Visual Big Data Analytics for Traffic Monitoring in Smart City

Dinesh Singh, C. Vishnu and C. Krishna Mohan
Visual Learning and Intelligence Group (VIGIL),
Department of Computer Science and Engineering,
Indian Institute of Technology Hyderabad, Kandi, Sangareddy-502285, India
Email: {cs14resch11003, cs16mtech11021, ckm}@iith.ac.in

***Abstract*—The application such as video surveillance for traffic control in smart cities needs to analyze the large amount (hours/days) of video footage in order to locate the people who are violating the traffic rules. The traditional computer vision techniques are unable to analyze such a huge amount of visual data generated in real-time. So, there is a need for visual big data analytics which involves processing and analyzing large scale visual data such as images or videos to find semantic patterns that are useful for interpretation. In this paper, we propose a framework for visual big data analytics for automatic detection of bike-riders without helmet in city traffic. We also discuss challenges involved in visual big data analytics for traffic control in a city scale surveillance data and explore opportunities for future research.**

Keywords*—Visual Big Data Analytics, Smart City, Traffic Control, Machine Learning.

I. Introduction

Nowadays video surveillance systems become an essential equipment in order to keep a watch on any kind of criminal or anti law activity in modern civilizations. Law enforcement agencies in modern cites all over the world deployed large network of CCTV cameras covering all sensitive public areas of a city like airports, railway stations, and road network. The road traffic monitoring is most important from the point of view of tracking criminals, detecting traffic violators, detecting road accidents, collecting evidences for investigation, etc. Automatic decision making systems are desirable in order to catch various kind of traffic violators. Two-wheelers are becoming very popular mode of transportation everywhere, however, a high risk is associated with it because there is no protection to the head part of human body. So, the Governments impose the use of wearing helmets to protect the head part of body for the persons who are riding two-wheelers. Observing the usefulness of helmet, Governments have made it a punishable offense to ride a bike without helmet and have adopted manual strategies to catch the persons who are violating the traffic rules. But, manual system of tracking people who are violating the traffic rules is not a feasible solution due to involvement of humans, whose efficiency may decrease over a long duration [1]. Automation of this process is highly desirable for reliable and robust monitoring of these traffic rule violations. Also, it can significantly reduce the amount of human resources needed for traffic monitoring. To make the urban areas as smart city, many countries are adopting systems involving surveillance cameras at public places for round the clock security monitoring. So, this automated solution for traffic monitoring is also cost-effective because it is using the existing infrastructure and there is a significant reduction in the man power involved for doing the same.

However, in order to adopt such automatic solutions, certain issues need to be addressed like real-time implementation, occlusion, direction of motion, temporal changes in weather conditions, and the quality of video feed [2]. So, the processing significant amount of information in a time constraint manner is a challenging task. As such applications involve tasks like segmentation, feature extraction, classification, and tracking, in which a significant amount of information need to be processed in short duration to achieve the goal of real-time implementation [1] [3]. As stated in [1], successful framework for surveillance application should have useful properties such as real-time performance, fine tuning, and robust to sudden changes. Keeping these challenges and desired properties in mind, we propose a visual big data analytics framework based solution for automatic detection of bike-riders without helmet in real time from traffic surveillance camera network of a smart city.

To date many frameworks are proposed for road traffic monitoring using surveillance cameras. A traffic monitoring system includes object detection and tracking, behavioral analysis of traffic patterns, number plate recognition, and automated security and surveillance on video streams captured by surveillance cameras. T. Abdullah *et al.* [4] presented a framework for stream processing in cloud that is capable of detecting vehicles from the recorded video streams. This framework provides an end-to-end solution for video stream capture, storage, and analysis using a cloud based graphics processor unit (GPU) cluster. It empowers traffic control room operators by automating the process of vehicle identification and finding events of interest from the recorded video streams. An operator only specifies the analysis criteria and the duration of video streams to analyze. These video streams are then automatically fetched from the cloud storage, decoded and analyzed on a Hadoop based GPU cluster without operator intervention. It reduces the latency in video analysis process by porting its compute intensive parts to the GPU cluster. An automatic license plate recognition system is presented by Y. Chen *et al.* [5] using cloud computing in order to realize massive data analysis, which enables the detection and tracking of a target vehicle in a city with a given license plate number. It realizes a fully integrated system with a surveillance network of city scale, automatic large scale data retrieval and analysis, and combination of pattern recognition in order to achieve contextual information analysis. C. Zhang *et al.* [6]

have proposed a hybrid cloud model for video surveillance system with mixed-sensitivity video streams. The hybrid cloud is used to address the security issues by keeping sensitive data in the private cloud, while relieving seasonal workload by pushing computation to the public cloud. To enhance usability and reduce the cost, a middle-ware is used that seamlessly integrates the private cloud with public cloud and scheduled the tasks effectively. A stream processing model in this hybrid cloud optimizes overall monetary cost to be incurred on the public cloud with the constraints of resources, security, and Quality-of-Service (QoS).

In this paper, we discuss the visual big data analytics framework and its underlying techniques and application for devising a framework for automatic detection of bike-riders without helmet in real time from the video feeds from the surveillance network of the city. The proposed framework in first phase detects bike riders from surveillance videos by applying background subtraction and object segmentation methods. In second phase, it locates the riders head and extracts suitable features in order to detect whether the bike-rider is wearing a helmet or not. A consolidation approach is also presented for alarm generation in order to reduce the false alarms and improve reliability of the proposed approach. In order to evaluate our approach, we have provided the performance comparison of three widely used feature representations, namely, histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and local binary patterns (LBP) for classification. The experimental results show detection accuracy of 93.80% on the real world surveillance data. It is also been shown that proposed approach is computationally less expensive and performs in real-time with a processing time of 11.58 ms per frame.

The remainder of this paper is organized as follows : Section II presents the proposed approach for automatic detection of bike-riders without helmet. The proposed visual big data framework is presented in Section III. Section IV discusses experiments and results. The conclusion is provided in section V.

II. Proposed approach for Automatic Detection of Bike-riders without Helmet

This section presents the proposed approach for real-time detection of bike-riders without helmet which works in two phases. In the first step, all the bike-riders in the video frame are detected and in the second step, the head of the bike-rider is located and determine whether the rider is using a helmet or not. In order to reduce false alarm generations, we consolidate the results from consecutive frames for final alarm generation. The block diagram in Fig. 1 shows the various steps of proposed framework such as background subtraction, feature extraction, object classification using sample frames.

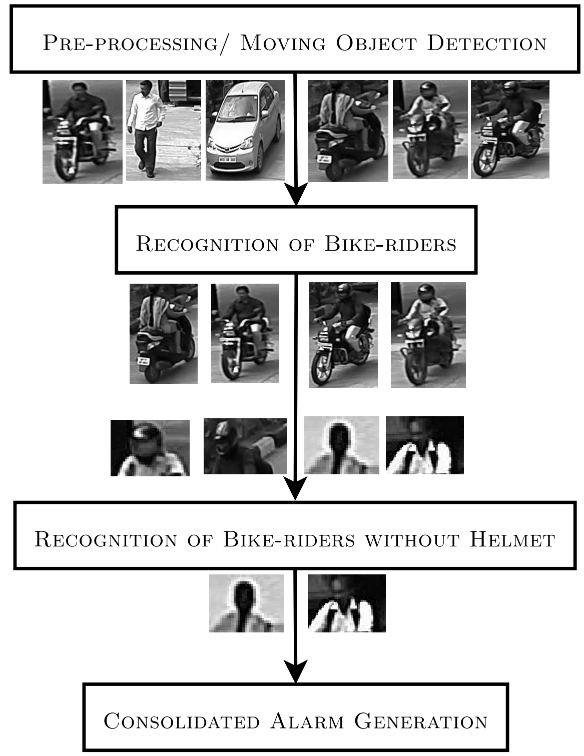

Fig. 1. Block diagram of proposed approach for automatic detection of bike-riders without helmet.

A. Pre-processing/ Moving Object Detection

In order to distinguish between moving and static objects, we apply background subtraction on gray-scale frames. The background subtraction method in [7] is used to separate the objects in motion such as bike, humans, cars from static objects such as trees, roads, and buildings. However, there are certain challenges when dealing with data from single fixed camera. Environment conditions like illumination variance over the day, shadows, shaking tree branches, and other sudden changes make it difficult to recover and update background from continuous stream of frames. In case of complex and variable situations, single Gaussian is not sufficient to completely model these variations [8]. Due to this reason, for each pixel, it is necessary to use variable number of Gaussian models. Here K, the number of Gaussian components for each pixel is kept in between 3 and 5, which is determined empirically. However, some errors may still occur due to the presence of highly occluded objects and merged shadows. Background model is approximated using on-line clustering method proposed in [7]. Subtracting background mask from current frame results in foreground objects. Gaussian filter is applied to foreground mask to reduce noise and then transformed into binary image using clustering based thresholding [9]. Morphological operations specifically close operations are used to further process the foreground mask to achieve better distinction between objects. Next, this processed frame is segmented into parts based on object boundaries. Background subtraction method retrieves only moving objects and ignore non-useful details such as static objects. There may still be many moving objects which are not of our interest such as humans, cars etc. These objects are filtered based on their area. The objective behind this is to only consider objects which are more likely to fall in bike-riders category. It helps in reducing the complexity of further steps.

B. Recognition of Bike-riders

This step involves detection of bike-riders in a frame. It uses objects, the potential bike-riders returned by background modeling and classify them as ‘bike-rider’ vs ‘others’, based on their visual features.

Object classification requires some suitable representation of visual features. In literature, histogram of oriented gradients (HOG) [10], scale invariant feature transform (SIFT) [11], and local binary patterns (LBP) [12] are proven to be efficient for object detection. For this purpose, we analyze three features viz; HOG, SIFT and LBP. HOG descriptors which are proven to be very efficient in object detection. These descriptors capture local shapes through gradients. SIFT tries to capture key-points in the image. For each key-point, it extracts feature vectors. Scale, rotation, and illumination invariance of these descriptors provide robustness in varying conditions. We used bag-of-words(BoW) technique to create a dictionary. Then mapping SIFT descriptors to dictionary words results in feature vectors. These feature vectors are used to determine similarity between images. LBP captures texture information in the frame. For each pixel, a binary number is assigned by thresholding the pixels in the circular neighborhood and the frequency histogram of these numbers is the feature vector. Fig. 2 visualizes the patterns of phase-I classification in 2-D space using t-SNE [13]. The distribution of the HOG feature vectors show that the two classes i.e 'bike-riders' (Positive class shown in blue crosses) and 'others' (Negative class shown in red dots) fall in almost distinct regions with only few exceptions. This shows that the feature vectors efficiently represent the activity and contains discriminative information, which further gives hope for good classification accuracy.

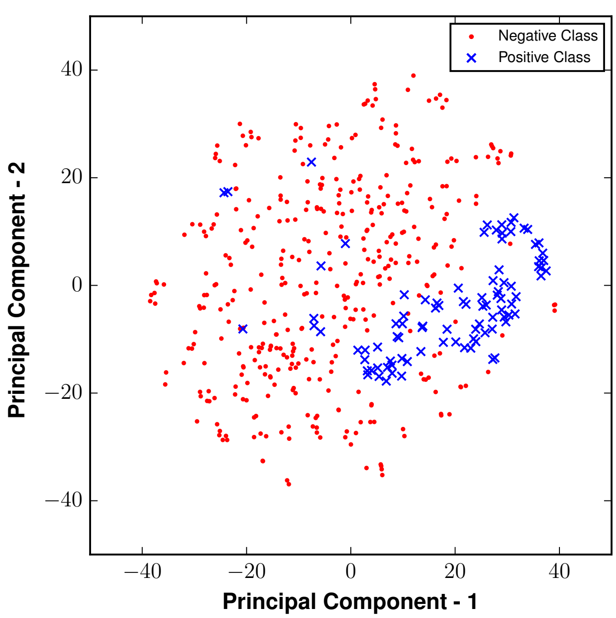

Fig. 2. Visualization of HOG feature vectors for 'bike-rider vs others' classification using t-SNE [13]. Blue cross represents bike-rider class and Red dot represents non bike-rider class [Best viewed in color].

After feature extraction, next task is to classify them as 'bike-riders' vs 'other' objects. Thus, this requires a binary classifier. Any binary classifier can be used here, however, we choose SVM due to its robustness in classification performance even when trained from less number of feature vectors. Also, we use different kernels such as linear, sigmoid (MLP), radial basis function (RBF) to arrive at best hyper-plane.

C. Recognition of Bike-riders Without Helmet

After the bike-riders are detected in the previous step, the next step is to determine if bike rider is using a helmet or not. Usual face detection algorithms would not be sufficient for this phase due to following reasons : i) Low resolution poses a great challenge to capture facial details such as eyes, nose, mouth. ii) Angle of movement of bike may be at obtuse angles. In such cases, face may not be visible at all. So, the proposed framework detects region around head and then proceed to determine whether bike-rider is using helmet or not. In order to locate the head of bike-rider, the proposed framework uses the fact that appropriate location of helmet will probably be in upper areas of bike rider. For that we consider only the upper one fourth part of object. Identified region around head of bike-rider is used to determine if bike-rider is using the helmet or not. To achieve this, similar features as used in phase-I i.e. HOG, SIFT and LBP are used. Fig. 3 visualizes the patterns for phase-II in 2-D using t-SNE [13]. The distribution of the HOG feature vectors show that the two classes i.e 'non-helmet' (Positive class shown in blue cross) and 'helmet' (Negative class shown in red dot) fall in overlapping regions which shows the complexity of representation. However, Table I shows that the generated feature vectors contain significant discriminative information in order to achieve good classification accuracy.

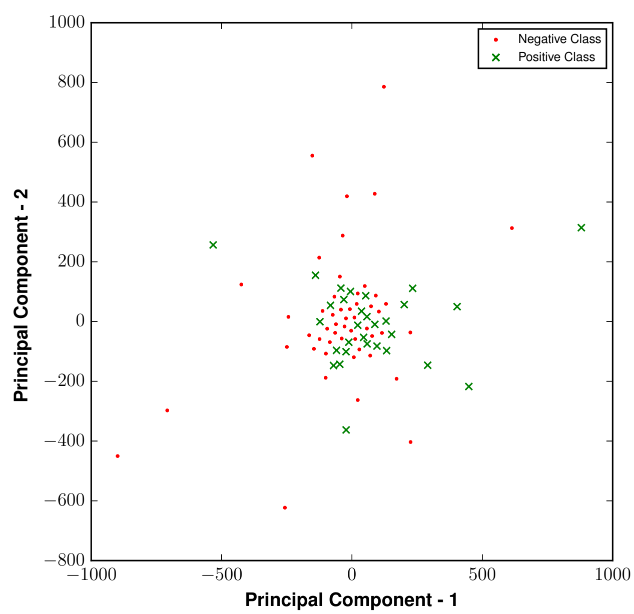

Fig. 3. Visualization of HOG feature vectors for 'helmet vs non-helmet' classification using t-SNE [13]. Red dot indicates helmet class and Green cross indicates non-helmet class [Best viewed in color].

The method needs to determine if biker is violating the law i.e. not using helmet. For this purpose, we consider two classes : i) Bike-rider not using helmet (Positive Result), and ii) Biker using helmet (Negative Result). The support vector machine (SVM) is used for classification using extracted features from previous step. To analyze the classification results and identify the best solution, different combination of features

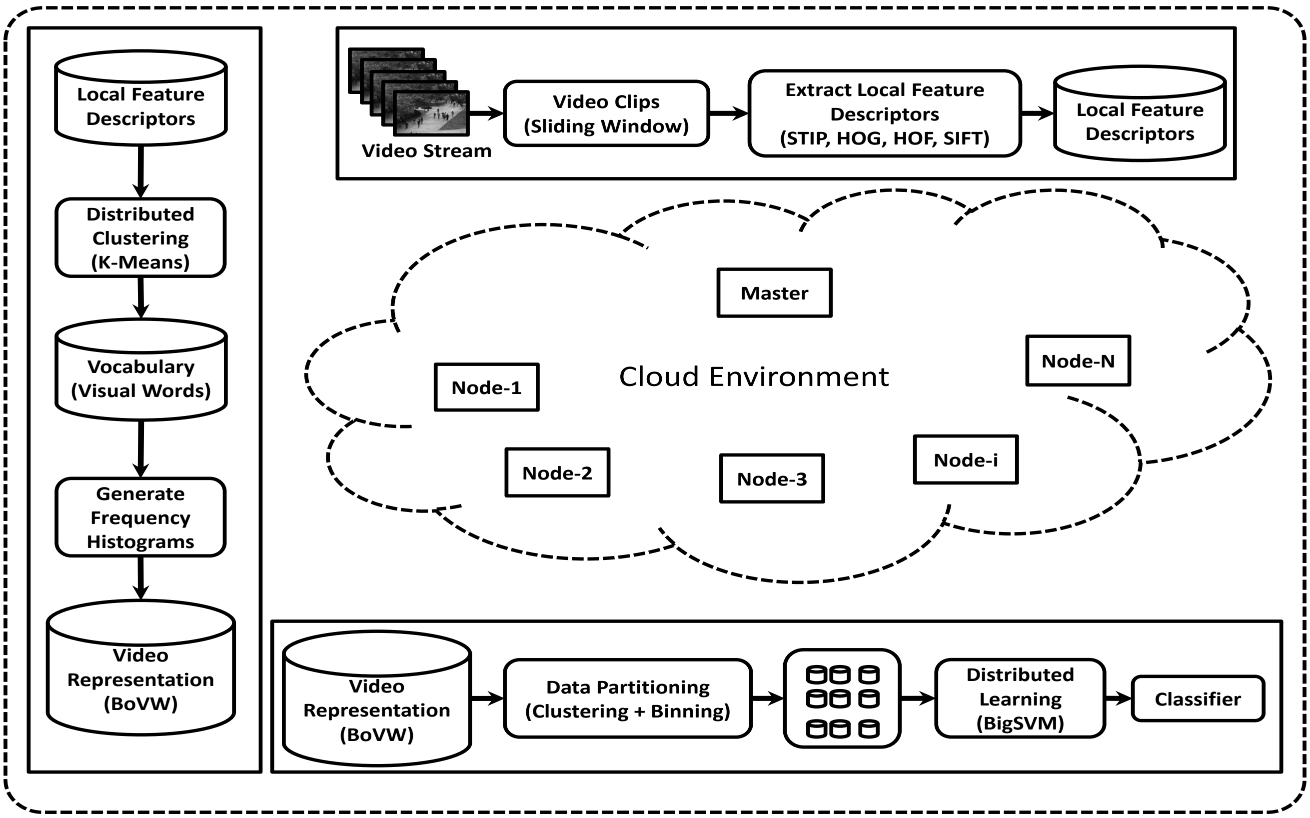

Fig. 4. Block diagram of visual big data framework over the cloud for traffic monitoring using large surveillance camera network of a smart city. Depicting use of distributed processing of feature representation using bag-of-words (BoW) and classification using distributed support vector machine (SVM).

and kernels are used. Results along with analysis is included in *Result* section.

D. Consolidated Alarm Generation

From earlier phases, we obtain local results i.e. whether bike rider is using helmet or not, in a frame. However, till now the correlation between continuous frames is neglected. So, in order to reduce false alarms, we consolidate local results. Consider y_i be label for i^{th} frame which is either +1 or -1. If for past n frames, $\frac{1}{n}\sum_{i=1}^{n}(y_i = 1) > T_f$, then framework triggers violation alarm. Here T_f is the threshold value which is determined empirically. In our case, the value of $T_f = 0.8$ and $n = 4$ were used. A combination of independent local results from frames is used for final global decision i.e. biker is using or not using helmet.

III. Visual Big Data Analytics Framework

In order to apply the proposed approach on a city scale surveillance camera data network, we proposed a framework called visual big data framework over the cloud. The entire framework for visual big data analytics over the cloud involves visual computing, big data analytics, and cloud computing. Visual computing involves processing and analyzing visual data such as images or videos in order to find semantic patterns that are useful for interpretation. The large scale video surveillance application such as traffic monitoring in smart cities needs to analyze the large amount (hours/days) of video footage in order to locate unusual patterns. The real-time analysis of such a big visual data is a challenging task due to involvement of large volume of data generated at high velocity. Thus, the overall problem becomes a visual big data problem for which existing visual computing techniques are unable to meet the desired performance. Also, the up-front infrastructure investment is costly, so we are leveraging the benefits of cloud computing in order to get computing resources on-demand at reduced cost. This framework uses a hybrid cloud architecture where continuously used resources are provided by private cloud. For computationally expensive tasks, it leverages the benefit of public cloud. For example, in surveillance system, some computing resources are used continuously viz; cameras, and attached computing resources to check against any malicious activity on a trained model. For this requirement, we can have our own physical infrastructure, a private cloud or a public cloud. For the purpose of long term storage of data and computationally expensive training process, it utilizes the services provided by a public cloud. As training is a short-term need, it is a cost effective solution. In cloud, we can setup a suitable cluster for distributed processing of visual data using MapReduce according to the need. Fig. 4 shows the block diagram of visual big data framework over the cloud for traffic monitoring using large surveillance camera network of a smart city. The visual computing applications consists of two main tasks, namely, feature extraction and classification. Both these task are computationally complex and while dealing large visual data, the situation further becomes worse. Here, we are using cloud computing to full fill our needs of short term computing resources. Training over a large datasets increase the performance of the pattern recognition tasks. The proposed approach involves computationally complex tasks like k-means clustering for vocabulary generation in bag-of-words (BoW) feature representation and training of support vector machine (SVM) classifier. For that, we propose a distributed framework shown in Fig. 4, where we distribute computations for BoW and SVM over a distributed environment like cluster or cloud. Bag-of-words approach makes use of clustering, however, the enlarged volumes of data becomes a challenge to clustering. Many distributed implementations are available for variety of clustering algorithm, however, we use PKMeans, a parallel k-means clustering algorithm proposed by W. Zhao *et al.* [14] for MapReduce. There are three functions in PKMeans, namely, map, combiner, and reduce. The map function assigns samples to a closest center, a combiner function calculates the sum of points assigned to each cluster by a single map function in order to reduce communication, while reduce function computes new centers from the arrays of partial sum by each map function. For SVM, we use divide and conquer (DCSVM) proposed by Hsieh *et al.* [15], where the training samples are divided into smaller partitions using k-means clustering and local SVM models are trained for each smaller partition independently using LIBSVM library. The support vectors of local SVMs are then passed as an input to next level SVM. Finally, a global SVM model is obtained via combining local SVM models. This reduces the overall time taken, specially when dealing with large scale data.

IV. Experiments and Results

The experiments are conducted on a cluster of two machines running Ubuntu 16.04 Xenial Xerus havining specifications Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz×48 processor, 128GB RAM with NVIDIA Corporation GK110GL [Tesla K20c]×2 GPUs and Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz×16 processor, 64GB RAM with NVIDIA Corporation GK110GL [Tesla K20c]×6 GPUs, respectively. Programs are written in C++, where for video processing we use OpenCV 3.0, the bag-of-words and SVM are implemented using OpenMP and OpenMPI with distributed k-means.

A. Dataset Used

We collected our own data from the surveillance system at Indian Institute of Technology Hyderabad campus because there is no public data set available. Total two hour surveillance data is collected at 30 frame per second. Fig. 5 present samples from the collected dataset. First one hour video is used for training the model and remaining for testing. Training video contains 42 bikes, 13 cars, and 40 humans. Whereas, testing video contains 63 bikes, 25 cars, and 66 humans.

B. Results and Discussion

In this section, we present experimental results and discuss the suitability of the best performing representation and model over the others. Table. I presents experimental results. In order to validate the performance of each combination of representation and model, we conducted experiments using 5-fold cross validation. The experimental results in Table I for bike vs. non-bike classification, show that average performance of classification using SIFT and LBP features is almost similar. Also, the performance of classification using HOG with MLP and RBF kernels is similar to the performance of SIFT and LBP. However, HOG with *linear* kernel performs better than all other combinations, because feature vector for this representation is sparse in nature which is suitable for linear kernel. We can observe that for head vs. helmet classification, average performance of classification using SIFT and LBP is almost similar. Also, the performance of classification using HOG with MLP and RBF kernel is similar to the performance of SIFT and LBP. However, HOG with linear kernel performs better than all other combinations.

Fig. 5. Sample frames from dataset

TABLE I. Performance of classification (%) of Detection of Bike-rider without Helmet

Feature	Kernel	Bike vs. Non-bike	head vs. helmet
HOG	*Linear*	**98.88**	**93.80**
	MLP	82.89	64.50
	RBF	82.89	64.50
SIFT	*Linear*	**82.89**	**64.51**
	MLP	82.89	64.51
	RBF	82.89	64.51
LBP	*Linear*	**82.89**	**64.53**
	MLP	82.89	64.53
	RBF	82.89	64.53

From the results presented in Table I, it can be observed that using HOG descriptors help in achieving best performance.

Fig. 6 & Fig. 7 presents ROC curves for performance of classifiers in detection of bike-riders and detection of bike-riders with or without helmet, respectively. Fig. 6 clearly shows that the accuracy is above 95% with a low false alarm rate less than 1% and area under curve (AUC) is 0.9726. Similarly, Fig. 7 clearly shows that the accuracy is above 90% with a low false alarm rate less than 1% and AUC is 0.9328.

C. Computational Complexity

To test the performance, a surveillance video of around one hour at 30 fps i.e. 107500 frames was used. The proposed framework processed the full data in 1245.52 secs i.e. 11.58 ms per frame. However, frame generation time is 33.33 ms, so

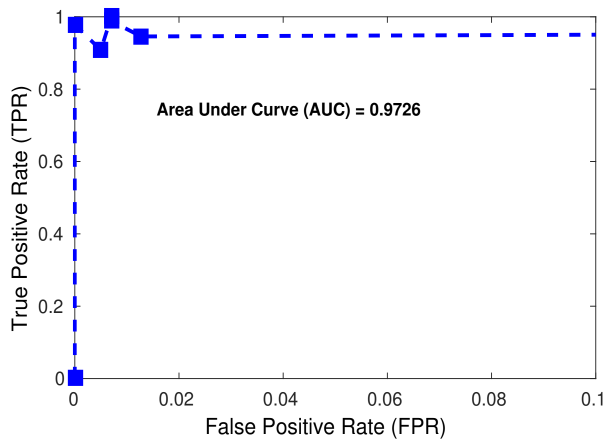

Fig. 6. ROC curve for classification of 'bike-riders' vs. 'others' in phase-I showing high area under the curve

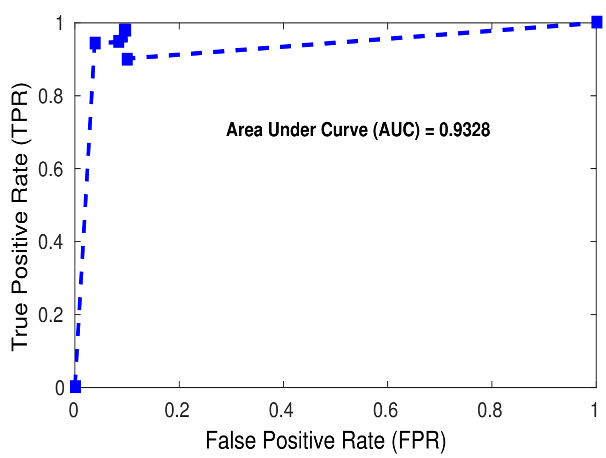

Fig. 7. ROC curve for classification of 'bike-rider with helmet' vs. 'bike-rider without helmet' in phase-II showing high area under the curve

the proposed framework is able to process and return desired results in real-time. Result included in section IV(B) shows that accuracy of proposed approach is either better or comparable to related work presented in [16] [17] [18] [19].

V. Conclusion

In this paper, we propose a visual big data analytics based framework for real-time detection of traffic rule violators who ride bike without using helmet in a city scale surveillance camera network. The proposed framework will also assist the traffic police for detecting such violators in odd environmental conditions viz; hot sun, etc. Experimental results demonstrate the high classification performance which is 98.88% and 93.80% for detection of bike-rider and detection of violators, respectively. Average time taken to process a frame is ≈11 ms, which is suitable for real time use. Also, the proposed framework automatically adapts to new scenarios if required, with slight tuning. This framework can be extended for detection of other rule violations as well as to detect and report number plates of violators.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, March 2008.

[2] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," in *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN)*, Vancouver, Canada, Jul.24-29 2016.

[3] B. Duan, W. Liu, P. Fu, C. Yang, X. Wen, and H. Yuan, "Real-time on-road vehicle and motorcycle detection using a single camera," in *Procs. of the IEEE Int. Conf. on Industrial Technology (ICIT)*, 10-13 Feb 2009, pp. 1–6.

[4] T. Abdullah, A. Anjum, M. F. Tariq, Y. Baltaci, and N. Antonopoulos, "Traffic Monitoring Using Video Analytics in Clouds," in *IEEE/ACM International Conference on Utility and Cloud Computing*, 2014, pp. 39–48.

[5] Y. L. Chen, T. S. Chen, T. W. Huang, L. C. Yin, S. Y. Wang, and T. C. Chiueh, "Intelligent urban video surveillance system for automatic vehicle detection and tracking in clouds," in *International Conference on Advanced Information Networking and Applications (AINA)*, 2013, pp. 814–821.

[6] C. Zhang and E.-c. Chang, "Processing of Mixed-Sensitivity Video Surveillance Streams on Hybrid Clouds," in *IEEE International Conference on Cloud Computing*, 2014, pp. 9–16.

[7] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, vol. 2, Aug.23-26 2004, pp. 28–31.

[8] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1999, pp. 246–252.

[9] "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Procs. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 886–893.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] Z. Guo, D. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, June 2010.

[13] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[14] W. Zhao, H. Ma, and Q. He, "Parallel K -Means Clustering Based on MapReduce," in *International Conference on Cloud Computing*, 2009, pp. 674–679.

[15] C.-J. Hsieh, S. Si, and I. Dhillon, "A Divide-and-Conquer Solver for Kernel Support Vector Machines," in *Proc. of Int. Conf. on Machine Learning (ICML)*, Beijing, 21-26 Jun 2014, pp. 566–574.

[16] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *Intelligent Transport Systems (IET)*, vol. 6, no. 3, pp. 259–269, September 2012.

[17] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, "Automatic detection of motorcyclists without helmet," in *Computing Conf. (CLEI), XXXIX Latin American*, Oct 2013, pp. 1–7.

[18] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Int. Conf. of Image and Vision Computing New Zealand (IVCNZ)*, Nov 2013, pp. 35–40.

[19] R. Rodrigues Veloso e Silva, K. Teixeira Aires, and R. De Melo Souza Veras, "Helmet detection on motorcyclists using image descriptors and classifiers," in *Procs. of the Graphics, Patterns and Images (SIBGRAPI)*, Aug 2014, pp. 141–148.