Traffic Monitoring Using Visual Big Data Analytics in Smart Cities

Rupali Bhartiya (Assistant Professor), ChetanBhilware

Department of Computer Science and Engineering,
Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India

---***---

Abstract— **In order to identify the individuals who are breaking traffic laws, an application like video surveillance for traffic control in smart cities has to evaluate a significant volume (hours/days) of video footage. Traditional computer vision methods are unable to process the vast amount of real-time visual data that is created. As a result, there is a need for visual big data analytics, which entails processing and analysing massive amounts of visual data, such as photographs or videos, in order to discover semantic patterns that may be used for interpretation. In this research, we suggest a visual big data analytics framework for the automatic detection of bike riders without helmets in city traffic. We also talk about the difficulties of using visual big data analytics for traffic control using city-scale surveillance data and propose potential areas for future study.**

Keywords—Visual Big Data Analytics, Smart City, Traffic Con-trol, Machine Learning.

I. INTRODUCTION

In today's modern civilizations, video surveillance systems have evolved into a necessary piece of technology for keeping an eye on any type of criminal or illegal activities. Modern cities all over the world have a vast network of CCTV cameras installed by law enforcement organisations that cover all of the city's important public spaces, including its airports, train stations, and road system. From the perspective of locating criminals, spotting traffic offenders, spotting accidents, gathering evidence for investigations, etc., road traffic monitoring is crucial. Automatic decision-making systems are preferred for catching different types of traffic offenders. Two-wheelers are a rapidly growing means of transportation worldwide, but they come with a considerable risk because the human body's head is not protected. Therefore, the governments mandate the usage of helmets for people riding two-wheelers in order to protect the head portion of the body. Due to the importance of wearing a helmet, governments have made it illegal to ride a bike without one and have implemented physical measures to apprehend offenders. However, because humans are involved and their effectiveness may degrade with time, a manual system of tracking those who violate traffic laws is not a practical option [1]. For dependable and effective monitoring of these traffic rule breaches, automation of this procedure is highly desirable. Additionally, it can greatly minimise the number of people required for traffic monitoring. Many nations are implementing systems that deploy surveillance cameras in public spaces for round-the-clock security monitoring in an effort to transform urban areas into smart cities. Because it uses the existing infrastructure and requires significantly less manpower to operate, this automated solution for traffic monitoring is also cost-effective.

However, specific concerns including real-time implementation, occlusion, direction of motion, temporal changes in weather conditions, and the quality of the video feed need to be addressed in order to use such automatic methods [2]. Therefore, digesting a sizable volume of information while under time pressure is a difficult task. To reach the goal of real-time implementation, such systems require tasks like segmentation, feature extraction, classification, and tracking, which require processing a sizable amount of data quickly [1] [3]. According to [1], an effective framework for a surveillance programme should have practical qualities including real-time performance, fine tuning, and robustness to abrupt changes. We provide a visual big data analytics framework-based approach for automatically detecting bike riders without helmets in real time from traffic surveillance camera networks of a smart city in consideration of these difficulties and desired qualities.

For deploying surveillance cameras to monitor road traffic, numerous frameworks have been presented to date. A traffic monitoring system combines automatic security and surveillance on video streams recorded by surveillance cameras, object detection and tracking, behavioural analysis of traffic patterns, number plate recognition, and automated security. A cloud-based system for stream processing that can identify automobiles from captured video streams was introduced by T. Abdullah et al. [4]. Using a cloud-based graphics processor unit (GPU) cluster, this framework offers a complete solution for video stream capture, storage, and analysis. By automating the process of vehicle identification and locating noteworthy occurrences from the recorded video streams, it gives traffic control room operators more control. Only the analysis criteria and number of video streams to be analysed are specified by an operator. Then, without the need for human interaction, these video streams are automatically retrieved from the cloud storage, encoded, and processed on a Hadoop-based GPU cluster. By moving its computationally complex portions to the GPU cluster, it decreases the latency in the video analysis process. By utilising cloud computing to carry out enormous data analysis, Y. Chen et al. [5] offer an automatic licence plate recognition system that enables the detection and tracking of a target vehicle in a city with a particular licence plate number. In order to perform contextual information analysis, it develops a fully integrated system with a city-scale surveillance network, autonomous large-scale data retrieval and analysis, and combination of pattern recognition. A hybrid cloud concept for a video surveillance system with mixed-sensitivity video streams has been put out by C. Zhang et al. [6].

By preserving sensitive data in the private cloud and shifting computing to the public cloud to reduce seasonal burden, the hybrid cloud helps to address security concerns. A middle-ware that successfully schedules the work and smoothly connects the private and public clouds is utilised to improve usability and lower costs. In this hybrid cloud, a stream processing model optimises the total cost to be paid on the public cloud while taking into account resource limitations, security concerns, and Quality-of-Service requirements (QoS).

In this research, we present a framework for the automatic real-time detection of bike riders without a helmet from video feeds from the city's surveillance network. We also cover the visual big data analytics framework and its underlying techniques and application. The first phase of the proposed approach uses object segmentation and background subtraction techniques to identify bike riders in surveillance videos. In the second phase, it identifies the cyclist's head and extracts the necessary features to determine whether or not the rider is wearing a helmet. In order to decrease false alerts and increase the dependability of the suggested approach, a consolidation approach is also offered for alarm production. Three commonly used feature representations—the histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and local binary patterns (LBP) for classification—have been compared for performance in order to assess our technique. According to the experimental findings, 93.80% of the real-world surveillance data may have been detected. 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.

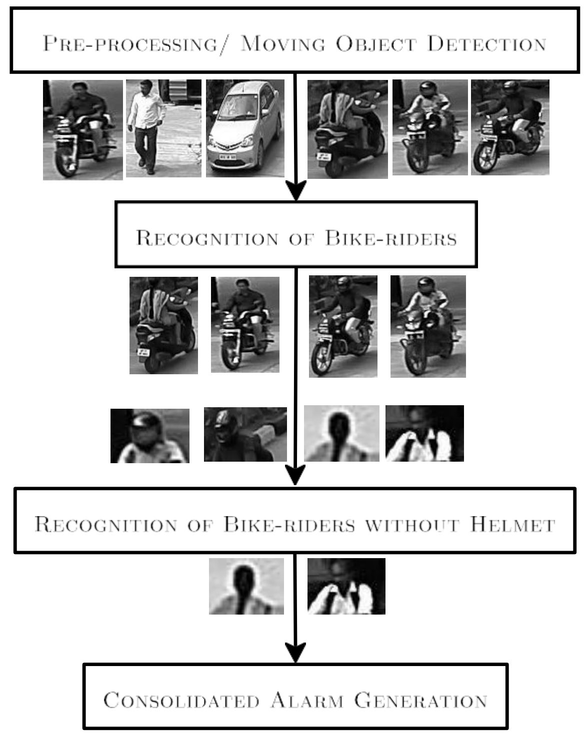


Fig. 1. Block diagram illustrates the suggested method for automatically identifying bicycle riders without helmets.

The rest of this essay is structured as follows: The suggested method for automatically identifying bike riders without helmets is presented in Section II. Section III presents the suggested framework for visual big data. The outcomes of experiments are covered in Section IV. Section V contains the conclusion.

II. PROPOSED APPROACH FOR AUTOMATIC DETECTION OF BIKE-RIDERS WITHOUT HELMET

The suggested method for detecting bike riders without helmets in real-time, which consists of two phases, is presented in this section. The first stage involves identifying every bike rider in the frame of the video, and the second involves finding the rider's head and determining whether or not the rider is wearing a helmet. We combine the findings from successive frames for final alert creation to lessen the generation of false alarms. Using sample frames, the block diagram in Fig. 1 illustrates the many processes of the proposed framework, including background subtraction, feature extraction, and object classification.

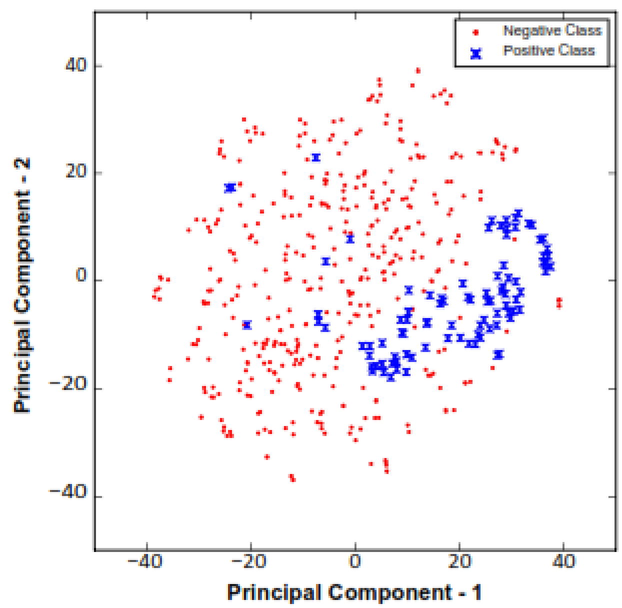
A. Pre-processing/ Moving Object Detection

We use background subtraction on grayscale frames to discriminate between moving and stationary items. The background subtraction method in [7] is used to distinguish between moving items, such bicycles, people, and cars, and stationary ones, like trees, roads, and buildings. However, working with data from a single fixed camera presents several difficulties.Environment conditions like illumination variance over the day, make it difficult to recover and update background from continuous stream of frames. A single Gaussian cannot accurately explain all fluctuations in complicated and changeable environments [8]. This necessitates the usage of a variable number of Gaussian models, one for each pixel. Here, K, the empirically determined range of the number of Gaussian components for each pixel, is maintained between 3 and 5. Due to the presence of heavily obscured objects and blended shadows, some errors may still happen. The online clustering method suggested in [7] is used to approximate the background model. Foreground objects are produced by subtracting background mask from the currently framed image. To minimise noise, a Gaussian filter is applied to the foreground mask before being thresholded using clustering [9]. Close morphological treatments are employed to further process the foreground mask in order to improve item differentiation. This processed frame is then divided into sections based on the boundaries of the objects. Only moving objects are retrieved using the background subtraction method, while static objects and other useless information are ignored. There may still be a lot of moving things that are not of interest to us, like people, automobiles, etc. Based on their area, these things are filtered. This has the goal of just taking into account items that are more likely to be used by bike riders. It aids in lowering the difficulty of subsequent steps.

B. Recognition of Bike-riders

This stage entails finding bicycle riders in a frame. It makes use of objects to distinguish between potential bike riders and other people based on their visual characteristics.

The classification of objects involves an appropriate representation of visual characteristics. Local binary patterns (LBP) [12], scale invariant feature transform (SIFT) [11], and histogram of oriented gradients (HOG) [10] have all been shown in the literature to be effective for object detection. We examine three features—HOG, SIFT, and LBP—for this aim. HOG descriptors, which have been shown to be particularly effective at detecting objects. Through gradients, these descriptors capture regional shapes. SIFT aims to capture important details in the picture. Feature vectors are extracted for each key-point. These descriptors' robustness under many circumstances is a result of their scale, rotation, and illumination invariance. To develop a dictionary, we employed the bag-of-words (BoW) technique. Feature vectors are then produced by mapping SIFT descriptors to dictionary words. The similarity between photos is assessed using these feature vectors. LBP records the texture data in the frame. By thresholding the pixels in the circular neighbourhood, a binary number is allocated to each pixel, and the frequency histogram of these numbers serves as the feature vector. Using t-SNE [13], Fig. 2 depicts the phase-I classification patterns in 2-D space. The HOG feature vector distribution reveals that, with very few exceptions, the two classes—"bike-riders" (positive class, shown in blue crosses) and "others" (negative class, shown in red dots]—fall in nearly separate geographic areas. This demonstrates how effectively the feature vectors describe the activity and how they contain discriminative information, which raises the prospect of accurate categorization.



Principal Component – 1

Fig. 2. HOG feature vector visualisation for the t-SNE classification of "bike-rider vs. others" [13]. The bike-rider class is represented by a blue cross, whereas the non-bike-rider class is represented by a red dot. [Best viewed in color]

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 12 | Dec 2022 www.irjet.net p-ISSN: 2395-0072

The next step after feature extraction is to classify the items as "bike-riders" or "other" objects. As a result, a binary classifier is needed. Any binary classifier can be employed in this situation, however we opt for SVM because of its robustness in classification performance even when trained with fewer 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

The following stage is to ascertain whether or not the bike riders are wearing helmets after the bikes have been spotted in the previous step. Due to the following factors, standard face detection algorithms would not be enough for this phase: I It is quite difficult to catch facial details like the eyes, nose, and mouth when the resolution is low. ii) The bike's movement angle can be acute. Face may not be discernible at all in such circumstances. In order to assess whether the bike rider is wearing a helmet or not, the suggested framework first detects the area surrounding the rider's head. The suggested framework makes use of the assumption that the bike rider's top sections are likely to be where the helmet should be placed in order to find the rider's head. We just take into account the top fourth of the object in this. To establish whether a bike rider is wearing a helmet or not, a certain area around their head is identified. HOG, SIFT, and LBP—features that were also utilised in phase I—are used to do this. Using t-SNE [13], Fig. 3 depicts the patterns for phase-II in 2-D. The two classes, "non-helmet" (Positive class indicated in blue cross) and "helmet" (Negative class shown in red dot), fall in overlapping regions, which demonstrates the complexity of representation, according to the distribution of the HOG feature vectors. However, Table I demonstrates that significant discriminative information is included in the produced feature vectors to achieve high classification accuracy.

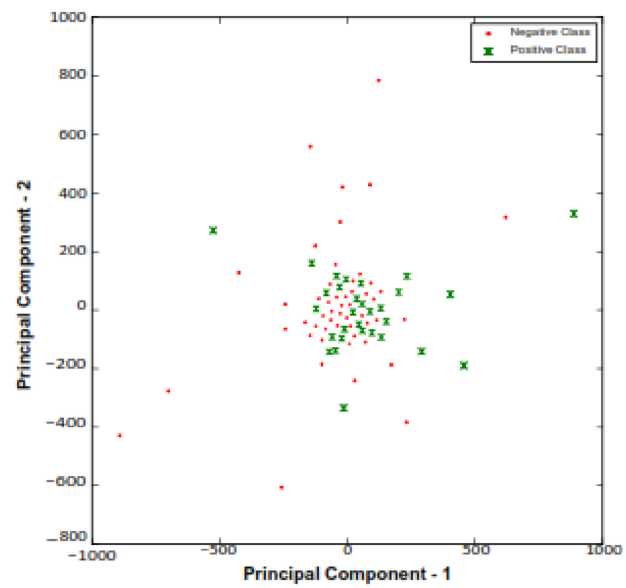


Fig. 3. Visualisation of HOG feature vectors for t-SNE-based categorization of "helmet vs. non-helmet" [13]. Green cross denotes a non-helmet class, whereas Red dot suggests a helmet class. [Best viewed in color].

The technique must ascertain whether the rider is disobeying the law, for as by not wearing a helmet. For this, we take into account two classes: Rider not wearing a helmet (Positive Result) and Rider wearing a helmet (Positive Result) (Negative Result). When classifying data, the support vector machine (SVM) is employed with the features that were extracted in the preceding stage.To analyze the classification results and identify the best solution, different combination of features and kernels are used. Results together with analysis is combined in *Result* section.

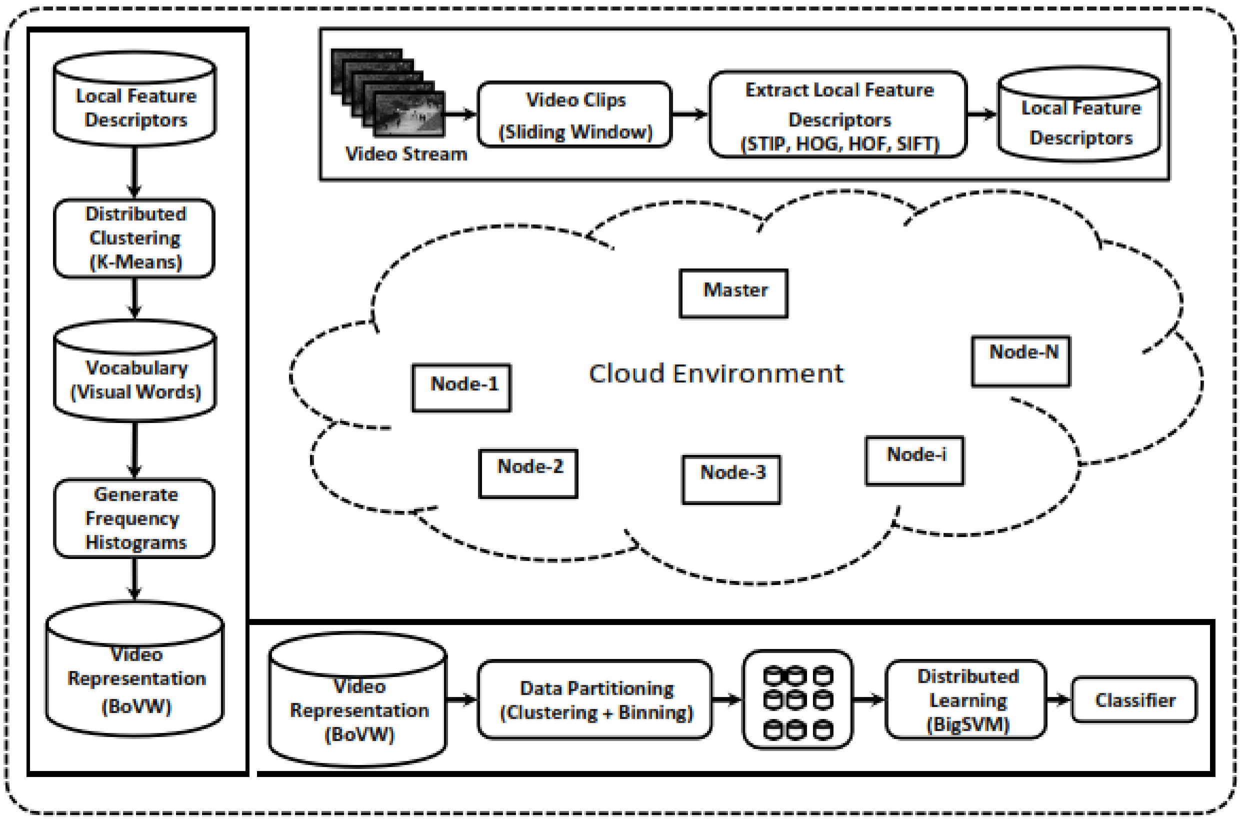


Fig. 4. Block diagram of visual big data framework over the cloud for traffic monitoring with the help of large surveillance camera network of a smart city. Illustration of distributed processing of feature representation using bag-of-words (BoW) and classification using distributed support vector machine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

D. Consolidated Alarm Generation

We collect local results, such as whether a bike rider is wearing a helmet or not, in a frame, from earlier phases. The link between continuous frames has, however, thus far been ignored. We therefore combine local results in order to lower false alarm rates.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 resources on-demand at reduced cost. This framework uses a 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

We suggested a framework dubbed the visual big data framework over the cloud to implement the proposed method on a city-scale surveillance camera data network. Visual computing, big data analytics, and cloud computing make up the full framework for visual big data analytics over the cloud. Finding semantic patterns that can be used for interpretation requires processing and interpreting visual data, such as pictures or videos.

In order to find odd patterns, a vast volume (hours/days) of video footage needs to be analysed for large-scale video surveillance applications like traffic monitoring in smart cities. Due to the significant number of data that is generated at a high rate, real-time analysis of such massive visual data sets is a difficult task. As a result, the total issue is now a visual big data issue for which current visual computing solutions fall short of the required performance. Also, the up- front infrastructure investment is costly, so we are leveraging the benefits of cloud computing in order to get computing hybrid cloud architecture where continuously used resources are provided by private cloud. It makes use of the advantages of public clouds for tasks that require lots of computing. For instance, in a surveillance system, cameras and any attached computing resources are continuously employed to check for any hostile activity on a trained model. We can use a private cloud, a public cloud, or our own on-site infrastructure to meet this demand. It makes use of public cloud resources for the long-term storage of data and computationally expensive training procedure. Training is a cost-effective option because it is a temporary demand. We may put up a suitable cluster in the cloud using MapReduce to distribute the processing of visual data as needed. The block diagram of a visual big data framework for cloud-based traffic monitoring employing a sizable network of surveillance cameras in a smart city is shown in Fig. 4. . The two primary tasks in visual computing applications are feature extraction and categorization. Both of these tasks are computationally challenging, and the difficulty increases when dealing with big visual data sets. Here, cloud computing is being used to meet all of our short-term computer resource needs. The performance of pattern recognition tasks improves with training over big datasets. The suggested method includes computationally challenging tasks like support vector machine (SVM) classifier training and k-means clustering for vocabulary synthesis in bag-of-words (BoW) feature representation. To address this, we bring forth a distributed framework (Fig. 4), in which computations for BoW and SVM are spread throughout a distributed environment, such as a cluster or cloud. Clustering is used in the bag-of-words approach, however with larger volumes of data, clustering is more difficult. For a variety of clustering algorithms, there are several distributed implementations available; however, we adopt PKMeans, a parallel k-means clustering algorithm published by W. Zhao et al. [14] for MapReduce. PKMeans has three different operations: map, combiner, and reduce. In order to reduce communication, the map function places samples in the nearest centre, the combiner function adds up the points each cluster received from a single map function, and the reduce function creates new centres from arrays of partial sums produced by each map function. We employ the divide and conquer (DCSVM) method for SVM, which was developed by Hsieh et al. [15]. Using k-means clustering, the training data are divided into smaller divisions, and local SVM models are independently trained for each smaller partition using the LIBSVM package. Next level SVM receives as input the support vectors from local SVMs. Finally, local SVM models are combined to create a global SVM model. This cuts down on total time, especially when working with huge amounts of 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

Since there is no publicly accessible data set, we gathered our own information from the surveillance system at the SVVV campus in Indore. A total of two hours' worth of surveillance data is gathered at a frame rate of 30. The samples from the gathered dataset are shown in Fig. 5. The first hour of the movie is used to train the model, while the second hour is utilised for testing. In the training video, there are 40 people, 13 cars, and 42 bikes. In contrast, the testing video features 66 people, 25 cars, and 63 bikes.

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

Fig. 5. Sample frames from dataset

We ran studies using 5-fold cross validation to validate the performance of each format and model combination. According to the experimental results in Table I, classification utilising SIFT and LBP features performs roughly equally well on average when classifying bikes versus non-bikes. Additionally, the performance of HOG classification utilising MLP and RBF kernels is comparable to that of SIFT and LBP. Because the feature vector for this representation is sparse in nature and suitable for a linear kernel, HOG with a linear kernel outperforms all other combinations. We can see that the average performance of categorization using SIFT and LBP for head vs. helmet is nearly identical Additionally, HOG classification using MLP and RBF kernels performs similarly to SIFT and LBP in terms of performance. HOG with a linear kernel, however, outperforms all other combinations.

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.

Figures 6 and 7 show ROC curves for classifiers' performance in detecting bike riders and detecting bike riders wearing or not wearing helmets, respectively. The accuracy is above 95% with a low false alarm rate of less than 1% and an area under the curve (AUC) of 0.9726, as shown in Fig. 6. AUC is 0.9328, and Fig. 7 clearly demonstrates that accuracy is above 90% with a low false alarm rate of less than 1%.

A. Computational Complexity

To test the performance, a surveillance video of around one hour at 30 fps i.e. 107500 frames was used. In 1245.52 seconds, or 11.58 milliseconds per frame, the suggested framework processed all of the data. 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 [16] [17] [18] [19].

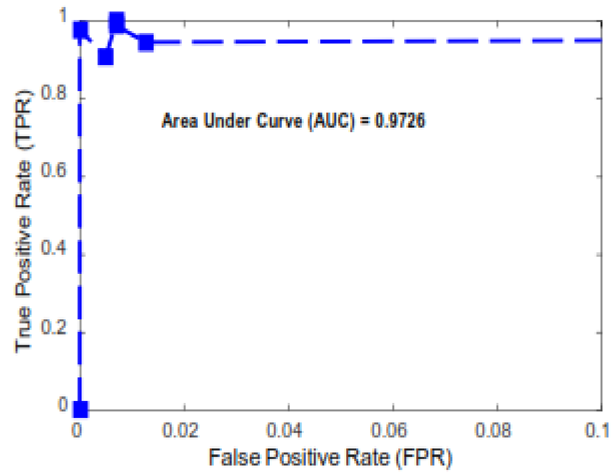


Fig. 6. ROC curve for categorization of 'bike-riders' vs. 'others' in phase-I depicting high area under the curve

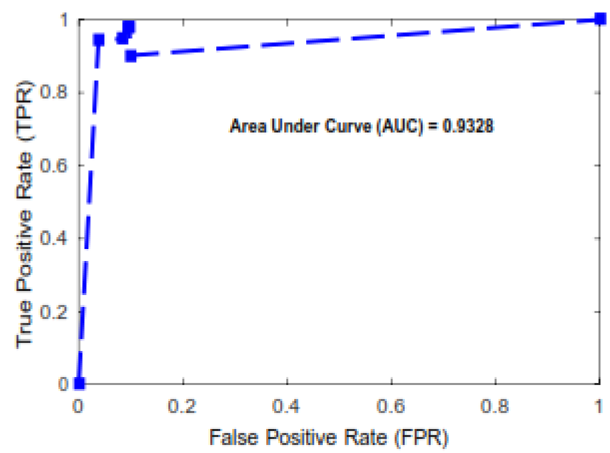


Fig. 7. ROC curve for categorization of 'bike-rider with helmet' vs. 'bike-riderwithout helmet' in phase-II depicting high area under the curve

V. CONCLUSION

In this research, we offer a visual big data analytics-based framework for real-time detection of traffic law violators who ride bikes without using a helmet in a network of city-scale surveillance cameras. The suggested framework would also help the traffic police catch such offenders in unusual weather, such as hot sun, etc. The high classification performance, which is 98.88% for the recognition of bike riders and 93.80% for the detection of offenders, is shown by the experimental findings. 11 ms is the average processing time per frame, which is appropriate for real-time use. Additionally, with a little tweaking, the suggested architecture automatically adapts to new conditions. This framework can be enhanced to find and report violators' licence plates as well as other types of rule violations.

REFERENCES

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

[2] "Real-time on-road vehicle and motorbike recognition using a single camera," in Proceedings of the IEEE International Conference on Industrial Technology (ICIT), 10-13 February 2009, pp. 1-6.

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

[4] 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.

[5] "Processing of Mixed-Sensitivity Video Surveillance Streams on Hybrid Clouds," in IEEE International Conference on Cloud Computing, 2014, pp. 9–16.

[6] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," Proceedings of the International Conference on Pattern Recognition (ICPR), vol. 2, August 23–26, 2004, pp. 28–31.

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

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

[9] "Histograms of directed gradients for person detection," Procs. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), June 2005, pp. 886–893.

[10] D. Unique visual features from scale-invariant keypoints, G. Lowe

[11] Int. computer vision journal, 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] A Divide-and-Conquer Solver for Kernel Support Vector Machines, C.-J. Hsieh, S. Si, and I. Dhillon, Proc. of International Conference on Machine Learning (ICML), Beijing, 21-26 Jun 2014, pp. 566-574.

[16] In September 2012, Chiverton published "Helmet presence categorization with motorbike recognition and tracking" in Intelligent Transport Systems (IET), vol. 6, no. 3..

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

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

[19] Helmet detection on motorcycle riders using visual descriptors and classifiers, Procs. of the Graphics, Patterns and Images (SIBGRAPI), August 2014, pp. 141–148. R. Rodrigues Veloso e Silva, K. Teixeira Aires, and R. De Melo Souza Veras..