferences.



**Dr. V. R. Vijaykumar** currently working as Associate Professor in Department of ECE, Anna University Regional Office, Coimbatore, India. He is having more than 17 years of teaching experience and guided 2 Ph.D. and 55 PG students. He is doing research in the area of signal and image processing, VLSI architecture and Wireless Communication. He has published 13 international journals and 3 national journals.