ISSN 2229-5518

Real-time Traffic Congestion Detection using Combined SVM

Sukriti Sharma, Vidya Meenakshi.K, Apurva VS, Saritha Chakrasali, Shreyas Satish.

Abstract— Traffic density and flow are important inputs for an intelligent transport system (ITS) to better manage traffic congestion. This paper proposes a cost effective traffic congestion detection system by using image processing algorithms and a combined Support Vector Machine (SVM) classifier and thereby demonstrates the efficiency of using key informative metrics like edge and texture features in real time. Effectiveness of the proposed method of classifying traffic density states is demonstrated on real time traffic images taken at different time periods of a day in Bangalore city.

Index Terms— Virtual loop detector (VLD), Traffic density, SVM Classifier, ITS, Feature extraction, Image processing, Combined classifier

1 Introduction

Heavy traffic congestion poses a big problem in almost all capital cities around the world, especially in developing regions resulting in massive delays, increased fuel waste and monetary losses..To reduce traffic congestion, various traffic management techniques are being developed continuously. ITS being one among them uses advanced applications which aims to provide innovative services relating to different modes of transport and traffic management thus enabling various users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks. Traffic density is required by the ITS to operate other higher level functionalities such as sequencing traffic lights and organizing control signals. The Loop Detector (LD) is one of the most widely applied techniques in detecting the traffic density. The paper by Luigi DI Stefano et al., "Evaluation of Inductive-Loop Emulation Algorithm for UTC Systems" uses LD. A loop detector is installed close to an intersection point and triggers some signals once a vehicle passes over the LD. The accumulated output signals from an LD estimates the traffic flow passing in the area where the LD is installed. However, an LD has limitations due to its difficulties in installation and maintaining [1].

To overcome the problem of LD, the paper by Zhidong Li et al., "On Traffic Density Estimation with a Boosted SVM Classifier " aimed to develop a Virtual Loop Detector (VLD) that simulates a real LD by using a machine learning approach to detect the traffic density in real-time. The paper proposes a boosted SVM classifier with some informative features to distinguish between a zero-occupancy (ZO) state and a non-zero-occupancy (NZO) state. Here, the traffic density is defined as the occupancy of vehicles on a road surface [1]. The outputs from the VLD are used to detect the traffic density at a given Region of Interest (ROI). However, the complexity of combining the results of individual SVMs is higher.

In this paper SVM classifiers are used to classify the traffic density into a high or low state. Instead of Adaboost [1], the complexity has been reduced by applying the product rule on the results of individual SVMs. Only critical texture and edge features such as energy, uniformity etc. are selected. Therefore, the sizes of the feature vectors are reduced, optimizing feature extraction.

This paper uses image processing algorithms for feature extraction that can be used to analyze CCTV camera feeds from traffic cameras which will further help the SVM to detect and classify traffic congestion levels into low density or high density in real time. If in a continuous flow of images, a majority of the images is classified as high density, a traffic congestion warning is sent to the respective authority.

2 Algorithm Description

System flow of the proposed algorithm is shown in Fig.1, which comprises of feature extraction, SVM training, and real-time traffic congestion detection as three main components. An ROI area is manually defined to indicate the coverage region of a VLD across the images [1]. The traffic density classification is in terms of low and high density of traffic.

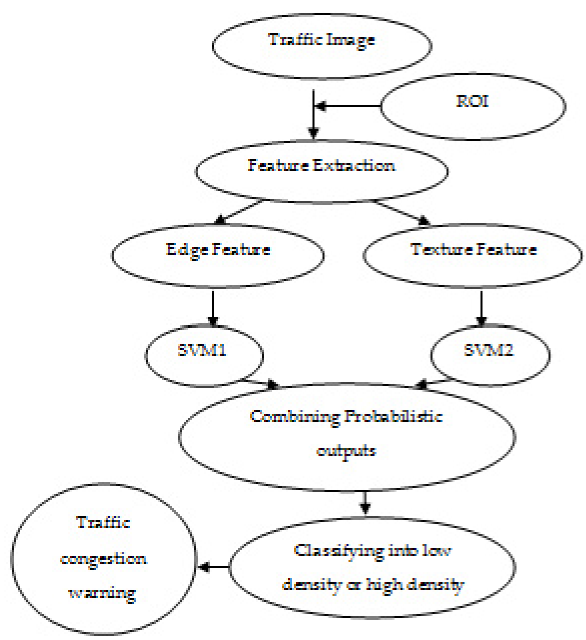

Fig. 1 System flow of the proposed algorithm

ISSN 2229-5518

3 Feature Extraction

Used in both pattern recognition and image processing, feature extraction is a special form of dimensionality reduction. Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

Features are efficient in distinguishing between the traffic density states in this system. Enhanced texture and edge features are used as they are efficient in distinguishing the traffic density states and are not too sensitive to illumination changes [13]. The gray scale image is the input for the proposed algorithm. Further ROI selection is the preprocessing applied to the input image.

3.1 Edge Feature Extraction

Edges are significant local changes of intensity in an image. Edges typically occur on the boundary between two different regions in an image. Edge features are supposed to reflect the fact that more vehicles on the road would result in more edges in the image. In this paper, an Edge Orientation Histogram algorithm is applied to the ROI area [14].

To extract edge features, the basic idea is to build a histogram with the directions of the gradients of the edges (borders or contours). The gradient is the change in gray level with direction. This can be calculated by taking the difference in value of neighboring pixels. The detection of the angles is possible through Sobel Operators. The next five operators could give an idea of the strength of the gradient in five particular directions (Fig.2).

$$\underset{(a)}{\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{matrix}} \quad \underset{(b)}{\begin{matrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{matrix}} \quad \underset{(c)}{\begin{matrix} 2 & 2 & -1 \\ 2 & -1 & -1 \\ -1 & -1 & -1 \end{matrix}} \quad \underset{(d)}{\begin{matrix} -1 & 2 & 2 \\ -1 & -1 & 2 \\ -1 & -1 & -1 \end{matrix}} \quad \underset{(e)}{\begin{matrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{matrix}}$$

Fig. 2 The Sobel masks for 5 orientations: (a): vertical, (b): horizontal, (c) and (d): diagonals, (e): non-directional

The input to the Edge Orientation Histogram algorithm is the source image which consists of the path to the ROI. The output is a vector with 5 dimensions representing the strength of the gradient in the five directions – horizontal, vertical, diagonals, non-directional. We first calculate the size of the image in terms of width and height and build a matrix with the same size and 5 dimensions to save the gradients for each direction. The filter for each direction is applied to the source image and the result is stored in the matrix created. The maximum gradient is obtained from the matrix and the type of the gradient is used to build the histogram. To eliminate redundant gradients, canny edge detection is applied to the source image. The type of maximum gradient is multiplied with the resulting image after applying edge detection. To build the histogram, the image is divided into clips and the histogram is built for each clip with 5 bins [14]. Then, the values of the histogram for each clip are averaged over each direction to obtain a final vector of 5 dimensions.

3.2 Texture Feature Extraction

Considering vehicles as the 'grains' in the texture, more cars residing in the Region of Interest brings a coarser texture while fewer or no vehicles would have a closer texture distribution to the road surface itself [1].

In this paper, the method adopted by Zhidong Li et al is applied to calculate the texture statistics [1].

Firstly, the ROI input image is scaled into 16 gray levels.

Secondly, the first order statistics are calculated. First order measures are statistics calculated from original image values and do not consider the pixel neighborhood relationships.

Let L be the number of distinct gray levels and z be the random variable denoting the gray-levels, then p (z_i), (where i=0, 1,...L-1), is the probability of a gray level occurring in an image. To calculate the probability of occurrence of intensities, first a histogram of intensity is built with 16 levels each representing a gray level. Then, the length for each gray level is divided with total number of pixels to obtain the probability of its occurrence.

We then calculate the 5 dimensions used for 1st order statistics. The dimensions are: Mean, Overall Standard deviation (SD), R-Inverse variance (IV), Overall uniformity, Overall Entropy.

Thirdly, the second order statistics are calculated-

To calculate the second order statistics, the positional operators used are: 0°, 90°, 45°, 135° with a distance of 1 unit. The symmetric GLCM and its corresponding statistical descriptors are calculated for each image with each positional operator. It's descriptors are then averaged over the four angles to give a rotationally-invariant texture feature description.

Let c_{ij} be a normalized element of a GLCM where i is the row position and j is the column position. The texture descriptors used are: Angular Second moment (ASM), Dissimilarity, Energy, Contrast, and Correlation.

3.3 Construction of Feature Vectors

Each image is represented by two separate feature vectors consisting of texture and edge features respectively.

V = [t1 t2 t3 t4 t5 t6 t7]

E = [e1 e2 e3 e4 e5]

In the texture vector V, ti (where i=1, 2, 3…7) represents the texture feature dimension. Only selected features are considered for the vector as explained in Chapter 5. t1 represents Inverse variance, t2- Uniformity, t3- Entropy, t4- ASM, t5- Dissimilarity, t6- Energy, and t7- Correlation. In the edge vector E, ej (where j=1, 2...5) represents the edge feature dimension. e1 to e5 represents the number of pixels with increasing intensity in the direction of maximum gradient, where e5 represents the number of pixels of maximum intensity in the direction of maximum gradient.

4 Combined Classifier

SVM is mainly used for classification. In the training phase, the input fed to the SVM is the feature vector extracted from the images. In this work, two SVM classifiers are trained with the edge feature vectors and texture feature vectors extracted from the sample training images respectively. After fitting the model, the feature vectors extracted from sample testing im-

ISSN 2229-5518

ages are fed into the respective SVM. The output is in terms of probability of the test image belonging to each class, low density and high density. The proposed algorithm aims at combining the results of the SVM on edge feature vector and SVM on the texture feature vector to improve the efficiency, using product rule. Product rule is an effective method to combine classifiers trained with different image descriptors. When the classifiers have small errors and operate in independent feature spaces, it is very difficult to combine their (probabilistic) outputs by multiplying them. Thus, this product rule is used to determine the final decision. First the posterior probability outputs $P_j k (x^k)$ for class j of n different classifiers is combined with the product rule as shown in (1).

$$P_j(x^1 \ldots x^n) = \underset{k=1}{\overset{n}{\pi}} P_j(x^k) \qquad (1)$$

Where x^k is the pattern representation of the k^{th} descriptor. Then the class with the largest probability product is considered as the final class label belonging to the input pattern [16].

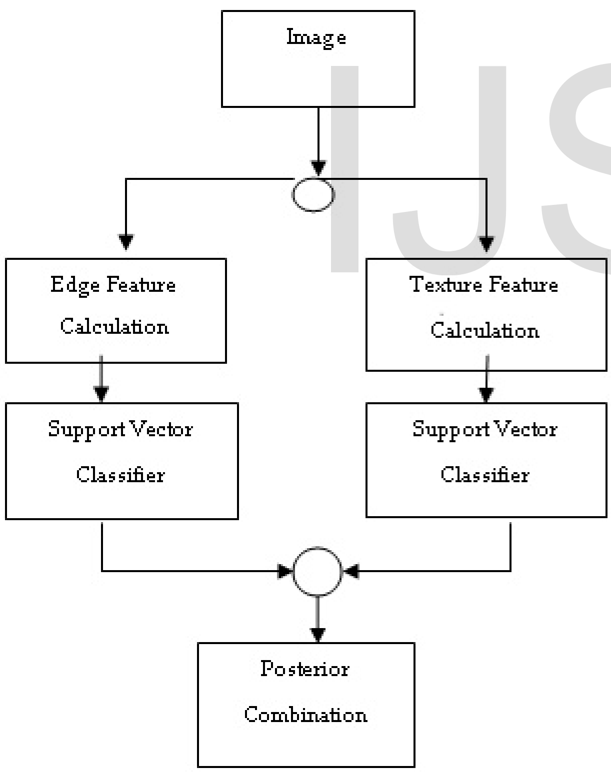

Fig. 3 Flow diagram of combined classifier

5 Experimental Results and Discussion

This work uses traffic images taken from the cameras installed at various junctions by the Bangalore Traffic Police. This contains both day time and night time images. Different approaches evaluated in this paper include an SVM with the texture feature, an SVM with the edge feature and combining both texture and edge features using product rule. The training data covers day time and night time images and similarly the testing is evaluated on the images covering both day and night images. Fig. 4, Fig. 5 shows some sample preprocessed training images used. Fig. 6 shows a few preprocessed testing samples used.

H1 H2 H3 H4

Fig. 4 Sample preprocessed training images-High density at day

Fig. 5 Sample preprocessed training images- Low density at day: Low1-5, Low density at night: NLow1-4

Test1 Test2 Test3 Test4

ISSN 2229-5518

Test5 Test6 Test7 Test8

Test9 Test10 Test11 Test12

Fig. 6 Sample preprocessed testing images

The edge results obtained after applying edge histogram descriptor is given in Table 1, which gives the strength of the gradient. It shows the number of pixels with varying intensity in the direction of maximum gradient. Here, E1 is the number of pixels with minimum intensity; E5 is the number of pixels with maximum intensity in the direction of maximum gradient. It is observed that the value for number of pixels of maximum intensity in the direction of maximum gradient have increased for high density, i.e., E5 have increased for high density in day time because there are more number of edges in that direction.

The texture results shown in Table 2 reflect that for images with low density of traffic, the pixel distribution will be closer to the road surface. As it is seen in Table 2, the values for ASM, Energy(Eng.), Uniformity (Unf.), IV increase for images with low density of traffic.
Since for images with high density of traffic, there is more variation around the mean, and more difference in gray levels, the values for dissimilarity(Diss.), entropy(Ent.) increase for those images. As it is seen in Table 3, there is no significant trend in the variations of values for mean and standard deviation(SD). Hence, it has been decided to remove those features from the texture vector as they may create problems for the SVM to classify accurately. Also, after extracting texture vectors for more images, it was found that the values for contrast do not show a significant trend. Hence, this feature has not been considered in the final texture vector used for training.

TABLE 1 RESULTING FEATURE VALUES AFTER APPLYING EDGE ORIENTATION HISTOGRAM

Ima--ges	Number of pixels of Min. Intensity in direction of maximum gradient → Max. Intensity in direction of maximum gradient				
	E1	E2	E3	E4	E5
L1	4.801e+02	2.603e-17	2.778e+01	1.506e-18	6.25e-02
L2	4.801e+02	2.603e-17	2.778e+01	1.506e-18	6.25e-02
L3	4.816e+02	2.611e-17	2.630e+01	1.425e-18	6.422e-02
L4	4.802e+02	2.603e-17	2.771e+01	1.502e-18	6.251e-02
NL1	2.769e01	9.155e-04	4.803e+02	6.256e-04	2.736e-02
NL2	2.769e+01	6.408e-04	4.803e+02	2.899e-04	2.899e-04
NL3	2.765e+01	8.544e-04	4.803e+02	3.967e-04	3.199e-02
NL4	2.771e+01	9.353e-03	4.803e+02	5.416e-03	4.470e-03
H1	5.074e02	2.750e-17	3.243e-01	1.758e-20	2.607e-01
H2	5.070e+02	2.748e-17	1.275e-01	6.914e-21	8.871e-01
H3	5.032e+02	2.764e-10	3.013e+00	1.633e-19	1.813e+00
H4	5.075e+02	2.751e-17	1.286e-01	6.976e-21	3.286e-01

TABLE 2 RESULTING TEXTURE FEATURE VALUES

Images	ASM	Diss.	Corr.	Eng.	Ent.	IV	Unf.
L1	0.26	0.34	0.76	0.52	2.21	0.25	0.34
L2	0.30	0.37	0.72	0.55	2.08	0.27	0.38
L3	0.11	0.44	0.79	0.32	2.77	0.15	0.18
L4	0.18	0.39	0.79	0.42	2.49	0.20	0.26
NL1	0.14	0.35	0.75	0.38	2.40	0.19	0.23
NL2	0.14	0.33	0.79	0.37	2.52	0.17	0.21
NL3	0.12	0.43	0.77	0.35	2.61	0.17	0.20
NL4	0.14	0.35	0.77	0.37	2.46	0.18	0.22
H1	0.05	0.76	0.84	0.23	3.30	0.11	0.12
H2	0.06	0.70	0.82	0.25	3.09	0.13	0.15
H3	0.06	0.80	0.80	0.24	3.15	0.12	0.14
H4	0.06	0.65	0.85	0.24	3.19	0.12	0.13

TABLE 3 RESULTING DISCARDED TEXTURE FEATURE VALUES

Images	Contrast	Mean	SD

ISSN 2229-5518

L1	1.01	109	41.71
L2	1.18	117	53.26
L3	1.20	77	35.07
L4	1.12	82	36.19
NL1	0.96	61	31.2
NL2	0.89	62	31.4
NL3	1.29	67	32.7
NL4	0.92	64	31.9
H1	2.21	259	64.19
H2	1.89	88	37.48
H3	2.37	98	39.55
H4	1.72	112	42.28

We have trained the SVMs with around 430 images which include day and night images. Table 4 shows the results of texture SVM and edge SVM for 12 sample test images. The results are classified into probability of low density, probability of high density for texture SVM and edge SVM .The cross marks "×" indicates wrongly predicted results. Texture SVM has 4 wrongly predicted results, edge has 3 wrongly predicted results .From the results in Table 4, it is observed that edge SVM is more efficient in distinguishing the traffic density states as it gives less incorrect predictions in comparison with texture SVM.

Table 5 shows the results of the combined classifier that is combination of texture SVM and edge SVM. Here cross mark "×" indicates the wrongly predicted results. It is observed that after combining the individual SVM outputs the number of incorrect predictions has decreased. Texture SVM has 4 incorrect predictions, edge SVM has 3 incorrect predictions but combined Classifier has only 2 incorrect predictions. Thus the combined classifier has improved the classification accuracy.

TABLE 4 RESULTS OF TEXTURE SVM AND EDGE SVM

Images	Texture SVM		Edge SVM	
	Probability of low density	Probability of high density	Probability of low density	Probability of high density
Test1	0.12	0.88	0.9 ×	0.1
Test2	0.41 ×	0.59	0.9	0.1
Test3	0.44	0.56	0.88 ×	0.12
Test4	0.80	0.20	0.9	0.1
Test5	0.66	0.34	0.9	0.1
Test6	0.21 ×	0.79	0.9	0.1
Test7	0.37 ×	0.63	0.9	0.1
Test8	0.10	0.90	0.4	0.6
Test9	0.72	0.28	0.9	0.1
Test10	0.58	0.42	0.9	0.1
Test11	0.49 ×	0.51	0.89	0.11
Test12	0.24	0.76	0.89 ×	0.11

TABLE 5 RESULTS OF COMBINED CLASSIFIER

Images	Texture SVM		Edge SVM		Combined SVM
	Probability of low density	Probability of high density	Probability of low density	Probability of high density	Class Label
Test1	0.12	0.88	0.9 ×	0.1	High density
Test2	0.41 ×	0.59	0.9	0.1	High density ×
Test3	0.44	0.56	0.88 ×	0.12	High density
Test4	0.80	0.20	0.9	0.1	Low density
Test5	0.66	0.34	0.9	0.1	Low density
Test6	0.21 ×	0.79	0.9	0.1	Low density
Test7	0.37 ×	0.63	0.9	0.1	Low density
Test8	0.10	0.90	0.4	0.6	High density
Test9	0.72	0.28	0.9	0.1	Low density
Test10	0.58	0.42	0.9	0.1	Low density
Test11	0.49 ×	0.51	0.89	0.11	Low density
Test12	0.24	0.76	0.89 ×	0.11	Low density ×

6 CONCLUSION

This work proposes a method for traffic density estimation using image processing and machine learning.Images are taken from the live camera feed.The preprocessing done on these images converts them into gray scale versions and identifies the ROI.The ROI is used for further processing.

From the values obtained in Table 1,it is obsereved that the values for the number of pixels of maximum intensity in the direction of maximum gradient have increased for high density,where the gradient refers to the change in gray level with direction.

From Table 2 it is observed that features like Dissimilarity, Correlation, Entropy have increased for high density, whereas features for ASM, Energy, IV, Uniformity values have increased for low density.The reasons for which were discussed earlier.After obtaining the featuresfor edge and texture, feature vectors were constructed.

As seen in Table 4 It is observed that the edge SVM shows slightly more efficiency than texture SVM.

From the results in Table 5 it is observed that number of incorrect predictions have decreased compare to individual SVM and an overall efficiency of 95% is obtained.

In this work, 5 cosecutive images with a time gap of 1 minute are passed. These images are classified as low or high density and based on value a congestion alert is provided to respective authorities.As 5 images are passed, the threshold value is set as 3.If 3 or more images shows high density, then the final result shows traffic congestion.The results match our expectation.This work can be extended to further junction of Bangalore city and future enchacement can be provided to remove shadows.

ACKNOWLEDGMENT

This work is supported by the B.N.M Institute of Technology,Bangalore and Mapunity Information Services(p) Ltd. Mapunity works closely with the Bangalore Traffic police and was responsible for helping establish India's first comprehensive traffic management system. It was this system that provided the required data for our project.

REFERENCES

[1] L.Zhidong et al., "On Traffic Density Estimation with a Boosted SVM Classifier," *IEEE Trans. Digital Image Computing: Technology and Applocations,* Dec 2008, doi: 10.1109/dicta.2008.30.

[2] S.Shiliang, Z.Changshui,Y. Guoqiang,"A Bayesian Network Approach to Traffic Flow Forecasting," *IEEE Trans. Intelligent Transportation Systems,* vol.7, no.1, March 2006, doi: 10.1109/ITTS.2006.869623.

[3] G.Surendra et al. (2002)," Detection and Classification of Vehicals," *IEEE Trans. Intelligent Transportation Systems,* vol.3, no.1, March 2002, doi: 10.1109/6979.994794.

[4] C.Rita, P.Massimo, M.Paola,"Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. ITS,* vol.1, no.2, June 2000, doi: 10.1109/6979.880963.

[5] W.Jingbin et al.,"Detection Objects of Variables Shape Structure with Hidden State Shape Models," *IEEE Trans. Pattern analysis and Machine Intelligence,* vol.30, no.3, March 2008, doi: 10.1109/TPAMI.2007.1178.

[6] F.Vittorio Ferrari et al., "Groups of Adjacent Contour Segments for Object Detection," *IEEE Trans. Pattern analysis and Machine Intelligence,* vol.30, no.1, Jan 2008, doi: 10.1109/TPAMI.2007.1144.

[7] S.W.Chee, K.P.Dong, J.P. Soo.," Efficient use of Edge Histogram Descriptor," *ETRI Journal*, 24(1), Feb 2002.

[8] T.Evan, C.Jing," Vehicular Traffic Density Estimation via Statistical Method with Automated State Learning," *Proc. IEEE Symp. Adavced Video and Signal based Surveillance (AVSS2007),* pp.164-167, 5-7, Sept 2007, doi: 10.1109/AVSS.2007.4425304.

[9] L.Xiaokun and M.P.Fatih," A Hidden Markov Model Framework for Traffic Event Detection using Video Features," *ICIP,* March 2004.

[10] W.Lei, X.Ping, L.C.Kap," Incorporating Prior Knowledge into SVM for Image Retrieval," *proceedings of the 17th International Conference on Pattern Recognition 2004 (ICPR 2004),* August 2004.

[11] Z.Shile, F.Jianping, L. Hong,"Salient Object Detection on Large-Scale Video Data," Proc. *IEEE Symp. Computer Vision and Pattern Recognition,* pp. 1-6, June 2007, doi: 10.1109/CVPR.2007.383495.

[12] A Berengolts, M. Lindenbaum, "On the Distribution of Salience," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.28, issue 12, pp 1973-1990, Dec 2006, doi: 10.1109/TPAMI.2006.249.007.

[13] C. G. Rafael and H. W Richard, 2002, *Digital Image Processing*, Prentice Hall, pp.34-134.

[14] U. Roberto. (2012) "Edge Orientation Histograms in Global and Local Features,"Jan 26, 2012. Available at: http://roberto.blogs.cultureplex.ca/2012/01/26/edge-orientation-histograms-in-global-and-local-features/ Feb 13, 2013.

[15] S.Scholkopf et al., 1999, *Advances in Kernel methods: Support Vector Learning,* MIT Press, Cambridge.

[16] A.Azizi, C. V .Remco, A. W. Marco, "An Ensemble of Deep Support Vector Machines for Image Categorization, Soft Computingand Pattern Recognition," *SOCPAR'09*, *International Conference*, DEC 2009.

DETAILS ABOUT AUTHORS

Sukriti Sharma completed her B.E in information science from BNM Institute of Technology, Bangalore, affiliated to VTU, India in July 2013. Her final year group project titled " Traffic Density Estimation using SVM Classifier" won best project award in computing division at JED-I national project challenge held at IISC, Bangalore June 2013. She is now working as a R&D Engineer in Nokia Solutions and Networks. E-mail:sukriti14.91@gmail.com

***Vidya Meenakshi K** completed her B.E in information science from BNM Institute of Technology, Bangalore, affiliated to VTU, India in July 2013. Her final year group project titled " Traffic Density Estimation using SVM Classifier" won best project award in computing division at JED-I national project challenge held at IISC, Bangalore June 2013.*
Email: vidya.meena09@gmail.com

***Apurva V.S** completed her B.E in information science from BNM Institute of Technology, Bangalore, affiliated to VTU, India in July 2013. Her final year group project titled " Traffic Density Estimation using SVM Classifier" won best project award in computing division at JED-I national project challenge held at IISC, Bangalore June 2013.E-mail:apurvavs91@gmail.com*

***Shreyas Satish** has been a software engineer for the past three years at Mapunity - a social technology lab in Bangalore - where he was responsible for building the entire technology stack for Technagara* (http://technagara.in) *(untilJuly,2013). He received his 'Bachelor of Engineering' in Information Science from the Visvesvaraya Technological University (Belgaum).E-mail:Shreyas.satish@gmail.com.*

***Saritha Chakrasali** is an associate professor at BNM Institute of Technology in the ISE department. Her research interests are image processing, information technology, cryptography and network security, intelligent agents. She has published in 5 international conferences and 2 national conferences. She is an organizing committee member for the fifth International conference on signal and image processing (ICSIP 2014).*
E-mail:Saritha.chakrasali@gmail.com