Vision-Based Cyclist Travel Lane and Helmet Detection

Jyoti Madake[1], Shripad Bhatlawande[1], and Madhusha Shete[1]

[1]Dept of E&TC Engineering, VIT Pune, India, 411037
{jyoti.madake, shripad.bhatlawande, madhusha.shete19}@vit.edu

Abstract. Cycling is an integral part of daily life for many people. This project presents a vision-based cyclist travel lane and helmet detection system. This system can serve as surveillance to detect whether the cyclist is traveling in a devoted lane and wearing a helmet for safety measures. The model involves the application of scale-invariant feature extraction (SIFT) algorithm for feature description. The detection method is based on six machine-learning classification algorithms. The classifiers are evaluated based on testing accuracy, and the best classifier is selected for the final model creation. The random forest classifier provided highest training accuracy of 99% and testing accuracy of 85.44% for cyclist travel lane detection. The same classifier provided the highest training accuracy of 99.53% and testing accuracy of 87.83% for cyclists' helmet detection. In future, this system can also serve as a small part of autonomous driver assistance systems by detecting the right lane.

Keywords: Cyclist travel lane detection, helmet detection, SIFT, PCA, machine learning.

1 Introduction

Bicycles have been a major part of transportation sources for ages. In the era of fancy cars and speedy motorcycles, many people still use bicycles for transportation and other purposes. Riding a bicycle is an agreeable as well as essential part of day-to-day life for individuals of any age group and capacity and is advantageous in many ways. Hence, cyclist travel lanes are being made to give separate space to cyclists from cars and motorcycles on heavy-traffic roads. These lanes are also referred to as protected cyclist lanes. This evolution toward cyclist travel lanes started when Seattle came up with the Bicycle Master Plan (BMP) in 2013 [16]. This drive aspired to encourage and let people to ride a bicycle. Since then many countries are making cyclist travel lanes a part of their road structure. There are a few cities in India too who are having these cyclist travel lanes. These lanes not only help to control traffic by parting cyclists in the other part of the road but also decrease the number of accidents that are caused due to the crashing of high-speed vehicles and comparatively low-speed cycles. But using cyclist travel lanes does not ensure the complete safety of cyclists. Cyclists must wear helmets to ensure their safety. According to government reports [18], it has been noted that many bicycle and bike users do not use helmets while riding. It is also necessary to check the presence of a helmet making it mandatory. Multiple such systems are being implemented by making use of disciplines such as machine learning algorithms.

The proposed system promotes the presence of cyclist travel lanes, increasing the number of these lanes in every city of India and making people aware of it. In vision-based cyclist travel lane and helmet detection, the system detects whether the cyclist is using the cyclist travel lane or normal road for cycling along with the detection of the helmet. The objective is to detect the presence of a cyclist travel lane with the help of parameters such as cyclist lane signs, cyclist travel lane structure, and appearance. It is mandatory to use a helmet even if the cyclist is using a protected lane for cycling. This project also detects the presence of helmets on cyclists' heads. Two classifiers are implemented to achieve the mentioned objectives. One of them classifies helmet and non-helmet inputs while the other classifies cyclist lanes and normal roads.

2 Literature Survey

This project aims to detect if a cyclist is traveling on a cyclist travel lane and wearing a helmet. The literature review presents an overview of the previously published works on the detection of cyclist travel lanes and helmets using OpenCV and other different algorithms. A new method for cyclist recognition in real time was presented in [1]. A cascaded detector was used. A histogram of gradients (HOG) features and an SVM classifier was implemented. The training was carried out on an open-source KITTI dataset containing images of bikes, cyclists, and pedestrians. The SVM classifier was implemented for training. The model was able to detect the cyclist in 0.32 seconds on video input. A UB-MPR dependent recognition proposition technique presented in [2] was a Fast R-CNN-based algorithm. The proposed strategy could recognize people on foot and cyclists simultaneously and separate them. The method proposed in [3] presented a helmet detection system wherein a video was captured through the camera and the frames were extracted using a Caffe model. Object detection was performed on the live video. The SVM-based model had an accuracy of 86% and a validation accuracy of 76%. A computer vision-based system was proposed in [4]. The segmentation was done for moving objects followed by classification and detection. In the first step adaptive moving gaussian was applied to segment moving objects from the background. The techniques implemented included the selection of the region of interest (ROI), circular Hough transforms (CHT), HOG, and Local binary pattern (LBP). The model could result in 91.37% accuracy.

A real-time automated system was proposed in [5]. The system could detect motorcycles with an accuracy of 99.5% with a sequential minimum optimization (SMO) classifier. For helmet detection, experimental results showed 96.98% accuracy with the Logistic classifier and an accuracy of 99.62% with the Custom convolution neural network (CNN) classifier. The study in [6] developed a safety as well as a security-based system to avoid outside bike riding accidents. The ultimate aim was to detect helmets. The data was collected from outdoor road traffic in video format. This model proposed a multiclass kernel-based SVM classification for determining humans with or without a helmet. Data preprocessing algorithms such as gaussian mixture modeling (GMM) background modeling, noise removal, thresholding, and morphological filtering were implemented. The classifying model could yield 96.67% accuracy. The proposed method in [7] presented methods such as aggregated channel features, deformable part model, region-based CNN, and stereo proposal-based fast R-CNN for cyclist detection. The dataset was collected from a camera capture of 22161 images. Testing was done over three difficulty levels namely, moderate and hard wherein hard gave good accuracy for cyclists with less than 80% occlusion, and moderate managed for less than 40% occlusion, and needs it to be fully visible.

The implementation in [8] presented a helmet detection model based on YOLOv5. The model was trained on 6045 images which were labeled as 'Alarm' if the helmet is not there and 'Helmet' if the helmet is there. The method trained and evaluated YOLOv5 (s, m, l, x) models with different depths and widths and compared them. The average detection speed of YOLOv5s reached 110 FPS. The method presented in [9] was a helmet detection model using YOLO V3. This model also presented the improved version of YOLO V3 full regression deep neural network architecture. The improved version could detect helmets in all possible conditions with low image resolution. The data was collected from the internet as well as camera capture and trained over 1500 epochs. The helmet detection approach introduced in [10] provided a hybrid descriptor for feature extraction based on LBP, HOG, and the Hough transform descriptor. The methodology included background detection followed by moving object segmentation. Future extraction was carried out using LBP. The SVM classifier was implemented. The model had a training accuracy of 97.67% and a testing accuracy of 94.23%. Cyclist path prediction carried out in [11] implemented RNN and Gaussian distribution. The dataset was composed of 51 videos of cyclists. The decision of cyclists to turn left or move ahead was made using this model. The model had an average error of 33cm while predicting the next position of the cyclist. Research in [12] promoted the use of local road topology. The implementation in [13] presented a stud that compared two Kalman Filter (KF) techniques for predicting cyclists' trajectories at a junction. Least-squares approximation method was used to predict the cyclist’s future position. A multilayer perceptron ANN model was implemented. The model could give the detection results in 1 to 2.5 seconds on video data.

The study proposed in [14] had two models for cyclist trajectory predictions. Motion Modelling Approach and data-driven Approach based on stacked long short-term memory (LSTM) were implemented. The LSTM and RNN-based model could give detection results in 3 seconds on video data. Stochastic gradient descent (SGD) optimization algorithm was implemented in [15] for object detection. The research concluded that integrating SGD into the optimization process results in improved detection speed. The training was done on a data set containing 15000 images. The model was trained using algorithms such as libSVM, SVMLight, and SGD. The study in [16] showed that the number of bicyclists who use helmets while riding was very low. The survey was carried out to know the number of male and female cyclists wearing helmets. The outcomes of the study in [17] stated that cyclists had a lack of attention while driving. A framework for the detection of cyclists as well as pedestrians was proposed in [19]. A fast R-CNN algorithm was implemented. The model had 89% test accuracy. The LSTM-based algorithm was proposed in [20] for the detection of cyclist paths. The LSTM model was able to detect the required output in 15.6 milliseconds. The object detection algorithm was proposed in [21]. The HOG feature extraction technique was followed by PCA. The detector model was trained using SVM. SGD algorithm was implemented for optimization.

3 Methodology

A simplified method for the detection of cyclist travel lanes and helmets is proposed in this paper. This vision-based system is mainly divided into two different detectors. The first detector detects whether the cyclist in the input source is traveling on the dedicated cyclist travel lane. The second detector detects whether the cyclist in the previous input source has a helmet on his/her head. These two models are clubbed in order to make a system for the detection of cyclist travel lanes and helmets.

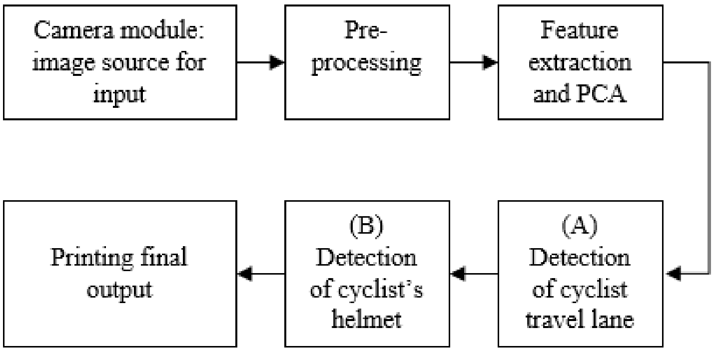

Fig. 1. Block diagram for cyclist travel lane and helmet detection system

Figure 1 presents the block diagram of the proposed system. An image of a cyclist is taken as an input source to the system. The inputted image is pre-processed for further use using methods such as resizing and gray scaling. The pre-processed image is passed to the SIFT feature detector and descriptor for the feature extraction purpose. The extracted features are clustered using the K-Means clustering algorithm. The principal component analysis is carried out on the inputted data to reduce the dimensions to insure superior computational complexity. The generated data is sent to the two different models. The first model aims to detect the features relevant to the cyclist's travel lane using a random forest classifier and the other model aims to determine the presence of a helmet on the cyclist's head using an SVM classifier. The random forest and SVM classifiers are selected by comparing all implemented classifiers based on their test accuracy as mentioned below in the paper.

3.1 Dataset and Preprocessing

The vision-based cyclist travel and helmet detection system is trained and tested on a custom dataset containing 17,000 images. These images belong to four distinct classes: cyclist lane, non-cyclist lane, cyclist with a helmet, and cyclist without a helmet. The images for the dataset compilation were collected from both the internet and camera capture.

Table 1. Dataset details

Sr. No.	Class	No. of images
1	Cyclist lane	1500
2	Non-cyclist lane	1500
3	Cyclist with helmet	7000
4	Cyclist without helmet	7000
	Total	17000

Table 1 illustrates the classes and the total number of images in each class. The images in the cyclist lane class included the images of cyclist travel lanes in India as well as abroad and the signs used to indicate the cyclist travel lanes. The non-cyclist lane class was composed of images showcasing normal road traffic with different motorbikes, cars, and other vehicles. The cyclist with helmet class contained images of cyclists wearing helmets along with some helmet or hard hat images. These images encompassed cyclists and helmets from all possible angles. The cyclist without helmet class was composed of images of cyclists who were not wearing helmets and some bare-head images of people doing various activities on the road. In the real-time surveillance system, the cyclist's travel lane and helmet scanning are mostly done by the cameras mounted on the poles on the side of the roads.

All images were transformed to grayscale while pre-processing. The main intention behind the gray scaling of images was to simplify the algorithm. The gray images are easier for the feature description operations in comparison with colored images and they require less processing time. The images were resized during pre-processing. Images from the cyclist lane and non-cyclist lane were resized to 200 X 200 pixels. Cyclists with helmets and cyclists without helmet classes were having images with dimensions equal to 150 X 150 pixels after pre-processing. The pre-processed images were fed to the feature descriptor for the extraction of features.

A B

C D

Fig. 2. Sample dataset images

Figure 2 represents the samples of the images in the proposed dataset. Images A, B, C, and D belong to the cyclist lane, non-cyclist lane, cyclist with helmet, and cyclist without helmet classes respectively.

3.2 Feature Extraction and Dimensionality Reduction

The principal goal of any feature descriptor algorithm is to take an input image and provide the output stating the pixel coordinates i.e. location of important portions in the image. The proposed method uses a SIFT feature descriptor for the extraction of features. The SIFT feature consists of some key points in the image, each with an orientation and a descriptor for the area surrounding the key points. The SIFT method refines the location and scale of feature points or key points after determining their approximate location and scale. The algorithm then derives the descriptors for each key point by evaluating the orientation(s) for each key point. There are a couple of reasons for selecting SIFT for feature extraction purposes. The first reason is that the size or orientation of the image does not affect SIFT features. This is advantageous for applications where any raw image captured from an unusual angle is fed to the descriptor.

The k-Means algorithm is used for the clustering of the data that is extracted using SIFT. K-Means is an unsupervised learning-based algorithm used to address clustering problems. It follows a centroid-based strategy. The clusters formed using K-Means have unique centroids. This technique is used to minimize the sum of distances between key points and the clusters that they belonged to. K-Means is implemented to complete two basic tasks. In the first task, an iterative technique is implemented to get the best value for K center points. In the second task, each data point was assigned to the K-valued center that was closest to it. The data points that are close to specific k-center forms the cluster. The value of K was decided using a method named elbow. This method is used to find the optimal number of clusters. Calculations for Within Cluster Sum of Squares (WCSS) are carried out in this method using equation 1. WCSS determines the total variations within a cluster.

WCSS = ∑da in Cluster 1 distance (Di C1) 2+ ∑di in Cluster 2 distance (Di C2) 2 + ….. + ∑di in Cluster n distance (Di Cn) 2 (1)

The d mentioned in above equation 1 is the data point. C is the centroid. The WCSS values are calculated based on the above-mentioned formula. The inbuilt library based on this formula was used in the code. The elbow method executed K-Means clustering on the dataset for different values of K ranging from 1 to 15. The WCSS value of each K value was compared to select the best K value. The value of K was taken as 5 for the helmet detection model and 9 for the cyclist lane detection model.

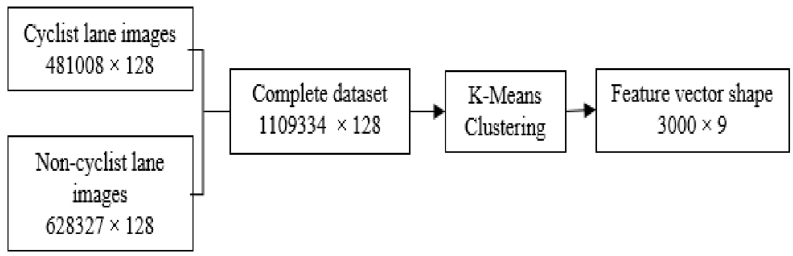

Fig. 3. K-Means block diagram for cyclist travel lane detector

Figure 3 represents the workflow of the K-Means algorithm for cyclist travel lane detection. The clustering algorithm was carried out for 9 clusters separately on the cyclist lane and non-cyclist lane images and then combined.

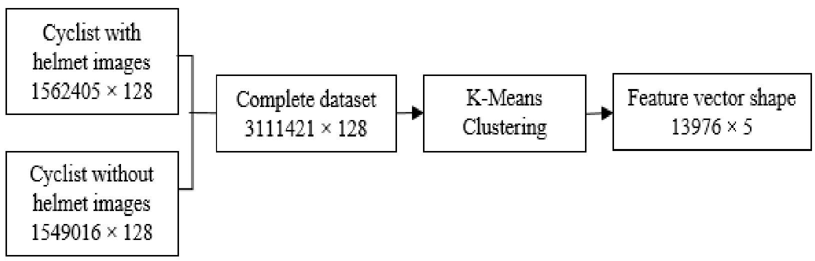

Fig. 4. K-Means block diagram for cyclist's helmet detector

Figure 4 represents the workflow of the K-Means algorithm for cyclist helmet detection. The clustering algorithm was carried out for 5 clusters separately on the cyclist with helmet and cyclist without helmet images and then combined.

The clustered data was further processed for standardization as building algorithms with big datasets involves scaling of features, which is considered an important step for getting better accuracy. The data outputted by SIFT and K-Means clustering algorithm is scaled using a standard scalar function. The standardization technique converts the data to a scale ranging from -1 to 1 by removing the mean and scaling the unit variance. The standard score of a data point x can be calculated by equation 2.

$$z = (x - u) / s \quad (2)$$

The parameters u and s mentioned in the above formula are the mean and standard deviation of the training samples. The principal component analysis was implemented on the scaled data for dimensionality reduction. The dimensionality of huge datasets is reduced by transforming a large set of variables into a smaller set by using PCA. The method preserves the majority of the information in the larger set. The PCA involves the computation of the covariance matrix of standardized data. A covariance matrix depicts the covariance among each pair of data elements of a random vector in probability theory and statistics. The Eigenvectors and Eigenvalues of the covariance matrix are calculated for determining the principal components of data. The feature vector is a matrix that has columns containing eigenvectors of the components that we decide to keep. The final dataset is achieved after PCA transformation. The final dataset can be described as:

Final dataset = (Feature vector) T * (Standardized original dataset) T

Algorithm 1: Preprocessing and Dimensionality Reduction

Input: Directory of input images
Output: Dataframe of training data
Reduce (D) {

1. N = Dataframe
2. **for all** I in D **begin**
    a. Ig = grayscale(I)
    b. N = N + SIFT(Ig)
    c. k= Kmeans(clusters = n)
3. **end for**
4. k = k.fit(N)
5. Z = Dataframe
6. **for all** I in D **begin**
    a. Ig = grayscale(I)
    b. q = SIFT(Ig)
    c. Z = Z + k.predict(q).bins()
7. **end for**
8. S = standardize(Z)
9. P = PCA(components = p)
10. P.transform(S)
11. **return S**

}

Algorithm 1 defines the workflow of the feature extraction and dimensionality reduction phase. This algorithm is used for extracting features of 4 different classes belonging to 2 models. The specifications of cluster (K) values and the number of PCA components are different for the 2 models. The helmet detection model has a K value equal to 5 and all 5 PCA components were taken into consideration while creating the final dataset for the helmet detection model. The cyclist lane detection model has K values equal to 9 but only 6 PCA components were considered while creating the final dataset for the cyclist lane detection model.

3.3 Classification and Analysis

The classification was done using multiple algorithms for both cyclist lane detection and helmet detection. The algorithms such as decision tree, random forest, KNN, SVM, GNB, and AdaBoost were implemented on training data.

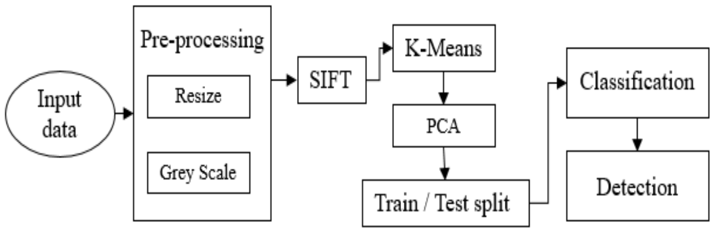

Fig. 5. Flowchart for training of model

Figure 5 showcases the flowchart of the proposed methodology for training the dataset. The input data which is data taken from the output of PCA is divided into train data and tests before passing it to the classifier. The training and testing images were taken such as there were 80% training images and 20% testing images. The classification result of all the algorithms is compared based on test accuracy and the best accuracy giving models for both the cyclist lane detector and cyclist's helmet detector are taken into consideration while making a system. The algorithm for the proposed system is mentioned below.

Algorithm 2: Vision based cyclist travel lane and helmet detection system.

Input: Input image source
Output: Detection of cyclist travel lane and helmet
Detect () {

1. img = input image
2. g = grayscale(img)
3. g = resize(g, resize())
4. N = N + SIFT(g)
5. K = Kmeans(clusters = n)
6. k = k.fit(N)
7. Z = Dataframe
8. Ig = grayscale(img)
9. q = SIFT(Ig)
10. Z = Z + k.predict(q)
11. S = standardize(Z)
12. P = PCA(components = p)
13. Test = P.transform(S)
14. Result1=RF_cyclist_model.predict(Test)
15. Result2=SVM_helmet_model.predict(Test)
16. Print(Result1)
17. Print(Result2)

}

Algorithm 2 showcases the workflow of a vision-based cyclist travel lane and helmet detection system. The input source image is given to the detector function. The required pre-processing and feature extraction strategy that includes gray scale transformation, resize operation, and SIFT-based feature description is implemented on the input image. The data frame of the input image is created by applying K-Means clustering with cluster numbers equal to the cluster value used while training the model. After feature extraction, PCA is executed on the extracted features. The resultant data frame is passed to the trained models for the prediction of respective classes. Firstly, the cyclist's travel lane is detected using a random forest model and then the helmet is detected using the SVM model. The desired output for vision-based cyclist travel lane and helmet detection is examined using these two above-mentioned predictions.

The evaluation of implemented classifiers was carried out based on precision, recall, and F1 score. The percentage of accurately predicted positive values (TP) to the total number of positive values (FP+TP) is defined as precision as given in equation 3. In equation 4, Recall is defined as the proportion of accurately predicted positive values to the entire absolute class (FN+TP). The F1 score is computed by taking the weighted mean of the accuracy and recall values in equation 5.

$$\text{Precision Score} = TP / (FP + TP) \quad (3)$$

$$\text{Recall Score} = TP / (FN + TP) \quad (4)$$

$$\text{F1 Score} = 2 * \text{Precision Score} * \text{Recall Score} / (\text{Precision Score} + \text{Recall Score}/) \quad (5)$$

4 Performance evaluation

The above-mentioned classifiers were tested on the data belonging to the 4 distinct classes and 2 different models. The performance evaluation of the cyclist travel lane detector model and cyclist's helmet detector model were carried out separately. The below-mentioned two sections represent the evaluation of each classifier belonging to the two distinct detectors.

4.1 Cyclist travel lane detector

The cyclist travel lane detector consists of a classification of images based on two classes: cyclist lane and non-cyclist lane. The cyclist travel lane class is labeled as '0' and the non-cyclist lane class is labeled as '1'. The classifiers such as decision tree, random forest, and KNN were implemented and tested on the images belonging to the two above-mentioned classes. The three classifiers were evaluated and compared based on their testing accuracy.

Table 2. Evaluation of cyclist travel lane detector

Sr. No.	Classifier	Train accuracy	Test accuracy	Precision	Recall	F1 Score
1	Random Forest	99%	85.44 %	0.84	0.97	0.90
2	Decision Tree	97.25%	83.22 %	0.84	0.93	0.88
3	KNN	76.08%	73 %	0.76	0.88	0.82

Table 2 showcases the performance analysis of the three classifiers trained for the detection of cyclist travel lanes. The random forest could give the highest training accuracy, testing accuracy, precision, recall, and F1 score followed by the decision tree and KNN.

4.2 Cyclist's helmet detector

The cyclist's helmet detector implies the classification of images based on two classes: cyclist with a helmet and cyclist without a helmet. The cyclist with helmet class is labeled as '0' and the cyclist without helmet class is labeled as ‘1’. The classifiers such as decision tree, random forest, KNN, GNB, AdaBoost, and SVM were implemented and tested on the images belonging to the two above-mentioned classes. The SVM classifier was implemented using 4 different kernels. Linear, polynomial, RBF, and sigmoid kernels were implemented in SVM. The six classifiers were evaluated and compared based on their testing accuracy.

Table 3. Evaluation of cyclist’s helmet detector

Sr. No.	Classifier	Train accuracy	Test accuracy	Precision	Recall	F1 Score
1	Decision Tree	99%	83.86%	0.83	0.84	0.84
2	Random Forest	99.53%	87.83%	0.86	0.86	0.88
3	KNN	89.23%	88.41%	0.86	0.91	0.89
4	GNB	85.50%	84.08%	0.81	0.88	0.85
5	AdaBoost	85.82%	84.97%	0.83	0.87	0.85
6	SVM					
	a) Linear:	85.49%	84.44%	0.81	0.89	0.85
	b)Polynomial:	84.99%	85.33%	0.89	0.80	0.84
	c) RBF:	91.12%	88.66%	0.87	0.91	0.89
	d) Sigmoid:	59.83%	59.62%	0.59	0.60	0.60

Table 3 displays the performance analysis of the six classifiers trained for the detection of cyclists’ helmets. The random forest could give the highest training and testing accuracy followed by the decision tree and SVM. The SVM classifier using a polynomial kernel could give notable precision, recall, and F1 score followed by KNN and random forest.

Table 4. Comparison of proposed system with existing systems

Ref. No.	Approach	Hardware	Run time	Accuracy
[1]	SVM	a computer with 8 cores of 2.7 GHz and a memory of 8 GB	0.28 s	-
[3]	CNN	Windows 10, NVIDIA GEFORCE 940 MX.	moderate time	80%
[20]	LSTM + RNN	GeForce GTX 1660 GPU	15.6 ms	91%
Proposed system	SIFT + Random forest	Intel Core i5-7200U CPU with 2.50GHz processor, Windows 10	Moderate time	Cyclist lane detection: 85.44% Helmet detection: 87.83%

The comparison of the proposed system and the existing systems for cyclist lane and helmet detection is illustrated in table 4. The model for vision-based cyclist travel lane and helmet detection was implemented in Jupyter notebook IDE on an Intel Core i5-7200U CPU with a 2.50GHz processor and windows 10 operating system. The algorithm giving the highest testing accuracy in both cyclist travel lane detector and helmet detector is taken into consideration for building the vision-based cyclist travel lane and helmet detection system. The random forest algorithm having a test accuracy of 85.44% for cyclist travel lane detection and 87.83% for cyclist’s helmet detection was used for cyclist helmet detection.

5 Conclusion and future scope

Cyclist travel lanes are being constructed to provide a distinct place for cyclists from vehicles and motorcycles on congested roadways. These are also known as protected cyclist lanes. The main motive of this paper was to classify whether a cyclist is traveling in his/her lane and wearing a helmet for safety purposes. Categorization of whether or not the cyclist is traveling in a specified lane is made which follows the decision of whether the cyclist is wearing a helmet or not. The important features in the dataset for travel lane and helmet detection were collected using SIFT. The comparison of binary classification of both the cyclist travel lane and cyclist's helmet model was carried out using machine learning algorithms such as KNN, decision tree, random forest, AdaBoost, and GMV. The random forest classifier had the highest training accuracy of 99% and testing accuracy of 85.44% for cyclist travel lane detection, while for cyclist helmet detection, the random forest had the highest training accuracy of 99.53% and testing accuracy of 87.83%. Even though this model produces a good result, factors such as better image quality and a larger number of images can help to improve the model's accuracy in future. Neural networks based algorithm can also improve the accuracy in future. This system can be further used in real-time surveillance and autonomous driver assistance system, for which the model must be trained with high-quality images.

References

1. W. Tian and M. Lauer.: Fast Cyclist Detection by Cascaded Detector and Geometric Constraint. In: 18th IEEE International Conference on Intelligent Transportation Systems, pp. 1286-1291. (2015)
2. X. Li et al.: A Unified Framework for Concurrent Pedestrian and Cyclist Detection. In: IEEE Transactions on Intelligent Transportation Systems, vol. 18. no. 2. pp. 269-281. (2017)
3. C. A. Rohith, S. A. Nair, P. S. Nair, S. Alphonsa and N. P. John.: An Efficient Helmet Detection for MVD using Deep learning. In: 3rd International Conference on Trends in Electronics and Informatics, pp. 282-286. (2019)
4. R. R. V. e. Silva, K. R. T. Aires and R. d. M. S. Veras.: Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers. In: 27th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 141-148. (2014)
5. Shine, L., C. V., J.: Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN. In: Springer -Multimedia Tools Application, pp. 14179–14199. (2020)
6. Kumar N., Sukavanam N.: Detecting Helmet of Bike Riders in Outdoor Video Sequences for Road Traffic Accidental Avoidance. In: Springer - Advances in Intelligent Systems and Computing, vol 941. pp. 24-33. (2018)
7. Xiaofei Li et al.: A new benchmark for vision-based cyclist detection. In: IEEE Intelligent Vehicles Symposium, pp. 1028-1033. (2016)
8. F. Zhou, H. Zhao and Z. Nie.: Safety Helmet Detection Based on YOLOv5. In: IEEE International Conference on Power Electronics, Computer Applications, pp. 6-11. Shenyang, China (2021)
9. F. Wu, G. Jin, M. Gao, Z. HE and Y. Yang: Helmet Detection Based On Improved YOLO V3 Deep Model. In: IEEE 16th International Conference on Networking, Sensing and Control, pp. 363-368. (2019)
10. R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras and A. Soares.: Automatic detection of motorcyclists without helmet. In: IEEE XXXIX Latin American Computing Conference, 2013, pp. 1-7. Caracas, Venezuela (2013)
11. E. A. I. Pool, J. F. P. Kooij and D. M. Gavrila.: Context-based cyclist path prediction using Recurrent Neural Networks. In: IEEE Intelligent Vehicles Symposium, pp. 824-830. Paris, France (2019)
12. E. A. I. Pool, J. F. P. Kooij and D. M. Gavrila.: Using Road topology to improve cyclist path prediction. In: IEEE Intelligent Vehicles Symposium, pp. 289-296. Los Angeles, CA, USA, (2017)
13. S. Zernetsch, S. Kohnen, M. Goldhammer, K. Doll and B. Sick: Trajectory prediction of cyclists using a physical model and an artificial neural network. In: IEEE Intelligent Vehicles Symposium, pp. 833-838. Gothenburg, Sweden (2016)
14. Saleh K., Hossny M., Nahavandi S.: Cyclist Trajectory Prediction Using Bidirectional Recurrent Neural Networks. In: Springer - Australasian Joint Conference on Artificial Intelligence, pp 284–295. (2018)
15. R. G. J. Wijnhoven and P. H. N. de With: Fast Training of Object Detection Using Stochastic Gradient Descent. In: 20th International Conference on Pattern Recognition, pp. 424-427. Istanbul, Turkey (2010)
16. Basch, C.H., Zagnit, E.A., Rajan, S. et al.: Helmet Use among Cyclists in New York City. In: Journal of Community Health, pp. 956–958. (2014)
17. Joceline Rogé, Daniel Ndiaye, Isabelle Aillerie, Stéphane Aillerie, Jordan Navarro, Fabrice Vienne.: Mechanisms underlying cognitive conspicuity in the detection of cyclists by car drivers. In: Accident Analysis & Prevention, Vol. 104. pp. 88-95. (2017)
18. François Char, Thierry Serre.: Analysis of pre-crash characteristics of passenger car to cyclist accidents for the development of advanced drivers assistance systems. In: Accident Analysis & Prevention, Vol. 136. (2020)
19. X. Li et al.: A Unified Framework for Concurrent Pedestrian and Cyclist Detection. In: IEEE Transactions on Intelligent

Transportation Systems, vol. 18, no. 2, pp. 269-281, (2017)

20. Zhi Huang, Jun Wang, Lei Pi, Xiaolin Song, Lingfang Yang.: LSTM based trajectory prediction model for cyclist utilizing multiple interactions with environment. In: Pattern Recognition, Vol. 112. (2020)
21. P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan: Object Detection with Discriminatively Trained Part-Based Models. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32. no. 9. pp. 1627-1645. (2010)