Flow Analysis of Vehicles on a Lane Using Deep Learning Techniques

Aruna Kumar Joshi [1,]* and Shrinivasrao B. Kulkarni [2]

[1] Department of Computer Science and Engineering, SKSVMA College of Engineering and Technology, Laxmeshwar, and Visvesvaraya Technological University, Belagavi, Karnataka, India

[2] Department of Computer Science and Engineering, SDM College of Engineering and Technology, Dharwad, and Visvesvaraya Technological University, Belagavi, Karnataka, India; Email: sbkulkarni_in@yahoo.com (S.B.K.)

*Correspondence: arunkumarjoshi.sudi@gmail.com (A.K.J.)

***Abstract*—Increased volume of vehicles on the road everywhere in the world. It becomes difficult to regulate the traffic manually across all the traffic signals throughout the cities for the entire day. It is costly and tedious for manual traffic control at traffic signals. An Intelligent Transportation System (ITS) is proposed as a solution for traffic management and to address various challenges caused by increased vehicular density. Traffic analysis from real time camera images is the most adopted method for traffic state estimations which is a most important component in smart traffic management. The current traffic density estimators from videos estimate the volume of vehicles or percentage of area occupied in the lane. But for intelligent traffic management, more accurate traffic state estimation in terms of classification, density, speed, and flow rate for different categories of vehicles in different segments of the lane is needed. This work introduces a deep learning-based fine-grained vehicle flow analysis from traffic videos. A fine-grained traffic density distribution for different categories of vehicles over the entire coverage area of the lane with flow information is referred to as a dynamic traffic state map. A continuous traffic state map integrating vehicle categorization, lane density estimation, vehicle flow estimation, and orientation flow analysis. Vehicle categorization based on high and low-level features is proposed instead of area-based thresholding. A novel deep learning-based vehicle density estimation integrating coherence-based region segmentation with convolutional neural network and density estimate from the segment is proposed. The solution is able to provide an estimate of the traffic at a fine-grained level.**

***Keywords*—lane density, deep learning, traffic state mapping, vehicle categorization, flow analysis, traffic estimator**

I. Introduction

An Intelligent Transportation System (ITS) is an advanced application which aims to provide innovative services relating to different modes of transport and traffic management. It enables users to be better informed and make safer, more coordinated, and "smarter" use of transport networks. It can be viewed as a multi criteria optimization problem in a dynamic traffic environment under many constraints. Estimation of traffic state is an important component of intelligent transportation systems and it affects many decisions related to traffic scheduling. Traffic state is represented in terms of three parameters: density, speed, and flow. Density represents the number of vehicles occupying the road segment. Speed is the average speed of vehicles passing a given location within a time interval. The flow is defined as the volume of vehicles passing a specific location within a given time interval. Intelligent traffic management aims to improve traffic conditions in peak hours, congestion, and emergencies. It is necessary to detect abrupt changes in the traffic state, such as lower speeds, flows, and higher densities. This information is used as input to apply a suitable control strategy. Thus, an accurate traffic state estimation is important for intelligent traffic management.

Modern transportation systems involve a wide range of tasks, including vehicle classification, lane density estimation, traffic state map construction, flow analysis, and traffic estimation. Accurate identification of vehicles is crucial for traffic management systems to function efficiently; while lane density estimation, traffic state map construction, and flow analysis help traffic engineers make informed decisions about road design, traffic signal timings, and other infrastructure related decisions. Despite the advances in computer vision and deep learning techniques, vehicle identification remains a challenging problem due to the variations in vehicle types, lighting conditions, and occlusions. In this study, we aim to develop a system that can accurately identify different vehicle types and estimate lane density in real time.

Computer vision and deep learning techniques, including Convolution Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, are being widely used for vehicle detection, tracking, and classification, as well as for traffic flow analysis and prediction. In the context of traffic management, CNNs have been used to classify vehicles into different categories based on their visual characteristics, such as size, shape, and color. They have also been used to estimate lane density and monitor traffic flow patterns in real time. A CNN classifies vehicles into predefined categories and analyzes images from cameras mounted on the road network for lane

Manuscript received September 22, 2022; revised April 10, 2023; accepted April 21, 2023; published December 7, 2023.

doi: 10.12720/jait.14.6.1354-1364

density estimation and traffic state map construction. The CNN is trained on a large dataset of vehicle images to learn the patterns and features necessary for accurate classification.

Traffic analysis from real-time camera images is the most adopted method for traffic state estimations due to many reasons like easy installation, low cost, and portability. Many methods have been proposed (detailed in the literature) to estimate traffic density. The current methods only provide an estimate in terms of the number of vehicles on the lane or the volume of lanes occupied. But this information alone is not sufficient for intelligent traffic management. A more accurate traffic state estimation in terms of density, speed, and flow for different categories of vehicles in different segments of the lane is needed. This work deals with this need and proposes a deep learning-based image analysis to estimate the traffic state with consideration for vehicle category and different lane segments. Compared to the existing method this work applies traffic segmentation at a fine-grained level and applies deep learning in each segment. Also using continuously captured images, the speed, and flow is estimated for different category of vehicles over different segments to create a traffic state map for the lane. This kind of dynamic traffic map is very useful for road conditions in India, where is no separate lane for each vehicle category. In such cases, traffic management needs more fine-grained information about the distribution of different vehicle categories on the lane.

Following are the contributions of this work:

- A continuous traffic state map applying an integration of vehicle categorization, density estimation, and orientation flow analysis is proposed. To the best of our knowledge, there has no such traffic state map proposed in earlier work with concise information for high-density traffic videos
- Vehicle categorization based on high and low level features is proposed instead of area based thresholding. This solution can work for any scale of frames in the videos.
- A novel deep learning-based vehicle density estimation integrating coherence-based region segmentation with convolutional neural network and density estimate from the segment is proposed. The solution is able to provide an accurate estimate of the traffic at a fine-grained level.

This work is organized as follows. Section II provides a detailed review of related work in the field of vehicle classification and lane density estimation. Section III presents the proposed solution for dynamic traffic state mapping, which involves different stages and traffic state map construction overall process flow. Section IV evaluates the system's performance on three different datasets, Section V concludes the paper, and Section VI discusses future work.

II. Related Work

The existing solutions on traffic flow estimation from videos and the issues in them are presented in this section.

The problem of traffic flow estimation from the video surveillance camera has proposed. A system to count and classify vehicles by driving direction. A faster Region-based Convolutional Neural Networks (R-CNN) detector along with a Simple Online and Realtime Tracking (SORT) detector was used. The vehicles were classified only based on the direction of travel without fine grained classification like heavy or light vehicles. The work only considered the scenario of crossroads and could not give details like vehicles waiting time at signals [1]. A deep Residual Network (ResNet) was used for the classification of vehicles. An ensemble of joint Fine Tuning (JF) and Drop CNN was trained for classifying eleven different vehicles like truck, bus, cycles, etc. Even though the approach was able to detect the type of vehicle, it was not able to track them and the detection failed in case of occlusions [2]. The both deep learning and optimization-based methods explored to address the challenges in traffic density estimation from real-world webcam data [3]. The Optimization-Based Model (OPT-RC) embeds scene geometry through the rank constraint of multiple block regressors and motivates the deep-learning model FCN-MT. FCN-MT shares the idea of mapping local features into vehicle density with OPT-RC, while it replaces the BG subtraction, feature extractor, and block Regressors with fully convolution networks. Both methods avoid detecting or tracking individual vehicle and adapt to different vehicle scales. The approach is not effective against occlusion and it is able to provide fine-grained density. Applied two methods of solutions for the problem of extremely overlapping vehicle counting [4]. The first family of solutions is based on counting instances of objects by scanning the image space. A detector trained using local image features was used in the scanning process. The second family of solutions count objects by learning a direct mapping from low-level imagery features to object density.

Araniosuto and Oumaabsaloms proposed a solution for vehicle detection for three cases of free-flowing, slow moving, and stationary scenes. Laplacian of Gaussian (LoG) edge detection and mathematical morphology are applied to image frames and their negatives to extract the vehicles and count them [5]. The solution was not effective against occlusions and it also results in over segmentation for larger vehicles. A vision-based approach for traffic density estimation is proposed. Estimation is done in terms of calculating the amount of road surface that is occupied. The region of interest in the lane is divided into blocks [6]. The block size is smaller than the length of the smallest vehicle. The variance of pixel intensities over the blocks across the frames is used as criteria to decide the occupancy of the blocks. This method to classify the traffic density into light, medium, and heavy based on the pixel distribution of images. The entire image is split into several cells, and then a region of interest is extracted based on finding pixel varieties in each of the cells. A feature descriptor called histogram of multi-scale block local binary pattern is extracted from the transformed cell images. This feature is used to train a

Support Vector Machine (SVM) classifier to classify the traffic as light, medium, and heavy [7].

Kurniawan proposed a traffic density algorithm based on measuring the road area occupancy. A background construction algorithm classifies the image into foreground and background parts. An edge detection algorithm is then run on foreground parts to calculate the occupied area in the road and based on it, traffic density is estimated [8]. A novel traffic density method. It classified traffic density into light, medium, and heavy based on static and dynamic features extracted from videos. Texture and edge histograms of frames are used as static features. Interframe pixel differences and edge differences are the dynamic features. An SVM classifier is trained using both static and dynamic features. The method was robust to environmental conditions [9]. A traffic density estimation method based on comparing the texture differences between the congested image and unobstructed image is proposed, A texture feature based on the energy and entropy in Gray Level Cooccurrence Matrix (GLCM) is obtained for the images. A threshold for the texture feature is learnt from different congested and unobstructed images. Based on this threshold, a road is decided whether to be congested or not. Even though this method is quite fast, it cannot provide a traffic state [10].

Chakraborty [11] proposed two different deep-learning techniques to detect traffic congestion from camera images. The entire image was trained to provide the traffic congestion in terms of the percentage of occupancy [12]. Traffic density estimation based on image segmentation with adaptive threshold is proposed by Nguyen and Nguye proposed a histogram analysis-based traffic density estimation algorithm. Features are extracted from the histogram of different traffic images and these features are used to train a neural network to provide a categorical traffic density estimation [13]. Extracted various image features from traffic images such as sift corner, gray histogram variance, gray level co-occurrence matrix of energy, and contrast. The extracted features are used to train a neural network to classify traffic congestion [14]. Impedovo *et al*. [15] analyzed several states of art object detectors, visual features and classification models used in traffic state estimations. Four different visual descriptions for describing the traffic state total vehicles, traffic velocity, traffic volume, and road occupancy was extracted from the video frames using background suppression and morphological transformations [15]. Shi *et al*. [16] estimated traffic state using a learning-based aesthetic model. Multiple video-based perceptual features are extracted and these features are used to train a random forest classifier to classify traffic conditions [16].

Nguyen *et al.* [17] proposed a method to classify the traffic congestion patterns on road. From the road images, Speeded Up Robust Features (SURF) are extracted. The extracted features are represented in a low dimensional vector by applying bag of features technique. A multiclass SVM is trained using these low-dimensional vectors to classify traffic congestion [17]. A method for estimating traffic density without extraction of individual vehicles is proposed. Spatial Interest Points (SIPs) which represent the number of vehicles and Spatiotemporal Interest Points (STIPs) which represents the moving vehicles are extracted from the video stream. A Gaussian mixture model estimates the traffic density using the SIPs and STIPs. A real-time traffic congestion detection using the image correlation coefficient of consecutive images is proposed [18]. A new image measure called image correlation coefficient is proposed in this work which is extracted based on Harr features. A threshold for the image correlation coefficient is learnt from multiple images to classify the road condition to normal or congested [19]. Traffic density estimation based on macroscopic and microscopic parameters extracted from video sequences is proposed. Microscopic parameters are extracted using motion detection and tracking methods. Macroscopic parameters are extracted by analyzing global motion in video scenes. Three different classifiers (K-Nearest Neighbors (KNN), Learning Vector Quantization (LVQ), and SVM) are trained using the extracted features to classify the road traffic into light, medium,and heavy [20].

III. Methodology

The proposed solution used images of traffic over the road lanes. These images are captured by traffic surveillance cameras. The frame is selected for analysis when a traffic management decision has to be made. The captured frame is analyzed to construct a dynamic traffic state map. The flows in the constructed dynamic traffic state map are then updated by correlation analysis with the previous frame.

The proposed solution for dynamic traffic state mapping involves the following stages

- Segmentation
- Density estimation
- Traffic state map construction

In the segmentation stage, the vehicles on the lane are segmented into vehicle clusters. A modified Extended Topological Active Net-based (ETAN) segmentation algorithm is proposed to segment the vehicles. The vehicle clusters are categorized into three vehicle classes small, medium, and heavy. Traffic density in each category of vehicle is estimated from the segments using hybrid handcrafted and deep learning features. A traffic state map is constructed for each category of vehicles with parameters like density, waiting time, etc. This traffic state map is updated dynamically after analysis of each subsequent frame of traffic video. Each of the stages is detailed below subsections.

A. Segmentation

The first step in dynamic traffic state mapping is the segmentation of traffic on the lane based on categories of vehicles. Segmentation is done into three categories of Small, Medium, and Heavy. Two-wheelers are in a Small category. Three-wheelers, cars, and mini-trucks are in the medium category. Buses and Lorries are in the Heavy category. The segmentation is based on splitting the lane into a cluster of grids where each cluster represents a vehicle category.

The traffic image of the lane must be preprocessed before segmentation. The color image is converted to a gray scale. To compensate for the non-uniform illumination of the scene, morphological top-hat transformation is done on the grayscale image. The transformation is done as:

$$g(x,y) = f(x,y) - \big(f(x,y) \times b(x,y)\big) \quad (1)$$

g (*x*, *y*) is the lane image without any vehicles. *f* (*x*, *y*) is the traffic image of the lane. $f\,(x, y) \times b\,(x, y)$ is the morphological opening of *f* (*x*, *y*) using structural element *b*(*x*, *y*). After transformation, the traffic image of the lane is smoothed using a median filter to remove any noise. The image is then contrast enhanced using a contrast stretching algorithm prior to segmentation. On the contrast-enhanced image, Laplacian of a Gaussian (LoG) is applied. The LoG on the image is given as:

$$\nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right].e^{\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where ∇^2 is the Laplacian operating on the Gaussian smoothed image $G(x,y)$ and σ is the standard deviation of image pixel intensities.

The grid mask of size N × N is then applied to the image resulting from LoG. The segmentation of the grid is done by applying a modified Extended Topological Active Net-based (ETAN) segmentation [21]. Original ETAN works by placing the grid mesh over the whole image and applying energy minimization on each node in the mesh to deform it and fit the objects. A greedy search is done to find the optimal solution. The energy of the node and its neighborhood in its current position is computed at each step. Node moves to the position of lowest energy value. The algorithm stops when nodes cannot be moved further. Original ETAN cuts the links based on the energy threshold. The links energy is calculated as:

$$E_{link} = (\textstyle\sum_{p\varepsilon A} DG_{evfc}(p).\frac{I(p)}{I_{max}}) \quad (3)$$

where E_{link} is the energy of the link, A is an area for which energy is computed and p is the pixel in that area. The original image is given as *I* and the maximum intensity value is given as I_{max}

The cutting threshold is fixed as the mean energy of the links. Holes are identified starting from misplaced internal nodes. For every internal node $r(n)$ is calculated as:

$$r(n) = \frac{E_{ext}(\mathrm{n})}{E_{ext}(\mathrm{n}) + E_{int}(\mathrm{n})} \quad (4)$$

where E_{ext} is the external energy, and E_{int} is the internal energy of the node *n*. The node with highest ratio is selected and if its *r* value is higher than threshold for holes.

The energy for all links starting from this node and to the neighbors is calculated and the highest one is removed. In this way more links can be deleted from the mesh and adjusted to the edge of the object.

The behavior of the original ETAN has to be modified to consider the case of the same category of vehicles in an area called a vehicle cluster. In this case, the original ETAN segments each vehicle into a separate segment, but this has to be modified to consider vehicle clusters into one segment. This work proposes a fuzzy extension to ETAN for vehicle clustering. The links of the grid mesh can be categorized below

- Links Completely within the vehicle cluster
- Links at the boundary of vehicle cluster and background.
- Links in the background

The links at the boundary must be removed, so that *the* remaining links represent the vehicle cluster. To speed up the process of removing the links, the links must be first classified. A Fuzzy rule-based classifier is trained using a training set of images whose links results are known and expressed as ground truths. The fuzzy rules are derived in the form of features of the link against the class label for the link. Each class label represents one of the three link positions defined above. The Gaussian function is generated as an output class label. Representing as a Gaussian function is advantageous as the parameters of the function can be updated simultaneously during the training and the membership of the link is modeled as a fuzzy decision with the probability of presence in each class instead of making a hard decision of associating a link to a particular class.

For each link, four different texture features are extracted. It is calculated along 4 axes as shown in Fig. 1 with the center point on the link (DV, DV', DH, DH'). Along each axis, a textured pattern is calculated and the feature of a link is represented as a feature vector.

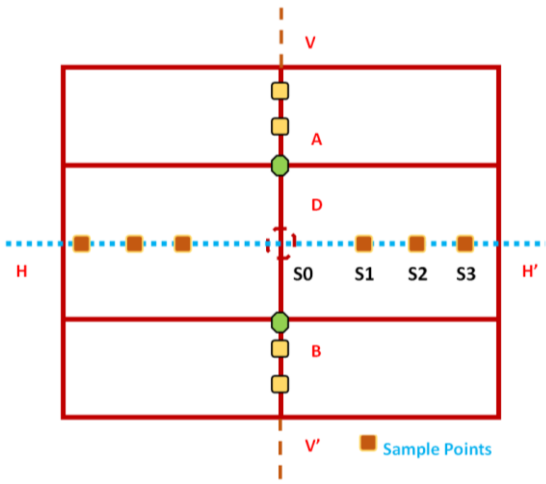


Figure 1. Feature vector of links.

The training dataset is created with a feature vector of link as input and the label [0, 1, 2] as output. Fuzzy C

Means clustering done on the dataset with C = 3. The cluster center after the fuzzy C means clustering is defined as C = {C_e, q, e = 1, 2, 3 and q = 1, 2, 3, 4, 5}, where C_e, q is the q^{th} coordinate of the eth cluster.

The closeness of the q^{th} feature of the r^{th} link f, r, q with q^{th} coordinate of e^{th} cluster is defined using the Gaussian function as

$$G\left(f_{r,q}, C_{e,q}, \sigma_{e,q}\right) = e^{\frac{(fr,q-Ce,q)^2}{\sigma e,p^2}} \tag{5}$$

where, the closeness of features of r^{th} link to the e^{th} cluster is givenas:

$$\sigma_{e,q} = \frac{1}{N_e}\sum_{r=1}^{N_e}(f_{r,q} - C_{e,q})^2 \tag{6}$$

The closeness of features of r^{th} link to the e^{th} cluster is given as:

$$\Psi_{r,e} = \prod_{q=1}^{P} G\left(f_{r,q}, C_{e,q}, \sigma_{e,q}\right) \tag{7}$$

The output label for e^{th} cluster is found from the linear regression of input features $f_{r,q}$ as:

$$\Phi_{r,e} = W_{e,0} + \sum_{q=1}^{P} W_{e,q,f_{r,q}} \tag{8}$$

where W is the regression coefficient of the e^{th} cluster. Since each of the r^{th} link has membership value to all three clusters, final label of that particular link is given by weighting the label of the link with its membership value as:

$$\bar{N}(r) = \sum_{e=1}^{3} \Psi_{r,e}\Phi_{r,e} \tag{9}$$

The value of $\bar{N}(r)$ calculated above may have an error with respect to $N(r)$ from training. The total error is calculated as:

$$E = \sum_{r=1}^{N} ||\bar{N}(r) - N(r)||^2 \tag{10}$$

The Gaussian parameters $C_{e,q}, \sigma_{e,q}$ and the regression coefficients $W_{e,p}$ are tuned to reduce the error defined above using gradient decent method.

$$\mathrm{C_{e,q}(t+1) \leftarrow C_{e,q}(t) +\ \eta_C \frac{\partial E}{\partial C_{e,q}}} \tag{11}$$

$$\mathrm{\sigma_{e,q}(t+1) \leftarrow \sigma_{e,q}(t) +\ \eta_\sigma \frac{\partial E}{\partial \sigma_{e,q}}} \tag{12}$$

$$\mathrm{W_{e,q}(t+1) \leftarrow W_{e,q}(t) +\ \eta_W \frac{\partial E}{\partial W_{e,q}}} \tag{13}$$

where t is the iteration number and $\eta_C, \eta_\sigma, \eta_W$ are the learning parameters. The iteration is stopped when error threshold is reached.

The segmentation decision is a fuzzy decision where the decision of a pixel is decided to be part of the object or in the background can be made an as a hard decision. So this work models the decision of an active link to be part of an object or background in terms of Fuzzy Gaussian membership function. From training, the Fuzzy Gaussian membership functions are obtained for each class. During the segmentation, for each link in the active mesh, the four features are calculated and from it, the fuzzy membership function for each class is found. The segmentation splits the traffic images into vehicle clusters

B. Density Estimation

The vehicle cluster must be categorized to one of three categories of small, medium and heavy and the density must be estimated for each cluster area. The different metrics used to classify vehicle density into small, medium, and high. IT varies depending on the specific application and context.

Number of Vehicles in a lane: These metric measures the number of vehicles present in a Lane. This metric used to classify vehicle density levels into small, medium, and high. Occupancy: Occupancy measures the percentage of a roadway or area that is occupied by vehicles. Higher occupancy values indicate higher vehicle density. Flow rate: Flow rate measures the rate at which vehicles pass through a given area. Slower flow rates can indicate higher vehicle density. Queue length: Queue length measures the length of any queues or backups of vehicles that are waiting to travel through a given area. Longer queue lengths can indicate higher vehicle density. This is identified since the last change in the frame while processing images of traffic.

A deep learning model integrating a coding network with multilayer perceptron is used for classifying the vehicle cluster to one of Small (S), Medium (M), and Heavy (H). The advantage of this model is that it combines the high-level features learned from the convolutional neural network and texture and color moment features for classification. A Convolution Neural Network (CNN) is trained in supervised mode to convert raw pixels of the image to high level feature vectors. Texture and color moment features are also extracted from segments. These two features are fused and the best features are selected using entropy correlation.

A classifier is trained using the selected features to categorize the vehicle cluster. Due to the use of both hand-crafted (texture, color moment) features and deep learning features, the classification accuracy can be improved. The flow for the categorization of vehicle clusters is shown in Fig. 2. A convolutional neural network is used as a coding network to convert vehicle cluster images to high-level features. CNN has a series of convolution and pooling layers. The convolution operation on the input image is given as:

$$y_j^r \leftarrow f(b_j^r + \sum w_{i,j}^{r-1} * x_i^r) \tag{14}$$

Here r is the corresponding layer in the CNN and f is the activation function. x_i is the i^{th} input feature map and y_j is the j^{th} output feature map. $w_{i,j}$ is the convolution kernel between x_i and y_j . b_j is the bias and $*$ is the convolutional operation.

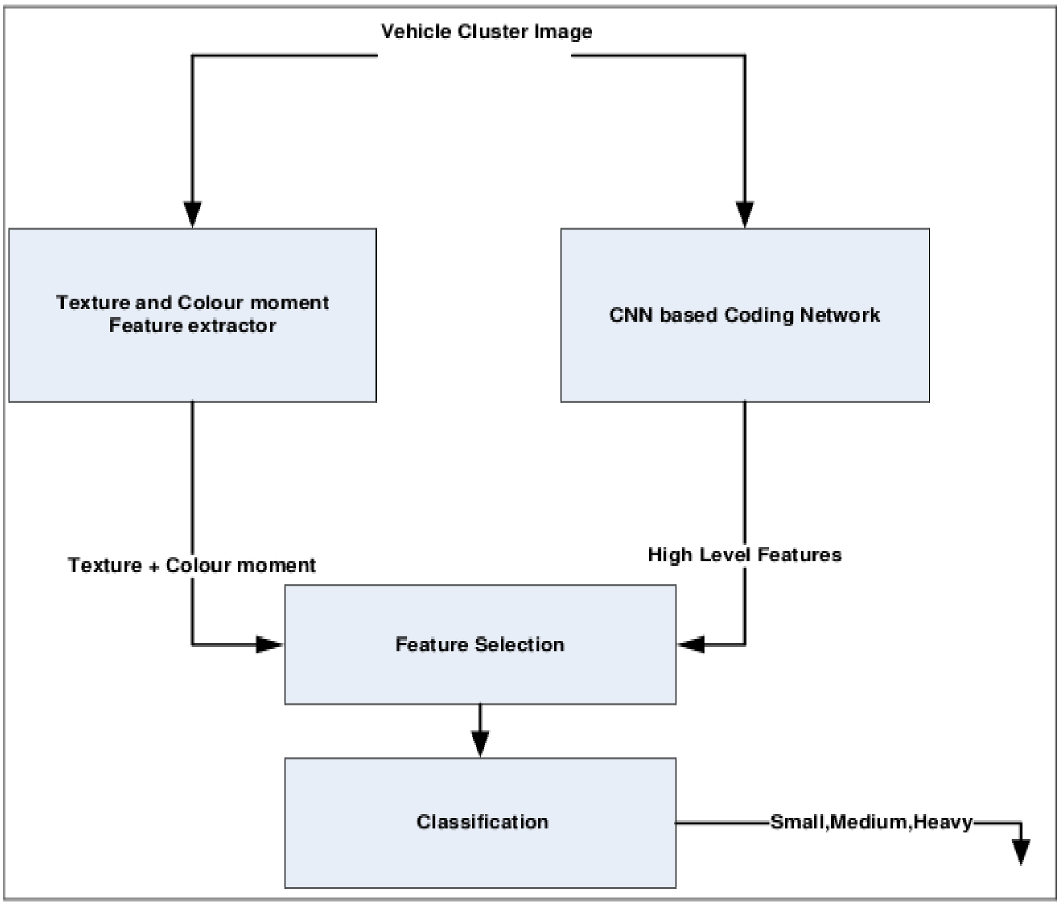


Figure 2. Classification of vehicle cluster.

For the activation function, this work uses Rectified Linear Units (ReLus) proposed, due to its faster convergence rate [22]. For extracting the texture feature of the vehicle cluster image, Gray Level Cooccurrence Matrix (GLCM) matrix G is calculated for the image, and features defined in Table I are extracted. From the input vehicle cluster image, three color moment features of mean, standard deviation, and third-order color moment are extracted as defined in Table I.

TABLE I. HAND CRAFTED FEATURES

Angular Second Moment (ASM)	$\sum_{i}^{s}\sum_{j}^{s} G(i,j)^2$
Entropy (ENT)	$-\sum_{i}\sum_{j} G(i,j)\log(G(i,j))$
Contrast (CON)	$\sum_{i}\sum_{j} G(i,j)(i-j)^2$
Correlation (COR)	$\frac{\sum_i^s \sum_j^s ijG(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$
Mean [Color moment]	$A_i = \frac{1}{N}\sum_{j}^{N} P(i,j)$
Standard deviation [Color moment]	$V_i = \sqrt{\frac{1}{N}\sum_{j}^{N}(P(i,j)-A_i)^2}$
Third order color moment [Color moment]	$S_i = \sqrt[3]{\frac{1}{N}\sum_{j}^{N}(P(i,j)-A_i)^3}$

Both the high level and low-level features are fused using weighted fusion as below

$$TF \leftarrow \alpha * LF + (1-\alpha)HF \tag{15}$$

where LF is the low-level feature (texture and color moment) and HF is the high-level feature (feature learnt using CNN). α is the user configured weight value from 0 to 1. From the fused features, the best features are selected using entropy correlation [23].

The selected features are passed to a SoftMax layer for classification. This layer can predict *n* different classes by computing the probability of belonging to each category.

$$p(y=j|x,\theta) \leftarrow \frac{e^{\theta_j^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}} \tag{16}$$

K is the number of classes (in this work k = 3) and θ_j^T is the weight vector. Once each vehicle cluster is categorized, the density is estimated for the vehicle cluster based on area occupancy. The number of grids occupied by a single vehicle in each category of Small, Medium, and Heavy is measured as N_s, N_m, N_L. The density of vehicle cluster is estimated as:

$$D(VC_S) \leftarrow \frac{total\ grids\ in\ VC}{N_S} \tag{17}$$

$$D(VC_M) \leftarrow \frac{total\ grids\ in\ VC}{N_M} \quad (18)$$

$$D(VC_L) \leftarrow \frac{total\ grids\ in\ VC}{N_L} \quad (19)$$

C. Traffic State Map Construction

A traffic state map is constructed for lanes with information on each vehicle cluster and its density. Each vehicle cluster is shown in a different color depending on the category and density estimation is tagged for each vehicle cluster. Different from other traffic flow analysis systems which provide flow results in terms of incoming and outgoing traffic, this work provides a fine-grained traffic analysis in terms of categories of vehicles and waiting time in traffic for each category (see Fig. 3).

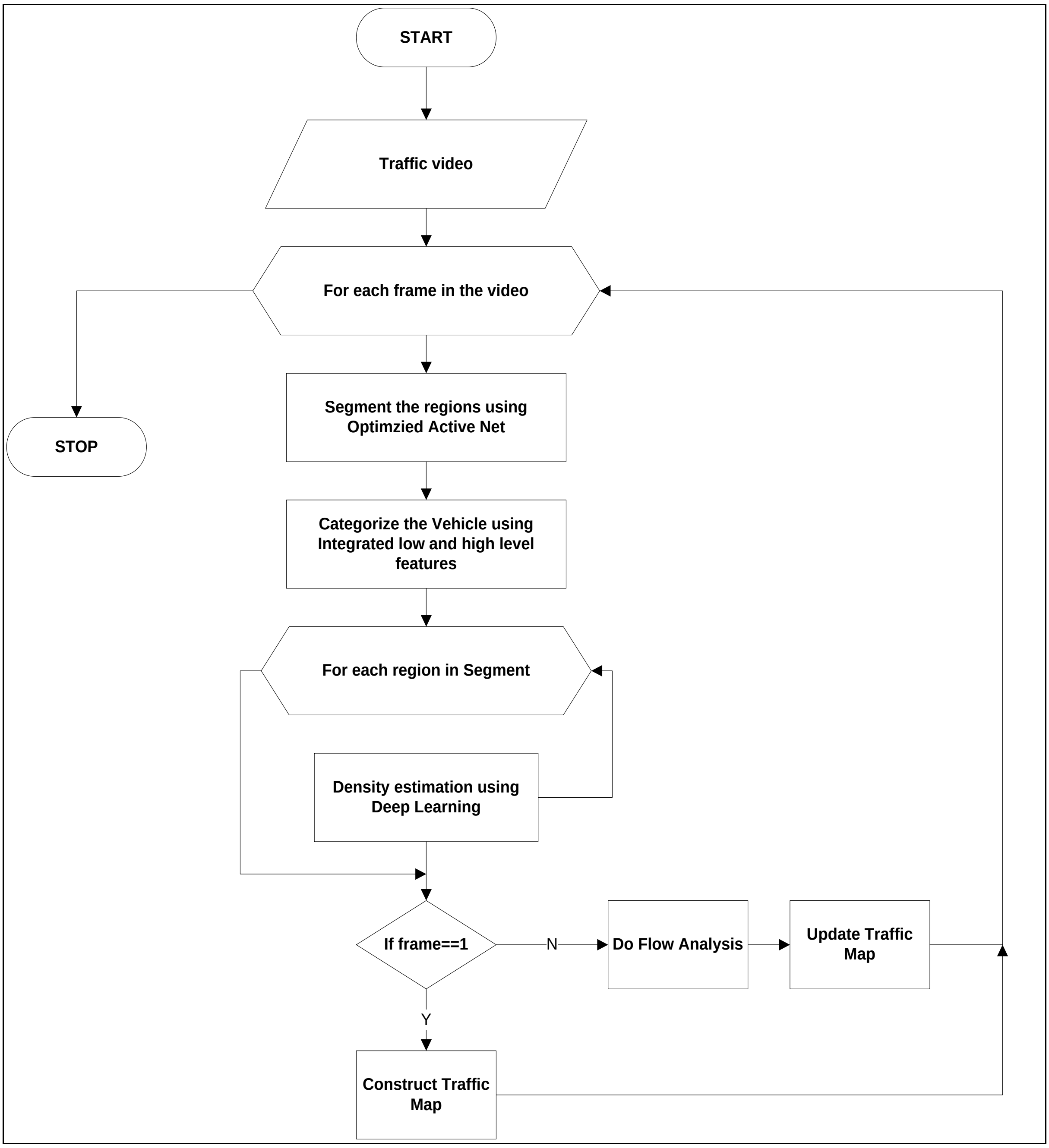


Figure 3. Traffic state map construction overall process flow.

A traffic state map is constructed for each lane at the intersection of the traffic signal. Each vehicle category is given a color code. A mesh of grids is placed over the lane covering the entire lane. Each grid is given a color code based on the vehicle category covered by the grid. In addition to the color code for the grid, the waiting time over the grid is calculated and displayed in the grid. A sample traffic state map for a lane is shown in the Fig. 4 from the traffic state map.

Fig. 4 shows the traffic state map construction, The rate of change of traffic and traffic density distribution over each lane is known at a fine-grained level and this enables more sophisticated traffic management or scheduling. The first step in dynamic traffic state mapping is the segmentation of traffic on the lane based on categories of vehicles. Segmentation is done on the 3 categories (1). Small, (2). Medium, (3) Large. Two-wheelers is in a small category. Three-wheelers, cars, and mini-trucks are in the medium category. Buses and Lorries are in a large category. The segmentation is based on splitting the lane into many grids and tagging each grid based on the vehicle covered majority in the grid. The traffic image is split into a grid and each grid image is given as input to the Convolutional Neural Network (CNN) to classify the vehicle category in the grid into Small, Medium, Large, and empty.

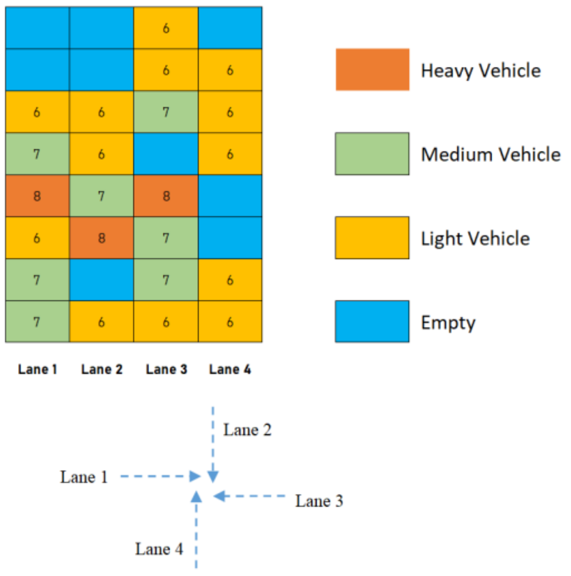


Figure 4. Lane density in traffic state map estimation.

Flow analysis is done by correlating two traffic state map- TSM_t and TSM_{t-1} . The correlation looks for vehicle cluster VC in TSM_t and TSM_{t-1} with same shape or with minimum distortion. This can be done by SURF feature mapping between the two traffic state maps of TSM_t and TSM_{t-1} and identify vehicle cluster present in TSM_t and its corresponding presence in TSM_{t-1}. If the corresponding vehicle cluster cannot be detected, then it means the vehicle cluster has moved out from lane in time between TSM_t and TSM_{t-1} . For each unmoved vehicle cluster, the weighting time is aggregated with time between TSM_t and TSM_{t-1} . In addition, the difference in density of vehicles in each category of small, medium, and large is calculated as below:

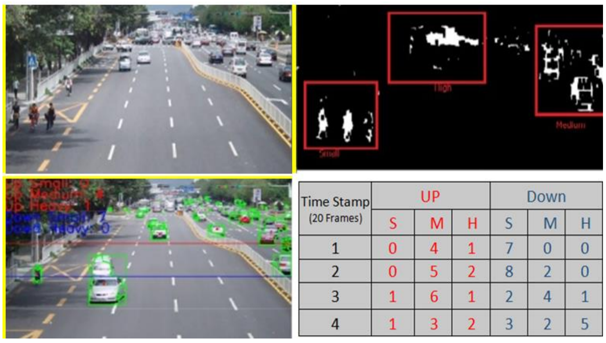
$$DD_s \leftarrow \sum_{TSM_t} D(VC_S) - \sum_{TSM_{t-1}} D(VC_S) \qquad (20)$$

$$DD_M \leftarrow \sum_{TSM_t} D(VC_M) - \sum_{TSM_{t-1}} D(VC_M) \qquad (21)$$

$$DD_L \leftarrow \sum_{TSM_t} D(VC_L) - \sum_{TSM_{t-1}} D(VC_L) \qquad (22)$$

The traffic state map is continuously updated on processing every frame. The dynamic traffic state map marks each vehicle cluster in different categories and tags its density, and waiting time. The traffic state map is tagged with the overall density difference in each category. The overall process flow of the proposed solution so far discussed is summarized in Fig. 3.

As shown in Fig. 5, the density is estimated for the small, medium, and heavy traffic regions marked in the segmented image with a counter for each. The number of vehicles moving up in the lane is shown for each category of Small, Medium, and High is Number of vehicles moving down in the lane is shown for each category of Small, Medium, and High with a counter for each. The traffic state map is generated as Excel sheet with a traffic state counter shown for every 20 frames.



Time Stamp (20 Frames)	UP			Down		
	S	M	H	S	M	H
1	0	4	1	7	0	0
2	0	5	2	8	2	0
3	1	6	1	2	4	1
4	1	3	2	3	2	5

Figure 5. Traffic state map construction.

IV. Results

The performance of the proposed (Dynamic traffic state mapping using Deep learning) system is tested against Indian Driving Dataset [24], MIO-TCD classification dataset [25] and TrafficDB dataset [26].

A system that utilizes IoT sensors and data analytics to provide real-time information about road and weather conditions to drivers, which can help them, make informed decisions while driving, the system consists of three main components: the data acquisition module, the data processing and analysis module, and the alert and feedback module [27]. The authors tested the system using simulation experiments and reported promising results. The article presents an interesting approach to using IoT technology to improve road safety. However, further research is needed to evaluate the system's effectiveness in real-world scenarios and its potential impact on reducing accidents caused by adverse weather and road conditions. In our work image take the major rule than the sensor.

A smart city system for intelligent traffic congestion control in vehicular networks using machine learning techniques. The system uses a fusion-based approach, which combines data from different sources such as vehicle sensors, GPS, and traffic cameras to provide real-time traffic information. The authors propose a machine learning algorithm based on fuzzy logic, which can accurately predict traffic congestion levels and adjust traffic signals accordingly to optimize traffic flow. The article presents an innovative approach to addressing traffic congestion in smart cities, highlighting the potential benefits of using machine learning techniques in traffic control systems [28].

A valuable contribution to the ongoing debate on the role of automation in transportation safety. It highlights the potential benefits of using automated vehicles, including the ability to reduce accidents caused by human error [29]. The article provides a useful overview of the potential benefits of using vehicle automation and following models for accident avoidance. It is recommended for researchers and practitioners interested in the development of automated vehicle systems [30].

Indian Driving dataset was collected as part of the funded project by Intel in November 2017 to collect traffic information in Hyderabad and Bangalore. The dataset has 10,000 images annotated with 34 classes and collected from 182 drive sequences on Indian roads. The MIO-TCD dataset consists of more than half a million images acquired at different times of day and different periods of the year by 8,000 traffic cameras deployed all over Canada and the United States. Those images have been selected to cover a wide range of localization challenges and are representative of typical visual data captured today in urban traffic scenarios. The TrafficDB dataset was used for the comprehensive evaluation of the proposed method. It consists of 254 videos 5 seconds each which have been annotated as light, medium, and heavy traffic respectively. Since there are no existing solutions for dynamic traffic state mapping, the performance of the solution is compared at different stages of the proposed solution [31].

The performance is compared against three stages

- Vehicle categorization
- Density estimation
- Flow estimation

The performance of the proposed vehicle categorization is compared against ResNet based classification scheme proposed in [2]. The performance is measured in terms of accuracy, precision, and recall.

Accuracy is measured as:

$$Accuracy = \frac{TP}{Total\ Test\ Images} \tag{23}$$

Precision is measured as:

$$Precision = \frac{TP}{TP+FP} \tag{24}$$

Recall is measured as:

$$Recall = \frac{TP}{TP+FN} \tag{25}$$

where *TP* is true positive, *FP* is false positive and *FN* is false negative.

The performance is measured for 1000 images from Indian Driving Dataset and the result is given in Table II.

TABLE II. PERFORMANCE FOR INDIAN DRIVING DATASET

	ResNet	Proposed
Accuracy	0.9695	0.9825
Precision	0.9530	0.9730
Recall	0.8970	0.9320

The proposed solution has 1.3% more accuracy than ResNet, 2% more precision than ResNet, and 3.5% more recall than Res Net.

The performance is measured for 1000 images from the MIO-TCD dataset and the result is given in Table III.

TABLE III. PERFORMANCE FOR MIO-TCD DATASET

	ResNet	Proposed
Accuracy	0.9395	**0.9525**
Precision	0.9230	0.9430
Recall	0.8670	**0.9220**

The proposed solution has 1.3% more accuracy than ResNet, 2% more precision than ResNet and 5.5% more recall than Res Net.

The performance is measured for 10 videos from TrafficDB dataset and the result is given in Table IV.

TABLE IV. PERFORMANCE FOR TRAFFICDB DATASET

	ResNet	Proposed
Accuracy	0.9195	0.9425
Precision	0.9130	0.9380
Recall	0.8470	0.9140

The results show an increased value of accuracy, precision, and recall in the proposed solution compared to ResNet [2]. The accuracy of vehicle classification is 1.4% higher in the proposed solution compared to ResNet [2]. The use of both coding networks derived high-level features and traditional low-level features with Deep Learning based classification has increased the accuracy in the proposed method compared to [2].

The performance of the proposed traffic density estimation is compared against Image ROI Analysis [8] and Adaptive threshold [12]. The performance is measured in terms of Mean Square Error (MSE) between the estimated and actual density. The MSE results are given in Table V.

TABLE V. MSE COMPARISON

	MSE forIndian Driving Dataset	MSE for MIO-TCD Dataset	MSE for TrafficDBDataset
ROI Analysis	8.145	9.145	9.345
Adaptive threshold	7.913	8.125	8.543
Proposed	3.121	3.675	3.986

The MSE in the proposed solution is 61.6% lower than ROI analysis [8] and 60.5% lower than adaptive threshold [12]. The performance of proposed flow estimation is measured against R-CNN tracker [1]. The performance is measured in terms of mean absolute error and given in Table VI.

TABLE VI. ERROR COMPARISON

	Mean absoluteerror for Indian driving dataset	Mean absolute errorfor MIO-TCDdataset	Mean absolute errorfor TrafficDBdataset
R-CNN	34.178	36.274	38.876
Proposed	20.12	21.14	22.34

The results show that flow estimation error in the proposed solution is comparatively lower than R-CNN [1] due to estimation in terms of category and aggregating them. Due to this, effect of background noise in flow estimation is avoided in the proposed solution (see Fig. 6).

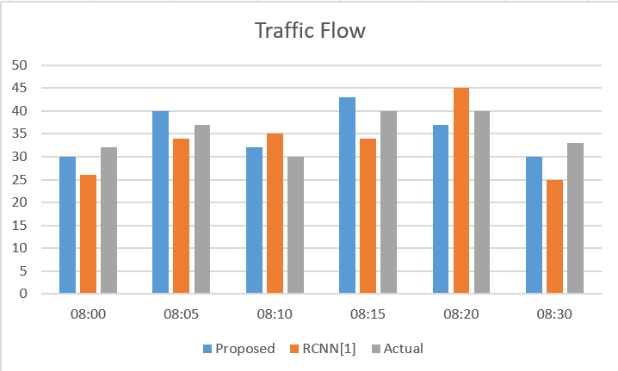


Figure 6. Traffic flow over time.

The traffic flow is estimated for every five minutes and the result is compared between proposed, R-CNN [1] and the actual for Indian driving dataset. The comparison is given in Fig. 5. From the results, it can be seen that the proposed solution estimation is very close to actual compared to R-CNN [1].

V. Conclusion and Future Word

A Deep learning model for dynamic traffic state mapping is proposed in this work. Fuzzy modified topological active net segmentation is proposed to segment the traffic image to vehicle clusters. A deep learning model integrating coding network-based high-level features and low-level features is proposed to classify the vehicle cluster and then estimate the density. Correlation based on SURF features is employed on vehicle clusters to build dynamics to the traffic state map. The performance of the proposed solution is compared in three dimensions vehicle categorization, density estimation, and flow estimation. The proposed solution is found to perform better in all three dimensions of vehicle classification, density estimation and flow analysis. The accuracy of vehicle classification is at least 1.3% higher than existing work. The density estimation compared in terms of MSE is 61.6% lower and 60.5% lower than existing works. Flow estimation measured in terms of MAE is 41% lower in the proposed solution compared to existing works. The approach can be further extended for the case of other vehicle categories like emergency vehicles. There is a potential to extend the current work to include different kinds of emergency vehicles and prioritizing a lane could have a significant impact on improving traffic management and emergency response times. This could involve the use of specific visual features to identify emergency vehicles, and the application of machine learning algorithms to optimize traffic flow while minimizing delays for emergency vehicles. The traffic state map constructed in the current work could also be used to inform the prioritization of lanes for emergency vehicles.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Arunakumar Joshi conceptualized the study, designed the research methodology, and performed data collection analysis, and implementing the deep learning techniques, also contributed to the interpretation of the results and drafted the initial manuscript. Dr S. B. Kulkarni monitored as the research guide, by providing valuable supervision and guidance in refining the research objectives, methodology, and data analysis, also reviewed and edited the manuscript, contributing to the intellectual content and ensuring its accuracy. The research was conducted at SDM College of Engineering and Technology Research Center Dharwad, where both authors actively collaborated and shared ideas to advance the study's objectives, and approved the final version of the manuscript for submission.

References

[1] A. Fedorov, K. Nikolskaia, and Ivanov, "Traffic flow estimation with data from a video surveillance camera," *J. Big Data*, vol. 6, 2019.

[2] H. Jung, M. K. Choi, J. Jung, J. H. Lee, S. Kwon, and W. Y. Jung, "ResNet-based vehicle classification and localization in traffic surveillance systems," in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, vol. 72.

[3] S. Zhang and G. Wu, "Understanding traffic density from large-scale web camera," *Data. Computer Vision and Pattern Recognition*, pp. 5898–5907, 2017.

[4] R. G. G. O. Olmedo, B. T. Jenez, R. L. Opezsastre, S. M. Bon, and O. Rubio, "Extremely overlapping vehicle counting," in *Proc. Iberian Conference on Pattern Recognition and Image Analysis*, 2015, vol. 216.

[5] D. Araniosuto and H. Oumaabsaloms, "Automatic road traffic density estimation using image processing," *Algorithms. Annual Conference on Sustainable Research and Innovation*, 2016.

[6] K. Garg, S. Lam, T. Srikanthan, and V. Agarwal, "Real-time road traffic density estimation using block variance," in *Proc. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[7] Z. H. Hu and Y. H. Sheng, "Traffic density recognition based on image global texture feature," *International Journal of Intelligent Transportation Systems Research*, pp. 171–180, 2019.

[8] F. Kurniawan and O. Harunosajati, "Image processing technique for traffic density estimation," *International Journal of Engineering and Technology*, 2017.

[9] M. P. Dinani, I. Ahmadi, and Gholampour, "Efficient feature extraction for highway traffic density classification," in *Proc. 9th Iranian Conference on Machine Vision and Image Processing*, 2015.

[10] L. Wei and D. H. Ying, "Real-time road congestion detection based on image texture analysis," *Procedia Engineering*, vol. 137, pp. 196–201, 2016.

[11] P. Chakraborty *et al.*, "Traffic congestion detection from camera images using deep convolution neural networks," *Transportation Research Record Journal of the Transportation Research*, vol. 2672, no. 45, 2018.

[12] L. A. T. Nguyen and T. N. T. Nguyen, "Traffic density analysis based on image segmentation with adaptive threshold," *Communications on Applied Electronics*, vol. 7, no. 19, 2018.

[13] L. A. T. Nguyen and T. N. T. Nguyen, "Traffic density identification based on neural network and histogram," *International Journal of Computer Applications*, vo. 172, no. 9, 2017.
[14] L. Guo and H. Cui, "Urban road congestion recognition using multi-feature fusion of traffic images," *Journal of Artificial Intelligence Practice*, vol. 1, no. 1, 2016.
[15] D. Impedovo, F. Balducci, and V. Dentamar, "Vehicular Traffic congestion classification by visual features and deep learning approaches: A comparison," *Dipartimento di Informatica*, vol. 19, 2019.
[16] X. Shi, Z. Shan, and N. Zhao, "Learning for an aesthetic model for esti- mating the traffic state in the traffic video," *Neurocomputing*, 2016.
[17] H. N. Nguyen *et al*., "Congestion pattern classification using multi-class SVM," in *Proc. the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, vol. 2.
[18] S. Chaudhary, S. Indu, and S. Chaudhury, "Video-based road traffic monitoring and prediction using dynamic Bayesian networks," *IET Intell. Transp. System*, vol. 12, no. 3, 2018.
[19] C. Lam, H. Gao, and B. Ng, "A real-time traffic congestion detection system using on-line images," in *Proc. IEEE 17th International Conference on Communication Technology*, 2017, vol. 15.
[20] O. Asmaa, K. Mokhtar, and Abdelaziz, "Road traffic density estimation using microscopic and macroscopic parameters," *Image and Vision Computing*, vol. 31, no. 11, 2013.
[21] N. I. Bova and Oscar, "Extended topological active nets," *Image and Vision Computing*, vol. 31, no. 12, 2013.
[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. International Conference on Machine Learning*, 2010, pp. 807–814.
[23] K. Zeng and X. Niu, "Feature selection with neighborhood entropy-based cooperative game theory," *Computational* Intelligence*and Neuroscience*, vol. 2, 2014.
[24] India Driving Dataset. [Online]. Available: https://idd.insaan.iiit.ac.in/
[25] Github Rpratesh. [Online]. Available: http://podoce.dinf.usherbrooke.ca/challenge/dataset/
[26] Highway Traffic Videos. [Online]. Available: http://visal.cs.cityu.edu.hk/downloads/trafficdb/
[27] J. Onesimu, A. Kadam, K. Sagayam, and A. Elngars, "Internet of things based intelligent accident-avoidance system for adverse weather and road conditions," *J. Reliab. Intell. Environ.*, pp. 1–15, 2021.
[28] M. Saleem, S. Abbas, T. M. Ghazal, M. A. Khan, and N. Sahawneh, "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 417–426, 2022.
[29] M. Ahmad, "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informatics Journal*, vol. 23, no. 3, 2022.
[30] A. Kadam, J. Andrew, K. M. Sagayam, and D. Hien, "Vehicle automation and car-following models for accident avoidance," *Prz Electrotechnics*,vol. 96, pp. 118–123, 2020.
[31] Z. Luo, F. Branchaud-Charron, C. Lemaire, and J. Konrad, "MIO-TCD: A new benchmark dataset for vehicle classification and localization," *IEEE Transactions on Image Processing*, vol. 27, no. 10, October 2018.