

# Detection & Classification of Vehicles in Varying Complexity of Urban Traffic Scenes

<sup>1</sup>Muhammad Umair Arif, <sup>1</sup>Zain ul Aabidin Lodhi, <sup>2</sup>Maheen Khan, <sup>1</sup>Rana Hammad Raza

<sup>1</sup>National University of Sciences and Technology, Karachi, Pakistan

<sup>2</sup>Bahria University, Karachi, Pakistan

## Abstract

*Detection and classification of vehicles is a paramount task in surveillance framework and for traffic management and control. The type of transportation infrastructure, road conditions, traffic trends and illumination conditions are some of the key factors that affect these essential tasks. This paper explores performance of existing techniques regarding detection and classification in local, day time, complex urban traffic videos with increased free flowing vehicle volume. Three different traffic datasets with varying level of complexity are used for analysis. The scene complexity is governed by factors such as vehicle speed, type and size of dynamic objects, direction of motion of vehicles, number of lanes, occlusion, length and camera viewing angle. The datasets include a big classification volume ranging to 1516 vehicles in NIPA (customized local dataset) and 1009 vehicles in TOLL PLAZA (customized local dataset) along-with a publicly available dataset with 51 vehicles namely, HIGHWAY II. Existing detection algorithms such as blob analysis, Kalman filter tracking and detection lines were applied for detection on all the three datasets and experimental results are presented. Results show that the algorithms perform well for low density, low speed, less shadow, better image resolution, appropriate camera viewing angle, better lighting conditions and occlusion free zones. However, as soon as the complexity of the scene is increased, several detection errors are identified. Further obtaining robust and invariant features of local vehicles design has been challenging during the process. A custom GUI is built to analyze results of the algorithm. This detection is further extended to classification of 231 vehicles of NIPA dataset which is a highly complex urban traffic scenario. Vehicles are classified as Small Vehicle (SV), Large Vehicle (LV) and Motorcycle (M) by using area threshold based classifier and dense Scale Invariant Feature Transform (SIFT) and Artificial Neural Network (ANN) classifier. Detailed comparison of both classifier results show that SIFT and ANN classifier performs better for classification tasks in highly complex urban scenarios and also points out that practical systems still require a robust classification scheme to get more than 80% accuracy.*

**Keywords:** Classification of vehicles, urban traffic, detection of vehicles, Neural Networks, dense SIFT

## Introduction

Surveillance industry has been one of the highly influenced businesses since the advancement in vision systems. Vision based surveillance systems have capacity to provide quality and financially feasible solutions with easy installation, maintenance and operation. Due to these attributes their usage is quite significant in Intelligent Transportation Systems (ITS) for components such as traffic data collection, traffic management and monitoring, incident detection and many other applications.

In the context of ITS, a lot of vision based research work has been reported; however there are still great challenges open for the research community. Different texture and model of vehicles, image size, lighting condition, camera viewing angle and occlusion are some of the complicating factors. Further, the identification within class differences makes the problem even harder.

The first comprehensive survey in this field is given at [3]. It discusses advantages/disadvantages of several computer vision techniques in the fields of traffic monitoring and automatic vehicle guidance and categorize them in input data i.e. feature-driven, area-driven, or model-based and the processing domain i.e. spatial/frame or temporal/video. Similarly, a more recent, valuable and very detailed review of computer vision techniques for urban traffic analysis is done by Buch *et al* [12]. The literature provides challenges of the urban domain of vehicle classification as compared to the highway domain along with several detection and classification techniques. Some of the commonly used vehicle detection techniques are inter frame differencing, background subtraction, optical flow estimation, Gaussian Mixture Model (GMM), deformable 3-D geometric model, graph cuts, object based segmentation and sensor fusion etc. Similarly, for classification tasks, feature based techniques are used that include SIFT, SURF and dense SIFT etc. Basic threshold, k Nearest Neighborhood (kNN), Support Vector Machine (SVM) and Neural Networks are then used to classify these features into known classes. Direct comparison of these proposed algorithms becomes challenging due different datasets being used with no common benchmark.

Zang *et al* [5] proposed a real time detection and classification system using un-calibrated video cameras. Issues like longitudinal occlusions, light reflections and camera vibrations affect the performance of the proposed system. Similarly, Leibe *et al* [8] has developed an optimized detector and tracker. It provides multi-view/ multi-category object detection and recognition using calibrated cameras and scene geometry. It suggests addition of stereo depth and adaptive background modeling for better performance on static cameras. Buch *et al* [10] present a review of the commercial video analytics systems. The authors have employed motion silhouettes and 3D Histogram of Oriented Gradients (HOG) for vehicle detection and classification. As per their findings, 3D HOG classifier performs better in challenging urban environments. The proposed work needs to be tested for diverse weather and operation conditions. Feris *et al* [13] apply motionlet classifiers i.e. classifiers that are learned with vehicle samples clustered in the motion configuration space. Vehicles are classified into Buses, trucks, SUV and cars. The authors claim that their system has the ability to handle high volume of activity, challenging urban conditions and occlusions because of semantic attribute based approach. Mithun *et al* [14] use multiple spatio-temporal images to identify the latent occlusions among vehicles for robust detection. They have used shape-based features for classification. The texture based features are used to sub-classify

the sub categories. Recently, Tang *et al* [17] reports vehicle detection and recognition on static images using Haar-like features and AdaBoost and suggest them to be useful for classification tasks.

As highlighted earlier, practical systems for vehicle detection and classification have to cater for various factors. Variation in size, shape, color, orientation and speed of vehicles are some multifaceted issues. Lighting conditions, complex cast shadows, man-made structures and occlusion due to camera viewing angle are some other aspects that need to be considered as well. This paper applies existing techniques to such practical scenarios and attempts to analyze them. Algorithms like background subtraction, blob analysis, Kalman filter tracking and detection lines are utilized. Here classification is performed using basic area based categorization and Neural Networks. Additionally, feature extraction and description is performed using dense SIFT.

To the best of our knowledge, existing algorithms have seldom been reported for complex urban traffic scenarios with dynamic and complex traffic trends. Similarly, the reported results are generally on controlled traffic conditions which barely reflect the robustness of such algorithms. In this paper, two complex custom datasets in local, daytime environment are acquired and their complexity levels are defined to analyze the performance of existing vision algorithms. We believe that this work will pave the way for developing more robust algorithms for practical systems. Section 2 covers evaluated datasets and system settings. Section 3 explains the methodology. Section 4 provides results and further analysis. Finally, Section 5 concludes the paper.

## Evaluated Datasets and System Settings

### Datasets Used

In order to analyze the algorithms, one online available test dataset and two local, day time, free flowing traffic datasets with varying level of complexity are used. Details of these datasets are appended below and also tabulated in Table 1, 2, 3.

- HIGHWAY II [4]
- NIPA (Customized local dataset)
- TOLL PLAZA (Customized local dataset)

First traffic scene (HIGHWAY II) used in the analysis contains single lane, high speed vehicles, no vehicle occlusion, cast shadows with medium size and strength and the camera viewing angle is top front view. The dataset provides feed of surveillance cameras generally installed on highways abroad. Second traffic scene (TOLL PLAZA) acquired from local highways contains two lanes, slow speed vehicles, varying vehicle types, cast shadows with large size and high strength, occlusion and the camera viewing angle is top front view. Third dataset (NIPA) also acquired from another local highway contains three lanes, high speed vehicles, varying vehicle types, cast shadows with small size and less strength, more occlusion and the camera viewing angle is top side view. Both TOLL PLAZA and NIPA are actual urban traffic scene.

All the datasets were thoroughly checked frame by frame for ground truth analysis. For.eg. NIPA dataset contains 1516 vehicles and TOLL PLAZA dataset contains 1009 vehicles. General classification is done on the basis of Motorcycle (M), Small Vehicle (SV) and Large Vehicle (LV). Primarily, these datasets

were used for analysis of results from several algorithms discussed in the methodology section.

### Custom GUI

A custom graphical user interface (GUI) was built to test the results of the algorithms that were implemented. A snapshot of the GUI is presented in Figure 1. Using this user-friendly GUI, results of the algorithms are compared with ground truth. An added feature to display lanes as well as registration lines/detected regions is incorporated so as to enable the user to identify the regions of interest.

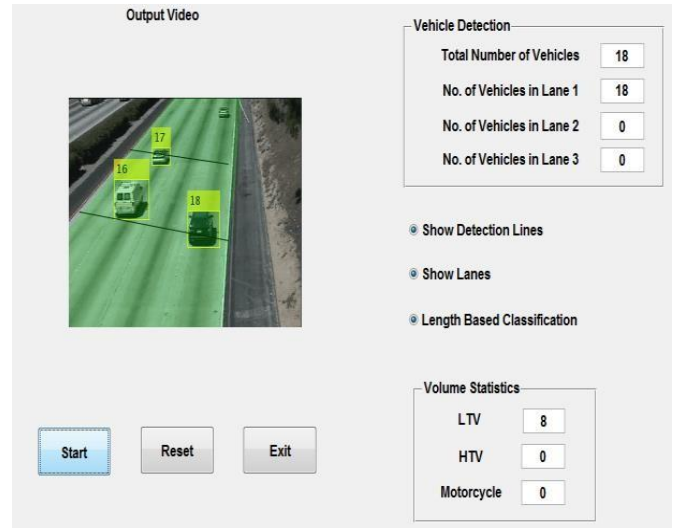



Figure 1: Custom GUI designed in MATLAB for HIGHWAY II dataset

### Testing Platform


The algorithms were executed on a Core i-7, Quadcore, 8 GB RAM processing device. VLfeat code [11] for dense SIFT, Matlab® 2014 Computer Vision and Neural Networks Toolbox were used for simulations.

Table 1: Highlights of the NIPA dataset


Video Frame	
Sequence Type	Outdoor
Video Length	00:08:16
Image Size	640 x 420
Shadow Strength	Low
Shadow Size	Small
Object Class	Vehicle
Object Size	Small

Object Speed (Pixels)	Fast
Noise Level	Medium
Camera position	Side view

**Table 2: Highlights of the HIGHWAY II dataset**

Video Frame	
Sequence Type	Outdoor
Video Length	00:00:33
Image Size	320 x 240
Shadow Strength	Medium
Shadow Size	Medium
Object Class	Vehicle
Object Size	Small
Object Speed (Pixels)	Medium
Noise Level	Low
Camera position	Top view

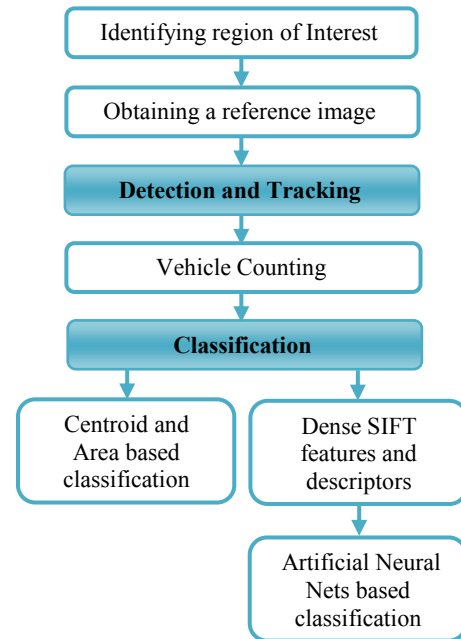
**Table 3: Highlights of the TOLL PLAZA dataset**

Video frame	
Sequence Type	Outdoor
Video Length	00:55:31
Image Size	720 x 576
Shadow Strength	High
Shadow Size	Large
Object Class	Vehicle
Object Size	Medium
Object Speed (Pixels)	Slow
Noise Level	Medium
Camera position	Front view

## Methodology

The algorithms which are to be implemented for the above complex datasets were shortlisted and applied to each dataset for acquiring results. Sub-sections below provide details regarding

detection of vehicles, classification on the basis of area based algorithm and classification using dense SIFT and Neural Networks. Figure 2 illustrates the methodology flowchart.



**Figure 2: Block diagram of the methodology employed**

### Identifying Region of Interest

As a preprocessing step, regions of interest i.e. lanes, are separated from the rest of the video with manual settings. This step is required to remove clutter from the video dataset.

### Foreground Detection

In order to detect a vehicle in a video, there is a requirement to estimate a primary frame with which other frames can be compared. This frame is referred to as the background or reference image. Generally for robust vehicle detection, the background image must have no moving objects and shall have constant illumination and ambient conditions across frames. Generally, it becomes complicated to acquire a background image in traffic scenes. Therefore, the background is estimated from the video stream. This estimation is an iterative process and requires pixel level processing. Every pixel in an image contains certain red, green and blue (RGB) colors ranging from 0 to 255. These colors combine to give the pixel its unique color and brightness. Every corresponding pixel in the new acquired frame is then compared with its previous frame counterpart at the same location. If the difference in the RGB values of the two compared pixels lies within the threshold value then the two pixels are considered similar. Modeling the pixels as Gaussians Mixture Models (GMM) is an effective approach to separate the background from foreground and is being used for real-time tracking [2]. In GMM the process of pixel comparison takes place throughout the frame and continues till all the pixels are declared part of the reference image. Once this criterion is met the frame is saved as a background image and acts as a benchmark for vehicle detection.

### Vehicle Detection and Tracking

Detection and tracking of vehicles is performed by estimating the motion of each detected vehicle using blob analysis and



Kalman filter respectively. An example for motion based object detection and tracking is available in MATLAB computer vision toolbox [16]. An insight was taken from it to understand the dynamics and build our own algorithm that can be tailored towards complex urban traffic datasets. In this algorithm, connected groups of foreground pixels called „blobs“ are considered which correspond with moving objects [6]. An array of tracks representing moving vehicles is generated so as to maintain the state of a tracked vehicle. Noisy detections in a frame tend to result in short-lived tracks. To cater for this the algorithm initiates vehicle tracking only after it is followed for some number of frames. Similarly, when no detections are associated with a particular track for several consecutive frames, the algorithm assumes that the vehicle has left the field of view thus deleting the track. A track may also get deleted as noise if it was tracked for a short time and marked invisible for most of the frames. Next foreground detector is used to obtain binary motion segmentation. Subsequently, morphological operations are performed on the resulting binary mask to remove noisy pixels and to fill the holes in the remaining blobs. Kalman filter is then used to predict the centroid of each track in the current frame and updates its bounding box accordingly [9]. Assigning object detections in the current frame to existing tracks is done by minimizing cost function. The cost function is defined as the negative log-likelihood of a detection corresponding to a track. The cost minimization involves following two steps:

*Step 1:* The cost of assigning all detections to each existing track is computed. This cost utilizes Euclidean distance between the centroid of the detection and the predicted centroid of the track. Furthermore, it contains Kalman filter confidence prediction.

*Step 2:* Assignment problem represented by the cost matrix and the cost of not assigning any detections to a track is solved. It uses the Munkres version of the Hungarian algorithm to compute an assignment which minimizes the total cost.

In the last part, the algorithm identifies assigned tracks and delete lost tracks. It also deletes recently created tracks that have been invisible for too many frames.

### Counting of Vehicles

The foremost task of vehicle counting entails that no vehicle gets counted more than once. For this purpose, lines are introduced in the program for each lane [8]. These lines are called the registration lines and further divided into two categories called „entrance registration lines“ and „exit registration lines“. The region between these two registration lines is known as „detection region“ and is used for vehicle counting and segmentation. This segmentation is further used for classification tasks. Results of implemented detector are shown in Figure 3.

The algorithm for vehicle counting comes into play during the vehicle detection phase. At least one registration line must be active for this algorithm to work. The RGB value of a pixel along the registration line in every frame is compared to the RGB value of the same pixel in the reference frame. This process is repeated for all the pixels along the registration line. If the comparison of the pixels yields a value below the threshold (e.g. threshold = 15), both pixels are declared similar. This in other words means that no vehicle has passed the registration line. Pixels are declared different if the difference value is beyond the threshold. The threshold value increases detection sensitivity because a little variation in threshold may detect pixels as part of a vehicle even if they are not. Similarly, the threshold value must be sufficiently

small so as to identify vehicles whose color is similar to the background color.



**Figure 3:** Implemented detector results. Registration lines are shown in blue color for each lane.

There is also likelihood that a vehicle might get counted more than once. To overcome this issue, the algorithm only counts the vehicle if it is not present in the previous frame. This is done by looking at the assigned ID number (details on assigning ID are provided in next section). As soon as a vehicle crosses the registration line, it gets counted. In the very next frame, the vehicle will still be crossing the registration line but it will not get counted because it was already there in the previous frame. This also proves fruitful in high traffic density situations because the distance a vehicle travels between two frames is not large and the vehicle still remains on the registration line.

### Area based Classification

When the blob comes into detection region, centroid and area of each blob are calculated and each blob is labeled and counted as a vehicle. The area is passed through a threshold module which classifies it as Motorcycle (M), Small Vehicle (SV) or Large Vehicle (LV).

The area based classification algorithm runs whenever the program is set to detect vehicles and the detection regions in which areas are supposed to be measured are turned on. In addition, area based classification may only be performed on a lane when counting is also enabled on the same lane. When a vehicle crosses one of the registration lines and enters into the detection region corresponding to each lane, the area based classification algorithm measures the area of the vehicle. This makes the areas of all the vehicles in a lane be measured at the starting point enabling the measured areas to be comparable thus minimizing the effect of camera view angle. Note that this is a relative area and does not represent the actual area of the vehicle. The area based classification merely steps along the longitudinal direction in detection region of each lane thereby counting the number of pixels present in the blob of detected vehicle.

In implementation, the length of vehicle is the length of the longitudinal line that is taken by the vehicle region [8]:

$$L = \sqrt{(e_x - s_x)^2 + (e_y - s_y)^2} \quad (1)$$

Where,  $s_x$  and  $s_y$  are the starting coordinates of the line and  $e_x$  and  $e_y$  are the end coordinates.  $L$  is the length of vehicle. Once the areas of all the vehicles present in detection region (that is between the registration lines) in pixels is obtained, it is stored in an array

and passed to a function for thresholding and tagging. Here, area of the each vehicle in pixels is thresholded to classify vehicles into one of the three different categories. These three categories are Large Vehicles (LV), Small Vehicles (SV) and Motorcycles (M). The value of threshold for large vehicles is 2000, vehicle's area shorter than half of the area of the large vehicle's threshold are classified as short vehicles and vehicle's area shorter than one fourth of this threshold are classified as motorcycles. These threshold values are selected by running the algorithm over and over with different threshold values and finding the optimal value. The output of this algorithm is the individual count of all large vehicles (trucks/buses), small vehicles and motorcycles detected.

### Dense SIFT Features and Descriptor for Training

In order to increase the breath of classification of vehicles, dense SIFT [7] was employed. Scale Invariant Feature Transform (SIFT) was developed by David Lowe which is an image descriptor. It is used for image-based matching and recognition tasks. It is well known to be invariant to translations, rotations and scaling transformations in the image domain. However, it has moderate level of invariance when it comes to illumination variations and perspective transformations. In practice, the SIFT descriptors are proven for object recognition and image matching under real-world conditions [1].



Figure 4: Sample images used to train Neural Networks into three categories. (M, SV, LV)

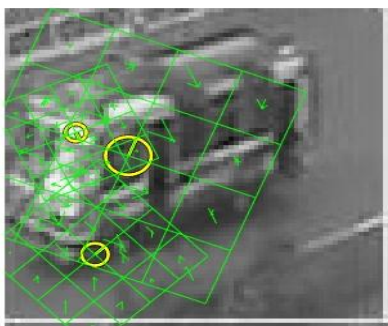


Figure 5: Selection of three features on a test vehicle

Applying SIFT to our vehicle classification task in NIPA dataset generated some issues. As can be seen from the dataset, the resolution of individual vehicle is very low and therefore there are few keypoints that can be processed. In most cases there is only one keypoint which is inadequate for training in Neural Networks. Therefore, we have used dense SIFT for visual object category classification. SIFT descriptors are computed over dense grids in

the image domain as opposed to sparse interest points obtained by an interest operator. Large set of local image descriptors computed over a dense grid are able to provide much more information as compared to their corresponding descriptors which are obtained at sparse image points.

SIFT identifies interest points using Difference of Gaussian Filtering (DoG), however, dense SIFT does not identify interest points. It simply divides the image into overlapping cells before using Histogram of Oriented Gradients (HOG) to describe them. We have used dense SIFT other than SIFT because we are assuming that the scale and orientation of the complex dataset does not change.

After preprocessing and blob analysis the system is trained. When each vehicle enters the detection region during training, feature and descriptor vectors are obtained using dense SIFT. The system is currently trained on 20 samples of bikes (M), cars, qinqhi, rickshaw, suzuki, minivans (SV) and buses and trucks (LV). A vector of 100 feature points for 20 testing images of each category were calculated for training using Neural Networks. Some training images and corresponding feature points are shown in Figure 4 and 5.

### Artificial Neural Networks

Artificial Neural Networks (ANN) can learn and therefore can be trained to recognize patterns, find solutions, forecast future events and classify data. ANN are well documented to be used for traffic related tasks [15]. Neural Networks learning and behavior is dependent on the way its individual computing elements are connected and by the strengths of these connections or weights. These weights can be adjusted automatically by training the network according to a specified learning rule until it performs the desired task correctly. ANN learn by example as do their biological counterparts; a child learns to recognize dogs from examples of dogs. ANN is a supervised learning method i.e. a machine learning algorithm that uses known dataset also known as training dataset. These known parameters help ANN to make predictions. Input data along with their response values are the fundamental components of a training dataset. This supervised learning algorithm along with the training datasets help to build a model that can make predictions of the response values for a new unknown and unlearned dataset. For model validation, a test dataset is employed. In order to have higher predictive power and the ability to generalize for several new datasets, the best way is to use larger training datasets.

The back propagation algorithm is the workhorse of learning in Neural Networks. Back propagation requires a known, desired output for each input value in order to calculate the loss function gradient. Back propagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The lowest possible error theoretically is termed as a global minimum. However, practical problems generally have a solution space which is quite irregular with numerous 'pits' and 'hills'. Such irregularities might be a reason for the network to settle down to a 'local minimum' which is not the best overall solution

The paper utilizes feed forward ANN from Matlab Toolbox. A feed forward ANN uses layers of non-linear "hidden" units between its inputs and outputs. It learns feature detectors by adapting the weights on the incoming connections of these hidden units. These hidden units and their connections enable the algorithm to predict the correct output for every input vector. There are many conditions in which the relationship between the

input vector and the correct predicted output is complicated. For such systems the network must have enough hidden units to model it accurately which consequently would result in many different settings of the weights. These different weight models will then be able to model the training set in the best possible manner, especially for a limited amount of labeled training data. The drawback is that all of these weights would perform low for the test data but work perfectly on the training data. The reason is that each of these weight vectors will make different predictions and the feature detectors have been tuned to work well together on the training data only.

Considering the above understanding of ANN, we choose NIPA dataset and employed feature data from dense SIFT for 8 different categories with 20 training images such that 100 feature points vector was taken for each training image. However, Neural Nets was unable to converge to a possible classification solution. Then, we moved on to three classification vector which was able to converge to a good point. We tested some other static images from the NIPA test video and the results are tabulated in Table 4.

**Table 4: Training and testing results of Neural Nets**

Testing results of classification into eight sub-categories using dense SIFT and Neural Networks					
Category		Tested	Pass	Fail	Percentage
M	Total	20	17	3	85
SV	Cars	15	13	2	86.6
	Qinqhi	15	13	2	86.6
	Rickshaw	15	12	3	80
	Suzuki	15	11	4	73.3
	Minivan	15	8	7	53.3
	Total	75	57	18	76
LV	Buses	10	10	0	100
	Trucks	10	8	1	80
	Total	20	18	1	90

## Results and Analysis

### Detection Results

Steps for the detection of vehicles mentioned in Section 3 were applied on NIPA dataset. A comparison table demonstrating ground truth versus the detection results and their classification is given in Table 5. It can be seen from the table that out of the total 1516 vehicles, the algorithm is able to detect 50%. The results show that for real complex urban traffic videos, the detection algorithms are absolutely inadequate.

Analyzing the individual lanes, we can see that the detection results of lane 1 and lane 3 is 43% and 49% respectively. The main reason for error in these lanes is the camera viewing angle which in turn increases occlusion. There have been cases where motorcycles were hidden behind other motorcycles or small vehicles and were therefore occluded and undetectable.

Second issue is the speed of vehicles. Because of high speed, the vehicles cross the detection and registering lines without being detected through the frame.

Third issue is the inadequate blob formation. The lighting conditions, color of vehicles and the camera resolution were sometimes a hindrance in formation of the blob. Since the blob was not formed properly, the vehicle was not detected.

**Table 5: Detection Results obtained for NIPA dataset**

NIPA				
1516 vehicles				
Ground Truth				
Category	Lane 1	Lane 2	Lane 3	Total
M	369	198	183	750
SV	311	209	218	738
LV	7	15	6	28
Total	687	422	407	1516
Detection				
Category	Lane 1	Lane 2	Lane 3	Total
M	153	90	87	330
SV	137	137	90	364
LV	17	34	23	74
Total	297	261	200	768
Correct vehicle detection	43.2%	61.8%	49.1%	50.6%

**Table 6: Detection results obtained for TOLL PLAZA dataset**

TOLL PLAZA			
1009 vehicles			
Ground Truth			
Category	Lane 1	Lane 2	Total
M	75	91	166
SV	171	265	436
LV	170	237	407
Total	416	593	1009
Detection			
Category	Lane 1	Lane 2	Total
M	60	58	118
SV	112	262	374
LV	145	235	380
Total	317	555	872
Correct Vehicle Detection	76.2%	93.59%	86.4%

For TOLL PLAZA dataset, the overall results are much better with around 86.4% in Table 6. The main reason is the camera angle which is perfect for least amount of occlusion. Secondly, the speed of vehicles is quite slow helping registration line algorithm

work better. Further, due to errors in background model pixels, sunlight and vehicle color the vehicle blob was divided into several small blobs. These small blobs were not counted as vehicles because of small area. The detection range was selected such that there would be minimum effect of the long cast shadows that are present in the video but there were some cases for which small vehicles were classified as large vehicles because of increased vehicle area due shadows.

Similar issues can be identified in the HIGHWAY II dataset in Table 7 with around 62% detection rate. This was a comparatively easier dataset but had detection errors. The reasons of detection errors are also vehicle speed and inadequate blob formation. Further, the road conditions and pixel values of the background on some places divided the blob area of vehicle and thus made it filter out near the register line. This in turn caused zero detection for that vehicle.

The results show that detection using blob analysis, detection line and Kalman filter in complex urban traffic have a wide variety of issues and limitations for practical and complex urban traffic scenarios.

**Table 7:** Detection results obtained for HIGHWAY II dataset

HIGHWAY II	
51 vehicles	
Ground Truth	
Category	Lane
SV	41
LV	1
Total	42
Detection and Classification	
Category	Lane
SV	25
LV	1
Total	26
Correct vehicle detection	61.9%

### Classification Results

For classification results, the NIPA dataset was selected with 231 vehicles on a 2 minute video. The ground truth in this case was developed by observing vehicles detected by the registration line. Therefore considering the detection as 100%, we analyzed the classification algorithms.

A confusion matrix is developed for area and ANN classifiers and is presented in Table 8. The following parameters are calculated for each classifier and are presented in Table 9.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Misclassification Rate = \frac{(FP+FN)}{(TP+TN+FP+FN)} \quad (4)$$

Here,

- FP = False Positive
- TP = True Positive
- TN = True Negative
- FN = False Negative

**Table 8:** Confusion matrix for area threshold and SIFT+ANN classifiers categorizing M, SV and LV in NIPA Data Set.

Area Threshold Classifier			
M	Predicted Yes	Predicted No	Total
Actual Yes	TP=62	FN=10	72
Actual No	FP=46	TN=113	159
Total	108	123	231
SIFT+ANN Classifier			
Actual Yes	TP=25	FN=47	72
Actual No	FP=4	TN=155	159
Total	29	202	231
Area Threshold Classifier			
SV	Predicted Yes	Predicted No	Total
Actual Yes	TP=71	FN=80	151
Actual No	FP=29	TN=51	80
Total	100	131	231
SIFT+ANN Classifier			
Actual Yes	TP=120	FN=31	151
Actual No	FP=66	TN=14	80
Total	186	45	231
Area Threshold Classifier			
LV	Predicted Yes	Predicted No	Total
Actual Yes	TP=7	FN=1	8
Actual No	FP=16	TN=207	223
Total	23	208	231
SIFT+ANN Classifier			
Actual Yes	TP=7	FN=1	8
Actual No	FP=9	TN=214	223
Total	16	215	231

Area based classification performed worse than the Neural Networks. Finding a proper threshold for area is difficult. The dense SIFT features on the other hand found 100 unique features with a descriptor size of 128 each and trained using Neural Networks. Even though the result of SIFT+ANN are better than area threshold, they are still not significantly different. This shows the limitation of SIFT+ANN when applied to practical urban traffic complex environments.

Results show 95% classification accuracy for large vehicles. The main rationale is that large vehicles are highly distinctive for area threshold as well as have enough SIFT features. Classification accuracy for motorcycles is 77.9%. Even though the results are better but are still not adequate. The errors here are due to lack of distinctive features. The accuracy of classifying small vehicles is 58%. It shows that ANN is unable to correctly distinguish between different types of small vehicles. The major reason for that are less



distinctive boundaries due which they are wrongly categorized as motorcycles or large vehicles. One of its causes is varying orientation due to camera viewing angle.

**Table 9: Comparison of classifiers based on confusion matrix**

Comparison of classification results			
M	Accuracy	Precision	Misclassify rate
Area Threshold	75.75%	57.40%	24.24%
SIFT+ANN Classifier	77.92%	86.20%	22.07%
SV	Accuracy	Precision	Misclassify rate
Area Threshold	52.81%	71.00%	47.18%
SIFT+ANN Classifier	58.00%	64.51%	42.00%
LV	Accuracy	Precision	Misclassify rate
Area Threshold	92.64%	30.43%	7.35%
SIFT+ANN Classifier	95.67%	43.75%	4.32%

## Conclusion

Existing vision algorithms have seldom been reported for complex urban traffic scenarios with dynamic and complex traffic trends. Similarly, reported results are generally on controlled traffic conditions which barely reflect the robustness of such algorithms. An effort is made to perform detection and classification of vehicles on challenging datasets of urban traffic using existing algorithms. The results were examined for online available datasets. Further two local, day time, complex urban free flowing traffic video feeds with varying camera viewing angle were acquired and analyzed. Even though it showed promising results for certain traffic lanes but it had errors for lanes affected by occlusion because of camera position, design and speed of vehicles. SIFT and ANN based classifiers perform better and show encouraging results in such complex scenarios. However, practical systems still require deep learning based robust classification schemes to get more accuracy.

## Acknowledgements

We are thankful to *Karachi Metropolitan Corporation* and *Think Transportation, Karachi* for the provision of datasets and valuable discussions.

## References

[1] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", IEEE International Conference on Computer Vision, 1999.  
 [2] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 1999.

[3] V. Kastinaki, M. Zervakis, K. Kalaitzakis, "A survey of video processing techniques for traffic applications," Image and Vision Computing, 2003.  
 [4] A. Prati, I. Mikic, M. M. Trivedi, R. Cucchiara., "Detecting Moving Shadows: Formulation, Algorithms and Evaluation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, 2003.  
 [5] G. Zhang, R. P. Avery, Y. Wang, "A Video-based Vehicle Detection and Classification System for Real-time Traffic Data Collection Using Uncalibrated Video Cameras", Transportation Research Board Annual Meeting, 2006.  
 [6] T. H. Chen, Y.F. Lin, T. Y. Chen, "Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance", IEEE International Conference on Innovative Computing Information and Control, 2007.  
 [7] A. Bosch, A. Zisserman, X Munoz, "Image Classification using Random Forests and Ferns", IEEE International Conference on Computer Vision, 2007.  
 [8] B. Leibe, K. Schindler, N. Cornelis, L.V. Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, 2008.  
 [9] C. Aydos, B. Hengst, W. Uther, "Kalman Filter Process Models for Urban Vehicle Tracking", IEEE International Conference on Intelligent Transportation Systems, 2009.  
 [10] N. Buch, M. Cracknell, J. Orwell, S. A. Velastin, "Vehicle localisation and classification in urban cctv streams," ITS World Congress, 2009.  
 [11] A. Vedaldi, B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," ACM international conference on Multimedia, 2010.  
 [12] N. Buch, S.A. Velastin, J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic," IEEE Transactions on Intelligent Transportation Systems, Vol. 12, 2011.  
 [13] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, S. Pankanti, "Attribute-based Vehicle Search in Crowded Surveillance Videos," ACM International Conference on Multimedia Retrieval, 2011.  
 [14] N. C. Mithun, N. U. Rashid, S. M. M. Rahman, "Detection and Classification of Vehicles From Video Using Multiple Time-Spatial Images," IEEE Transactions on Intelligent Transportation Systems, Vol. 13, 2012.  
 [15] S. Fazli, S. Mohammadi, M. Rahmani, "Neural Network based Vehicle Classification for Intelligent Traffic Control," International Journal of Software Engineering & Applications, Vol.3, 2012.  
 [16] MATLAB, "Motion based Multiple Object Tracking," Computer Vision Toolbox example, 2013.  
 [17] Y. Tang, C. Zhang, R. Gu, P. Li, B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance system," Springer International Journal on Multimedia Tools and Applications,



## Author Biography



*Muhammad Umair Arif received his BE in Electrical Engineering from NED University of Engineering and Technology, Pakistan (2009) and his MS degree in Fuzzy controls from National University of Sciences and Technology (NUST), Pakistan (2013). He is currently a PhD scholar at NUST. His work is focused on Computer Vision solutions for Intelligent Transportation Systems and their applications on Embedded systems.*



*Zain ul Abideen Lodhi received his BE in Telecommunication Engineering from Institute of Space Technology (2012). He is currently pursuing his MS from National University of Sciences and Technology (NUST) in Control Systems. His work is focused on Computer Vision system and Control of Swarm Robots.*



*Maheen Khan received her BE in Electrical Engineering from Bahria University. She is currently a research assistant working on implementation of Vision based algorithms using MATLAB as software platform and FPGA as hardware platform for Vehicle Detection and Classification.*



*Dr Raza has been working in academia and industry for around 10 years. He has completed his PhD in Electrical and Computer Engineering at Michigan State University (MSU), Michigan USA (2013). He is a recipient of prestigious US Fulbright scholarship, MSU Graduate Excellence fellowship and various other awards including HEC Indigenous fellowship. He has also worked at renowned Exploratory Computer Vision Group at IBM T. J. Watson Research Center, New York, USA. He is currently an Assistant Professor at National University of Sciences and Technology (NUST), Karachi, Pakistan. His broader areas of interest are visual analytics, situation awareness, activity and group behavior recognition, real time location systems, object localization/tracking and their application in robotics. In addition to working on core research he has a passion of incorporating STEM education in K-12 grades and comes with more than five years of teaching experience in USA.*