Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time

Kunal Dahiya, Dinesh Singh, C. Krishna Mohan
Visual Learning and Intelligence Group (VIGIL),
Department of Computer Science and Engineering,
Indian Institute of Technology, Hyderabad, India
Email: {cs11b15m000001, cs14resch11003, ckm}@iith.ac.in

***Abstract*—In this paper, we propose an approach for automatic detection of bike-riders without helmet using surveillance videos in real time. The proposed approach first detects bike riders from surveillance video using background subtraction and object segmentation. Then it determines whether bike-rider is using a helmet or not using visual features and binary classifier. Also, we present a consolidation approach for violation reporting which helps in improving reliability of the proposed approach. In order to evaluate our approach, we have provided a 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 has 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.**

I. Introduction

Two-wheeler is a very popular mode of transportation in almost every country. However, there is a high risk involved because of less protection. To reduce the involved risk, it is highly desirable for bike-riders to use helmet. 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 violators. However, the existing video surveillance based methods are passive and require significant human assistance. In general, such systems are infeasible due to involvement of humans, whose efficiency decreases over long duration [1]. Automation of this process is highly desirable for reliable and robust monitoring of these violations as well as it also significantly reduces the amount of human resources needed. Also, many countries are adopting systems involving surveillance cameras at public places. So, the solution for detecting violators using the existing infrastructure is also cost-effective.

However, in order to adopt such automatic solutions certain challenges need to be addressed: 1) *Real-time Implementation:* 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] [2]. 2) *Occlusion:* In real life scenarios, the dynamic objects usually occlude each other due to which object of interest may only be partially visible. Segmentation and classification become difficult for these partially visible objects [3]. 3) *Direction of Motion:* 3-dimensional objects in general have different appearance from different angles. It is well known that accuracy of classifiers depends on features used which in turn depends on angle to some extent. A reasonable example is to consider appearance of a bike-rider from front view and side view. 4) *Temporal Changes in Conditions:* Over time, there are many changes in environment conditions such as illumination, shadows, etc. There may be subtle or immediate changes which increase complexity of tasks like background modelling. 5) *Quality of Video Feed:* Generally, CCTV cameras capture low resolution video. Also, conditions such as low light, bad weather complicate it further. Due to such limitations, tasks such as segmentation, classification and tracking become even more difficult. As stated in [1], successful framework for surveillance application should have useful properties such as *real-time performance, fine tuning, robust to sudden changes and predictive*. Keeping these challenges and desired properties in mind, we propose a method for automatic detection of bike-riders without helmet using feed from existing security cameras, which works in real time.

The remainder of this paper is organized as follows : Section II reviews the related work with their strengths and shortcomings. The proposed approach is presented in Section III. Section IV provides all the experimental details, results and their analysis. The last section summarizes the paper.

II. Existing Work

Automatic detection of bike-riders without helmet falls under broad category of anomaly detection in surveillance videos. As explained in [4], effective automatic surveillance system generally involve following tasks: environment modeling, detection, tracking and classification of moving objects. In [5], Chiverton proposed an approach which uses geometrical shape of helmet and illumination variance at different portions of the helmet. It uses circle arc detection method based on the Hough transform. The major limitation of this approach is that it tries to locate helmet in the full frame which is computationally expensive and also it may often confuse other similar shaped objects as helmet. Also, it oversees the fact that helmet is relevant only in case of bike-rider. In [6], Chen *et al.* proposed an efficient approach to detect and track

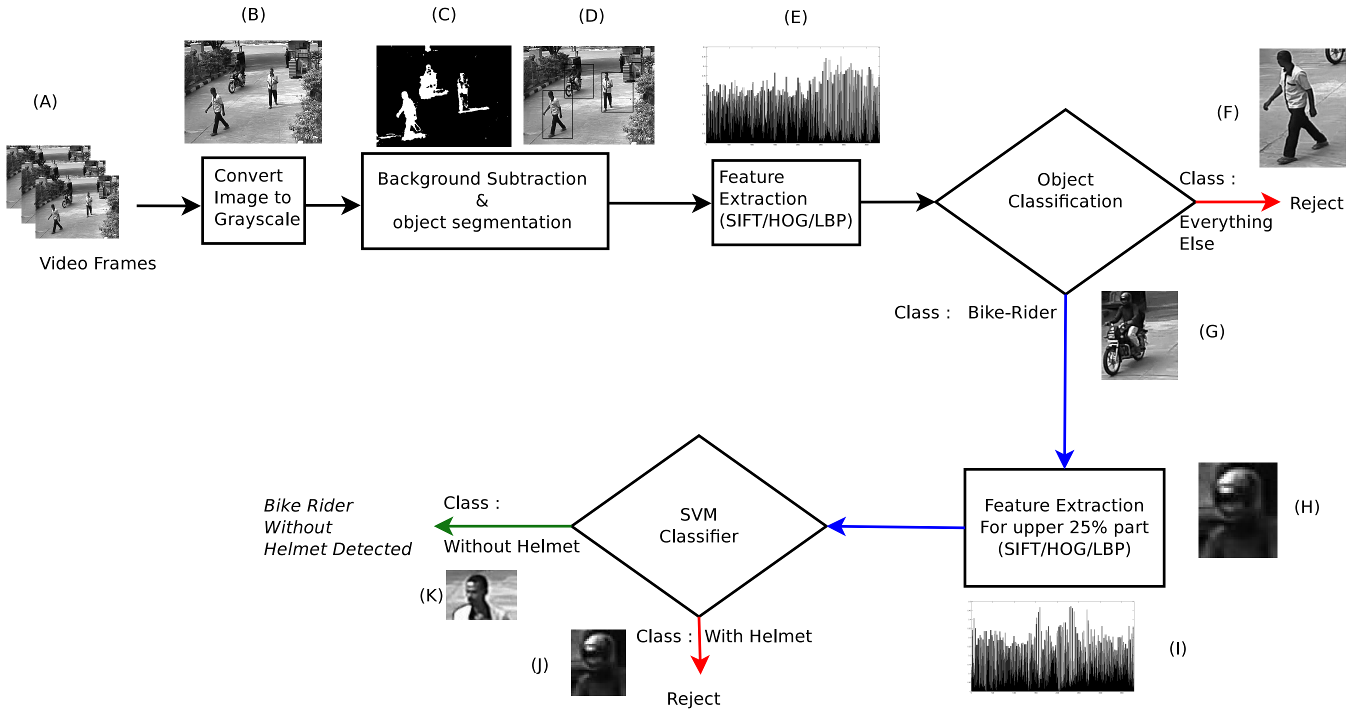
(A)
Video Frames
(B)
Convert Image to Grayscale
(C)
(D)
Background Subtraction & object segmentation
(E)
Feature Extraction (SIFT/HOG/LBP)
Object Classification
(F)
Class : Everything Else
Reject
Class : Bike-Rider
(G)
Feature Extraction For upper 25% part (SIFT/HOG/LBP)
(H)
(I)
SVM Classifier
Class : Without Helmet
Bike Rider Without Helmet Detected
(K)
Class : With Helmet
(J)
Reject

Fig. 1. Proposed approach for detection of bike-riders without helmet. A) Input frame sequence, B) A sample frame, C) Foreground mask for sample frame, D) Bounding box around foreground objects, E) Sample features of objects from D, F) Object classification as non-bike rider, G) Object classification as bike-rider, H) Localized head of the bike-rider, I) Sample Features of objects from H, J) Bike-rider classified as 'with helmet' class and, K) Bike-rider classified as 'without helmet' class.

vehicles in urban traffic. It uses Gaussian mixture model along with a strategy to refine foreground blob in order to extract foreground. It tracks a vehicle using Kalman filter and refine classification using majority voting. In [2], Duan *et al.* suggest a robust approach for tracking of vehicles in real-time from single camera. In order to accelerate the computation, it used integrated memory array processor (IMAP). However, it is not an efficient solution due to its requirement of dedicated hardware. In [7] [8], Silva *et al.* proposed an approach which starts with detection of bike-riders. Then it locates the head of bike-riders by applying Hough transform and then classifies it as head or helmet. However, Hough transform for locating head of bike-rider can be computationally expensive. Also, in [8] experiments are performed on static images only. Broadly, there are two major limitations in the existing work discussed above. Firstly, suggested approaches are either computationally very expensive [5] [7] or passive in nature [2] [8] which are not suitable for real time performance. Secondly, the correlation between the frames is underutilized for final decisions [5] [7], as the results from consecutive frames can be combined in order to raise more reliable alarms for violations. The proposed approach overcome above discussed limitations by providing an efficient solution which is suitable for real-time application.

III. Proposed Work

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

As helmet is relevant only in case of moving bike-riders, so processing full frame becomes computational overhead which does not add any value to detection rate. In order to proceed further, we apply background subtraction on gray-scale frames, with an intention to distinguish between moving and static objects. Next, we present steps involved in background modeling.

Background Modeling: Initially, the background subtraction method in [9] 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 [10]. Due to this reason, for each pixel, it is necessary to use variable number of Gaussian models. Here K, number of Gaussian components for each pixel is kept in between 3 and 5, which is determined empirically. Variable number of Gaussian components enables the background model to easily adjust it's parameters according to situation. However, some errors may still occur due to presence of highly occluded objects and merged shadows. Let us consider $I^1, I^2....I^t$ be the intensity of a pixel for past t, consecutive frames. Then at time t probability of observing intensity value for a pixel is given by:

$$P(I^t) = \sum_{j=1}^{K} w_j^t \times \eta(I^t, \mu_j^t, \sigma_j^t), \tag{1}$$

where, w_j^t is weight and $\eta(\cdot, \cdot, \cdot)$ is j^{th} Gaussian probability density function with mean μ_j^t and σ_j^t as variance at time t. For each pixel, the Gaussian components with low variance and high weight correspond to background class and others with high variance correspond to foreground class. At time t, the pixel intensity I^t is checked against all Gaussian components. If j^{th} component satisfies the condition :

$$\left|\mu_j^t - I^t\right| < e_j \sigma_j^t, \tag{2}$$

then j^{th} component is considered to be a match. Also, the current pixel is classified as background or foreground according to the class of j^{th} Gaussian model. The weight update rule is given by :

$$w_j^t = (1-\alpha) w_j^{t-1} + \alpha(M_j^t), \tag{3}$$

$$M_j^t = \begin{cases} 0, & \text{for matched model} \\ 1, & \text{otherwise ,} \end{cases} \tag{4}$$

where, α is learning rate which determines how frequently parameters are adjusted. Here, e_j is a threshold which has significant impact when different regions have different lightning. Generally the value of e_j is kept around 3, as $\mu^t \pm 3\sigma_j^t$ accounts for approximately 99% of data [9]. Also, other parameters of matched models are updated as:

$$\mu^t = (1-\rho)\mu^{t-1} + \rho I^t, \tag{5}$$

$$(\sigma^2)^{(t)} = (1-\rho)(\sigma^2)^{(t-1)} + \rho(I^t - \mu^t)^2. \tag{6}$$

Here, $\rho = \eta(I^t | \mu_j, \sigma_j)$. When there is no matched component, a new Gaussian model is created with current pixel value as mean, low prior weight and high variance. This newly created model replaces the least probable component or added as a new component if maximum number of components is reached or not, respectively. Background model is approximated using on-line clustering method proposed in [9]. Subtracting background mask from current frame results in foreground mask. In order to segment foreground mask as objects, image processing operations such as noise filter, morphological operation are used. Gaussian filter is applied to Foreground mask to reduce noise and then transformed into binary image using clustering based thresholding [11]. Morphological operations specifically close operation 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. Still there may be many moving objects which are not of our interest such as humans, cars etc. These objects are filtered based on their area. Let $\mathbf{B}_j$ be the j^{th} object with area a_j then B_j will be selected if $T_l < a_j < T_h$. Here T_l and T_h are threshold for minimum and maximum area, respectively. The method assumes that for a fixed camera, area of closing boundary of bikes is well differentiated from objects with very large area such as bus or very small area such as noise. 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.

A. *Phase-I: Detection Bike-riders*

This phase involves detection of bike-riders in a frame. This step uses objects $\mathbf{B}_j' s$, the potential bike-riders returned by background modeling step and classify them as 'bike-rider' vs 'others', based on their visual features. This phase involves two steps : feature extraction and classification.

1) Feature Extraction : Object classification requires some suitable representation of visual features. In literature, HOG, SIFT and LBP are proven to be efficient for object detection. For this purpose, we analyze following features :

- Histogram of Oriented Gradients [12] : HOG descriptors are proven to be very efficient in object detection. These descriptors capture local shapes through gradients. We used 9 bins, 8×8 pixels per cell and 2×2 cells per block. The resulting feature vector is $\mathbf{h}$, where $\mathbf{h} \in \mathbb{R}^n$, and n is 3780.
- Scale Invariant Feature Transform [13] : This approach 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 technique to create a vocabulary $\mathbf{V}$ of size 5000. Then mapping SIFT descriptors to $\mathbf{V}$ results in feature vector $\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^n$, and n is 5000. Feature vector $\mathbf{s}$ is used to determine similarity between images.
- Local Binary Patterns : These features capture texture information in the frame. For each pixel, a binary number is assigned by thresholding the pixels in the circular neighborhood [14] gives feature vector $\mathbf{l} \in \mathbb{R}^n$, where n is 26.

Fig. 2 visualizes the patterns of phase-I classification in 2-D space using t-SNE [15]. 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.

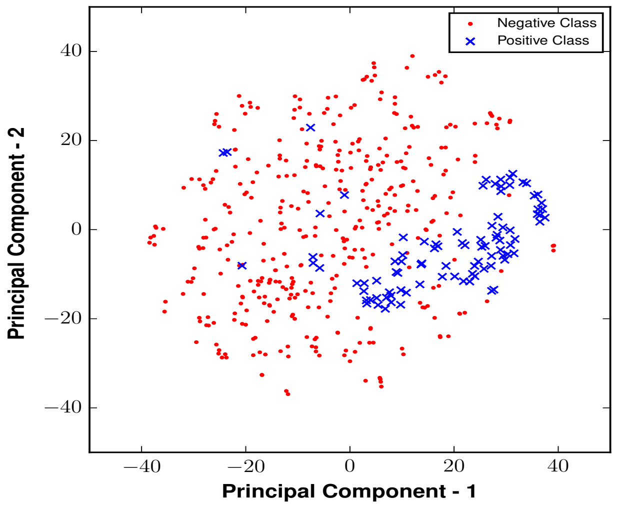

Fig. 2. Visualization of HOG feature vectors for 'bike-rider vs others' classification using t-SNE [15]. Blue cross represent bike-rider class and Red dot represent non bike-rider class [Best viewed in color]

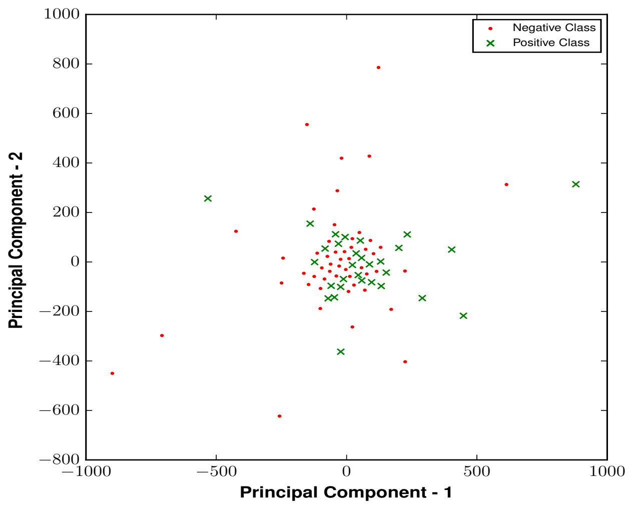

Fig. 3. Visualization of HOG feature vectors for 'helmet vs non-helmet' classification using t-SNE [15]. Red dots indicate helmet class and Green cross indicate non-helmet class [Best viewed in color]

2) Classification: After feature extraction, next step 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.

B. Phase-II: Detection of Bike-riders Without Helmet

After the bike-riders are detected in the previous phase, 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 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, proposed framework uses the fact that appropriate location of helmet will probably be in upper areas of bike rider. Consider $\mathbf{O}_{1/4}$ be upper one fourth part of object, and $\mathbf{B}_{1/4}$ be upper one fourth part of same object in binary, taken from background modeling step. For a moving bike, pixels in head region will have intensity of 1 i.e. white in $\mathbf{B}_{1/4}$. So, $\mathbf{B}_{1/4} \land \mathbf{O}_{1/4}$ gives region only around head. This step is very efficient which is reflected in our classification results for phase-II. Also, proposed approach is computationally less expensive than circular Hough transform which is used in related literature [7] [8] [16], as time complexity of logical "and" operation is $O(n)$ which is lower than $O(n^2)$ of circular Hough Transfrom [17].

1) Feature Extraction: 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 [15]. 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 II shows that the generated feature vectors contain significant discriminative information in order to achieve good classification accuracy.

2) Classification: 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 to classify using extracted features from previous step. To analyze the classification results and identify the best solution, different combination of features and kernels are used. Results along with analysis is included in *Result* section.

C. Consolidation of Results

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 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.

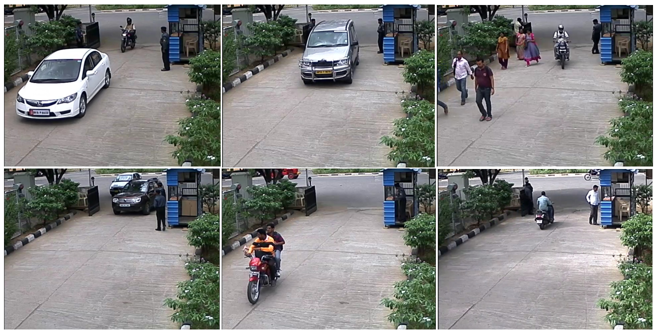

Fig. 4. Sample frames from dataset

IV. Experiments and Results

For purpose of related experiments, standalone Linux machine with specifications Intel Xeon(R) CPU E5620@ 2.40GHz x 8 was used. In our experiments, we used OpenCV 3.0 and scikit-learn 0.16 [18].

A. Dataset Used

As there is no public data set available for this purpose, we collected our own data from the surveillance system at Indian Institute of Technology Hyderabad. Here, we collected 2 hour surveillance data with frame rate of 30 fps. We used 1^{st} hour video for training the model and remaining for testing. Training video contain 42 bikes, 13 cars and 40 humans. Whereas, testing video contain 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 results for bike-rider detection using different features viz; HOG, SIFT, LBP and kernels viz; linear, sigmoid (MLP), radial basis function (RBF). 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 show that average performance of classification using SIFT and LBP 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 a suitable for linear kernel. Table I displays the accuracy of detecting a bike-rider in a frame.

Table II presents results for detection of bike-rider with or without helmet using different features viz; HOG, SIFT, LBP and kernels viz; linear, MLP, RBF. In order to validate the performance of each combination of representation and model, we conducted experiments using 5-fold cross validation. From Table II we can observe that 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.

TABLE I
PERFORMANCE OF PHASE-I CLASSIFICATION (%) OF DETECTION OF BIKE-RIDER

Feature	Kernel	Fold1	Fold2	Fold3	Fold4	Fold5	Avg.
HOG	*Linear*	97.93	99.59	98.35	99.38	99.17	**98.88**
	MLP	80.99	80.99	84.30	84.71	83.47	82.89
	RBF	80.99	80.99	84.30	84.71	83.47	82.89
SIFT	*Linear*	80.79	84.30	83.68	83.47	82.23	**82.89**
	MLP	80.79	84.30	83.68	83.47	82.23	82.89
	RBF	80.79	84.30	83.68	83.47	82.23	82.89
LBP	*Linear*	82.64	84.71	81.61	82.44	83.06	**82.89**
	MLP	82.64	84.71	81.61	82.44	83.06	82.89
	RBF	82.64	84.71	81.61	82.44	83.06	82.89

TABLE II
PERFORMANCE OF PHASE-II CLASSIFICATION (%) OF 'BIKE-RIDER WITH HELMET' VS 'BIKE-RIDER WITHOUT HELMET'

Feature	Kernel	Fold1	Fold2	Fold3	Fold4	Fold5	Avg.
HOG	*Linear*	90.12	95.06	93.83	95.00	95.00	**93.80**
	MLP	62.96	67.90	70.37	61.25	60.00	64.50
	RBF	62.96	67.90	70.37	61.25	60.00	64.50
SIFT	*Linear*	67.90	60.49	66.67	62.50	65.00	**64.51**
	MLP	67.90	60.49	66.67	62.50	65.00	64.51
	RBF	67.90	60.49	66.67	62.50	65.00	64.51
LBP	*Linear*	64.20	60.49	64.20	67.50	66.25	**64.53**
	MLP	64.20	60.49	64.20	67.50	66.25	64.53
	RBF	64.20	60.49	64.20	67.50	66.25	64.53

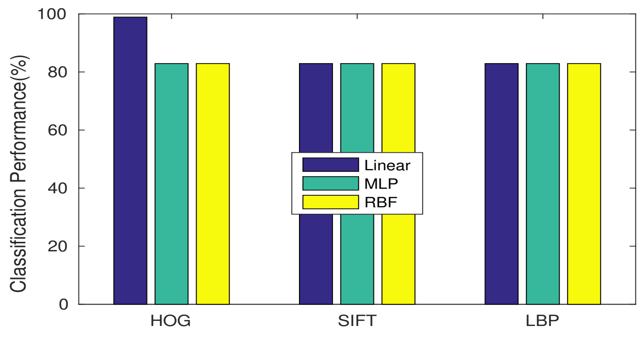

Fig. 5. Performance comparison of classification (%) of 'bike-riders' vs. 'others' in phase-I for different features and kernels.

From the results presented in Table I & Table II, it can be observed that using HOG descriptors helps in achieving best performance. Fig. 7 & Fig. 8 presents ROC curves for performance of classifiers in detection of bike-riders and detection of bike-riders with or without helmet, respectively. Fig. 7 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. 8 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 pro-

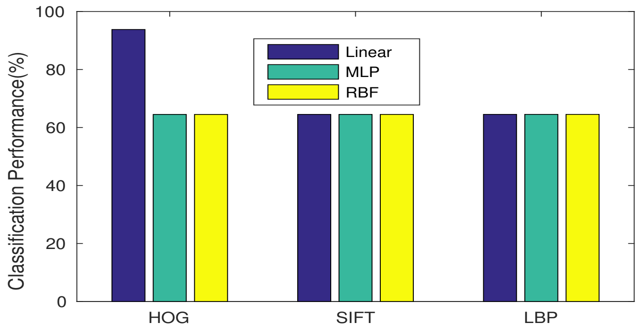

Fig. 6. Performance of phase-II classification (%) of 'bike-rider with helmet' vs 'bike-rider without helmet' for different features and kernels.

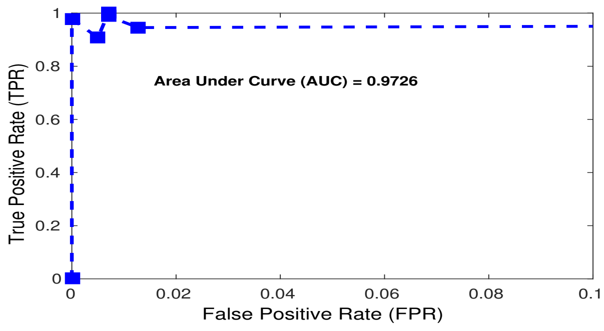

Fig. 7. ROC curve for classification of 'bike-riders' vs. 'others' in phase-I showing high area under the curve

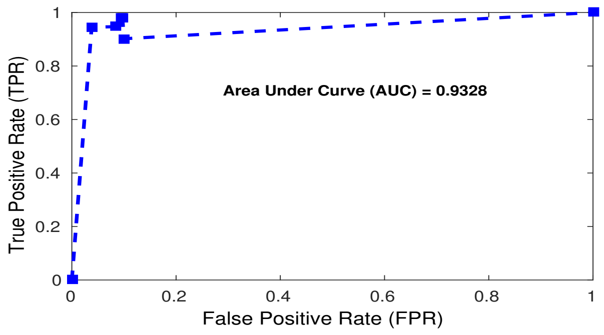

Fig. 8. ROC curve for classification of 'bike-rider with helmet' vs. 'bike-rider without helmet' in phase-II showing high area under the curve

posed 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 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 [5] [7] [16] [8].

V. Conclusion

In this paper, we propose a framework for real-time detection of traffic rule violators who ride bike without using helmet. Proposed framework will also assist the traffic police for detecting such violators in odd environmental conditions viz; hot sun, etc. Experimental results demonstrate the accuracy of 98.88% and 93.80% for detection of bike-rider and detection of violators, respectively. Average time taken to process a frame is 11.58 ms, which is suitable for real time use. Also, proposed framework automatically adapts to new scenarios if required, with slight tuning. This framework can be extended 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] 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.

[3] C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, "Motorcycle detection and tracking system with occlusion segmentation," in *Int. Workshop on Image Analysis for Multimedia Interactive Services*, Santorini, June 2007, pp. 32–32.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, Aug 2004.

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

[6] Z. Chen, T. Ellis, and S. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *Procs. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITS)*, Anchorage, AK, Sept 2012, pp. 951–956.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

[14] 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.

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

[16] 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.

[17] D. Ioannou, W. Huda, and A. F. Laine, "Circle recognition through a 2d hough transform and radius histogramming," *Image and vision computing*, vol. 17, no. 1, pp. 15–26, 1999.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.