Comparative analysis of feature descriptors and classifiers for real-time object detection

Vikas J. Nandeshwar[1], Sarvadnya Bhatlawande[1], Anjali Solanke[2], Harsh Sathe[1], Shivanand Satao[1], Safalya Satpute[1], Atharv Saste[1]
[1]Department of Engineering Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, India
[2]Department of Electronics and Telecommunication Engineering, Marathwada Mitra Mandal's College of Engineering, Pune, India

Article Info

Article history:

Received Apr 9, 2024
Revised Oct 4, 2024
Accepted Oct 18, 2024

Keywords:

Classifiers
Comparative analysis
Computer vision
Driver assistance system
Feature descriptors
Obstacle detection systems
Supervised machine learning

ABSTRACT

Detecting objects within complex environments, such as urban settings, holds significant importance across various applications, including driver assistance systems, traffic monitoring, and obstacle detection systems. Particularly crucial for these applications is the accurate differentiation between cars and roads. This study introduces a novel approach that leverages traditional feature descriptors and classifiers for real-time object detection. It conducts an exhaustive comparative analysis of feature descriptors and classifiers to identify the most effective model for real-time object detection. Handcrafted features of images are extracted using algorithms such as scale invariant feature transform (SIFT), oriented fast and brief (ORB), fast retina key-point (FREAK), and local binary pattern (LBP). Seven classifiers are employed, including support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), decision tree (DT), logistic regression (LR), Naive Bayes, and extreme gradient boosting (XGBoost). The performance of the 28 generated combinations of feature descriptors and classifiers is evaluated based on the parameters of accuracy, precision, F1 score, and recall. The model utilizing LBP and XGBoost achieves the highest accuracy, reaching 83.59%. The system architecture comprises a camera, a high-speed computing unit, a display, and an audio subsystem, with the algorithm implemented on a Raspberry Pi 4B (8 GB).

Corresponding Author:

Anjali Solanke
Department of Electronics and Telecommunication Engineering
Marathwada Mitra Mandal's College of Engineering
Pune, Maharashtra, India
Email: anjali.solanke1@gmail.com

1. INTRODUCTION

The task of object detection holds pivotal significance within the field of computer vision, finding extensive utility across diverse sectors. As per the findings of the U.S. National Highway Traffic Safety Administration, a substantial majority, exceeding 88% of traffic incidents arise due to errors in judgment or delayed reactions on the part of drivers [1]. As a solution, an intelligent driver assistance system for road safety must be developed which would alert drivers in dangerous situations [2]. Technological advancements in embedded systems and vehicle electronics are being driven by the goal of enhancing vehicle safety and energy efficiency [3]. The emergence of real time technologies like driver assistance systems and obstacle avoidance systems has highlighted the need for precise and instantaneous object detection methodologies. Notably, in these application domains, vehicles and roadways emerge as central entities, having profound influence over transportation networks and safety frameworks.

This study is dedicated to devising an object detection framework ported on an embedded system for instantaneous segmentation of cars and roads. The method employs hand crafted features extracted from traditional feature descriptors and statistical machine learning models. These models are assessed based on the parameters of accuracy, precision, F1 score, and recall. Through combination of traditional feature descriptors with classifiers, our objective is to enhance the effectiveness of object detection in real-time environments while maintaining the rapid response times and low computational power both critical for real-time applications.

2. LITERATURE SURVEY

In recent years, deep learning algorithms like region-based convolutional neural networks (R-CNN), you only look once (YOLO), and single shot multibox detector (SSD) have gained widespread adoption across various domains for object detection tasks [4]. Convolutional neural networks (CNNs) are widely recognized as one of the most representative and influential models in the domain of deep learning [5]. A method enhancing faster R-CNN for obstacle detection and recognition using U-V disparity maps and an improved network structure involving context aware modules trained on datasets from KITTI and CCD cameras is proposed in [6]. Dairi *et al*. [7] introduced a stereo vision method for obstacle detection in urban environments. They utilized a deep stacked auto-encoders (DSA) model and an unsupervised K-nearest neighbors (KNN) algorithm to detect obstacles reliably. A YOLO-v5-based object detection model capable of detecting objects of various sizes, including large, small, and tiny objects utilizing a multi-scale mechanism to adaptively determine the optimal scales for vehicle detection within a scene was proposed by Carrasco *et al*. [8]. Zhou *et al*. [9] proposed a road detection and tracking algorithm that accurately delineates the road region using a graph-cut-based approach. It employs a fast homography-based tracking technique, utilizing features from scale invariant feature transform (SIFT) and speeded up robust features (SURF) algorithms alongside the fast feature detector. A system designed to identify drivable and non-drivable roads, with the goal of mitigating traffic accidents was introduced which employed a combination of SIFT and binary robust invariant scalable key-points (BRISK) algorithms to extract features. These extracted features were subsequently input into classifiers. The system attained a 70.9% accuracy rate when utilizing the support vector machine (SVM) with a radial basis function (RBF) kernel in [10]. An algorithm utilizing principal component analysis-SIFT (PCA-SIFT), exploiting scale-invariant key points to ensure rotation and scale invariance was introduced where the PCA method, based on variable covariance matrices, efficiently handles, compresses, and extracts feature vectors was proposed in [11].

Retraining a YOLOv3 model for vehicle detection using a dataset of 400 unmanned aerial vehicle (UAV) images on a graphics processing unit (GPU), while demonstrating the computational demands of deep learning algorithms like YOLO, the model exhibited occasional failure to detect objects and generated false positives in some instances [12]. This article presents a comparative study of three detectors histogram of oriented gradient (HOGs), local binary pattern (LBP), and Haar features, for the detection of animals. These features were individually trained on a dataset using the AdaBoost algorithm. LBP-AdaBoost/HOG-SVM demonstrated effective performance under daytime conditions. However, limitations were observed during the nighttime operation [13]. An ensemble learning-based decision model for predicting vehicle lane-changing behavior employing random forest (RF), demonstrated higher precision in its predictions [14]. An embedded system for traffic surveillance that utilizes the NVIDIA Jetson TX1, integrating a deep detector named MF faster R-CNN which can identify various classes of traffic objects, including pedestrians, cars, buses, and motorbikes, concurrently was proposed in [15]. Zhang [16] proposed that employing the Canny edge detection algorithm can yield high accuracy as it identifies and removes unstable edge points. A solution to integrate multiple low-cost sensors, including infrared and ultrasonic technologies, offering improved reliability and reduced mathematical complexity was introduced in [17]. A lossy compression framework for input images was proposed to reduce memory traffic and power consumption, balancing accuracy and compression performance [18]. Conversion of image-based depth maps to light detection and ranging (pseudo-LiDAR) representations for object detection was suggested. This approach achieved a 74% detection accuracy within a 30 m range [19]. A method exploiting K means and PCA for large-scale data sets was presented that leads to both computational time and memory benefits. It utilized randomized preconditioning transformations to achieve accurate data sparsification, reducing variance in estimates [20]. A system utilizing ATMega328p microcontroller, ultrasonic sensor for object detection, MQ3 sensor for alcohol detection and Image processing to detect driver’s drowsiness is proposed in [21]. A configuration including Raspberry Pi board-based control system, a monocular camera and an earphone for recognizing the position of an object was exploited in [22]-[24]. SIFT feature descriptor for feature extraction and classifiers such as KNN and RF are leveraged in [25]. A security system integrating facial features from FaceNet and Mediapipe for face detection was observed to achieve an accuracy of 80.5% [26].

3. METHOD

This research paper presents a real time system designed for the classification and detection of vehicles on roadways, comprising a camera and a fast-computing processor-based system, as illustrated in Figure 1. The system captures the data from its environment, which is then provided as an input to the fast-computing processor. A specialized obstacle detection algorithm, implemented on a Raspberry Pi 4B (8 GB) board, is designed to understand and interpret road surroundings. This algorithm segments vehicles and roadways, deduces priority information, and provides details about the drivable road. The system translates vehicle presence into textual and auditory feedback, which is then relayed to the driver through a compact audiovisual system integrated into the car's dashboard, ensuring efficient communication.

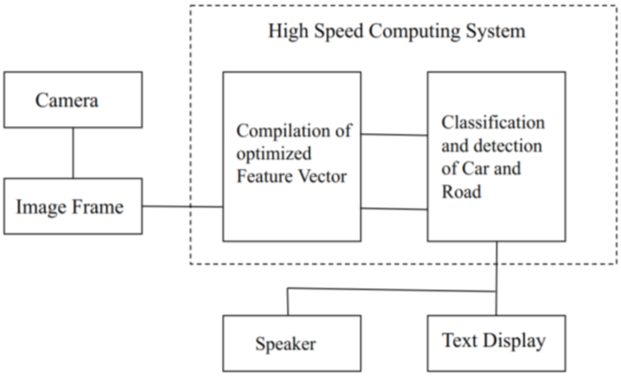
High Speed Computing System
Camera
Image Frame
Compilation of optimized Feature Vector
Classification and detection of Car and Road
Speaker
Text Display

Figure 1. System for detection of cars and roads

3.1. Details of the dataset and pre-processing

The dataset utilized in this study comprises a total of 5,400 images, encompassing 2,700 positive images depicting cars on roads and an equal number of negative images depicting empty roadways. The images in the dataset were curated by the authors. The authors employed a 50-megapixel camera on an android mobile phone to capture vehicle photographs in landscape orientation. The composition of images in the dataset is presented in Table 1. Each image within the dataset underwent a standard resizing process, resulting in dimensions of 280×430 pixels, and was subsequently converted to grayscale. The edge detection algorithms of Prewitt and Canny were applied uniformly across the entire dataset. These techniques were employed to identify and extract edges from the images while minimizing noise and extraneous elements, thereby enhancing the overall visual quality of the images.

Table 1. Details of the dataset

Image type	Train images	Test images
Positive images	2160	540
Negative images	2160	540
Total images	4320	1080

3.2. Feature extraction

The traditional feature descriptors SIFT, ORB, FREAK, and LBP were employed to extract key-points and hand-crafted features for each image x_i in the dataset (where i ranges from 0 to N-1 where, N represents the total number of images, i.e., 5400). SIFT was opted for due to its well-established robustness and resilience to affine transformations and noise change factors [27]. ORB was selected for its combination of the FAST detector's speed with the BRIEF descriptor's durability, offering rotation invariance and real-time performance [28]. Fast retina key-point (FREAK) was selected owing to its accelerated performance compared to SIFT and SURF, leveraging insights from human visual subregion structure to obtain information [29]. Notably, SIFT key-points were provided to the FREAK descriptor in this study. LBP was preferred due to its computational efficiency, ease of implementation, and generation of a concise feature vector. Additionally, LBP finds widespread application in texture analysis and facial recognition tasks [30]. F represents the set of feature descriptors {SIFT, ORB, FREAK, and LBP} where the application of a feature descriptor F_j on x_i image generates a feature vector $F_j(x_i)$ representing the image's characteristics. The feature extraction process is denoted by (1):

$$V_l = \bigcup_{i=0}^{N-1} F_j(x_i) \quad (1)$$

where V_l denotes the feature vector obtained by appending the feature vectors of x_i images. The feature vectors generated by different feature descriptors (F_j) are as follows: SIFT produces a vector with dimensions (3,419,678 rows×128 columns), ORB generates one with (3,314,807 rows×32 columns), FREAK results in (2,731,168 rows×64 columns), and LBP creates a vector of (5,400 rows×26 columns).

3.3. Feature transformation

The resulting feature vectors obtained from each descriptor tend to be high-dimensional, presenting challenges such as heightened computational complexity and vulnerability to overfitting. To mitigate these challenges, a two-step novel approach for feature transformation, encompassing K-means clustering followed by PCA was adopted. K-means clustering was employed, which is an unsupervised machine learning technique, to group similar data points into a predetermined number of clusters (K), determined using the elbow method [31], resulting in K=14. Following the application of the K-means clustering algorithm, the initial high dimensional feature vectors obtained from feature descriptors were transformed into a reduced dimensionality of (5,400×14). PCA is further utilized to acquire uncorrelated features and reduce the dimensionality of the feature vector while maintaining the essential structure and variability of the original data. Two scenarios are investigated: one focused on preserving 90% of the information and the other targeting 95% within the transformed feature vectors. The determination of the retained information percentage is performed by analyzing the cumulative explained variance ratio and subsequently transforming the feature vector through the selection of principal components (p). The overall process of dimension reduction is shown in Figure 2.

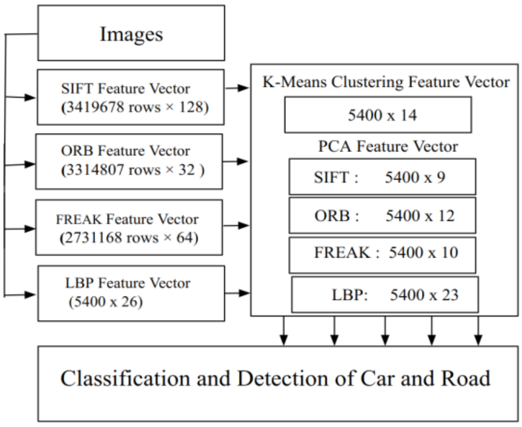
Images
SIFT Feature Vector (3419678 rows × 128)
ORB Feature Vector (3314807 rows × 32)
FREAK Feature Vector (2731168 rows × 64)
LBP Feature Vector (5400 x 26)
K-Means Clustering Feature Vector
5400 x 14
PCA Feature Vector
SIFT : 5400 x 9
ORB : 5400 x 12
FREAK : 5400 x 10
LBP: 5400 x 23
Classification and Detection of Car and Road

Figure 2. Workflow for feature transformation process

This dimension reduction can be denoted by (2):

$$\Psi(V_l) = (V_l * C[:, 0:p]) \quad (2)$$

where, $\Psi(V_l)$ represents the final dimensionally reduced feature vector and C denotes the principal components matrix. After obtaining the final feature vector, labels are appended to the last column to prepare the data for subsequent training and testing of classifiers. Specifically, a label of 0 represents the class "Car on Road," while a label of 1 signifies “Not Car on Road”. The final feature vector obtained after the application of K-Means and PCA is presented in Table 2.

Table 2. Final feature vector obtained after applying K-Means and PCA

Sr No.	Case 1: (90% of information retained)		Case 2: (95% of information retained)	
	Feature descriptor	Feature vector	Feature descriptor	Feature vector
1.	SIFT	5,400×5	SIFT	5,400×9
2.	ORB	5,400×9	ORB	5,400×12
3.	FREAK	5,400×6	FREAK	5,400×10
4.	LBP	5,400×19	LBP	5,400×23

3.4. Object detection and classification

The feature vector obtained post PCA was divided into training and testing the data with a ratio of 80:20. The training set, comprising 80% of the data, was utilized to train the classifiers, while the remaining 20% served as the testing set for evaluation. The classifiers employed in this study include: i) decision tree (DT) with a maximum depth set to 13; ii) RF with the number of estimators set to 100; iii) RBF kernel-SVM; iv) KNN with a parameter for the number of neighbors set to 20; v) logistic regression (LR); vi) extreme gradient boosting (XGBoost); and vii) Gaussian Naive Bayes (GNB). These classifiers were aimed at detecting obstacles categorized into two classes: "Car on Road" (label 0) and "Not Car on Road" (label 1). G represents the set of classifiers where G={DT, RF, RBF-SVM, KNN, LR, XGBoost, and GNB}. The classification process can be represented by (3):

$$c_{ik} = G_k\,(F_j(x_i))) \tag{3}$$

where c_{ik} denotes the predicted class label for the i-th image by the k-th classifier from the set G and $G_k\,(F_j(x_i))$ represents the model of the j-th feature descriptor from F and the k-th classifier from the set G. Considering feature descriptors (SIFT, ORB, FREAK, and LBP) and classifiers (SVM, DT, RF, LR, KNN, XGBoost, and GNB), the total number of models ($|M|$) generated can be expressed using (4):

$$|\,M| = (\,^{|F|}C_f \,*\, ^{|G|}C_{\,g}\,) \tag{4}$$

$$|\,M| = \frac{|F|!}{f\,!\,*(|F|-f)!} \,*\, \frac{|G|!}{g\,!\,*(|G|-g)!} \tag{5}$$

where, $|F|$ and $|G|$ denote the total number of feature descriptors and classifiers in sets F and G, respectively. Conversely, f and g denote the number of feature descriptors and classifiers selected from sets F and G. The values of, f and g always remains one as the total number of models ($|M|$) produced are evaluated on a set of parameters (PM), which include accuracy, precision, F1, and recall scores. Figure 3 showcases how the $|M|$ i.e., 28 combinations of feature descriptors and classifiers are generated. In (5) provides a detailed mathematical foundation for (4), explaining how the total number of models |M| is computed by considering all possible combinations of feature descriptors and classifiers from sets F and G.

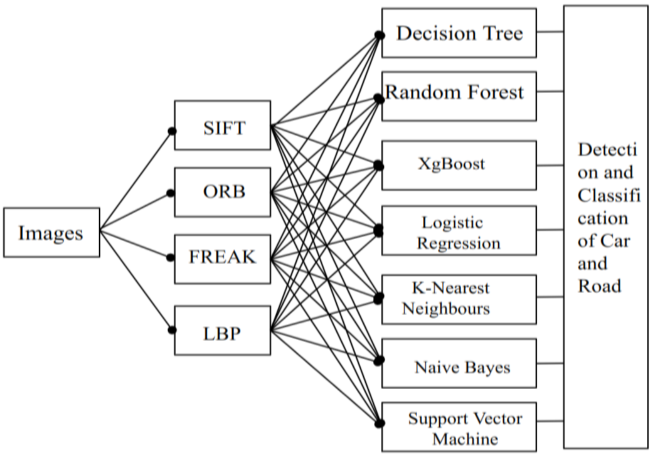

Figure 3. Illustrates the generation of 28 combinations of feature descriptors and classifiers

.

The performance of classifiers such as KNN, RF, and DT was found to exhibit minimal improvements when their respective hyperparameters were optimized to enhance accuracy. This hyperparameter tuning process encompassed experimenting with diverse parameter settings, including modifying K-neighbor values for KNN, optimizing the number and depths of trees for RF, and fine-tuning the maximum depths of DT. However, the resultant changes in accuracy were surprisingly slight, generally fluctuating within a narrow range of 1% to 1.4%. The hyperparameter values which provided the optimal results were utilized in the respective models.

3.5. Comparative analysis

To determine the most reliable model for real-time object detection, each of the $|M|$ combinations, undergoes meticulous evaluation, which relies on an exhaustive examination of every combination's performance across critical parameters of accuracy, F1 score, recall score, and precision. Through this comparative analysis, insights into the effectiveness and robustness of each model are obtained, allowing for the identification of the optimal feature descriptor-classifier combination suited for real-time object detection. The novelty of this system lies in its utilization of a comparative analysis method aimed at identifying the model ($M(F_j, G_k)$) of the j-th feature descriptor and k-th classifier from the sets F and G, with the paramount gradings across all parameters. If the scores for the selected model decline, this comparative analysis is performed again, which returns the model with superior scores than the previous model. This iterative process can be described by (6):

$$M_{max}(F_j, G_k) = \arg\max\left(\sum w_{PM}\ P(F_j, G_k, PM)\right) \tag{6}$$

where, $PM = \{Accuracy, Precision, F1, and\ Recall\ scores\}$ and $P(F_j, G_k, PM)$ defines the performance scores of the combination of feature descriptor F_j and classifier G_k with respect to the parameters in the set PM. Additionally, w_{PM} denotes the weight assigned to each parameter in the set PM. The gradings of model M_{max} across all parameters are observed and if there is a decline in these scores, (5) is utilized again for ascertaining the model M_{max} which has superior results across all metrics mentioned in set PM. The functioning of this framework ported on an embedded system is executed as shown in Algorithm 1.

Algorithm 1. Feature extraction and feature transformation

```
Input: Image dataset of N Images
Output: Classification of Car and Road
Data$_{F_j}$=[ ]; // data frame for features extracted by feature descriptors |$F$|
For image in dataset do
Preprocessing
Extract SIFT, ORB, FREAK, and LBP features of $x_i$ images
$Data_{F_j}$=append features //for $F_{j\,th}$ feature descriptor
$V_l$=Data$_{F_j}$. Append ($F_j(x_i)$)
End For
 For each image feature $F_j(x_i)$ in $V_l$ :
 $K(V_l)$=Pre-trained K-Means [K=14] ( $V_l$)
 //Dimension of $V_l$ is (5400 x 14)
Nm($V_l$) =Normalize ($V_l$ );
Data$_{F_j}$.append(N)
End For
$\Psi(V_l) = PCA(V_l)$ [ with principal components ($p$)]
//Dimension of $\Psi(V_l)$ is (N x $p$ )
Train classifiers from $G$ on 0.8N images
$c_{ik}$ = $G_k(F_j(x_i))$) //$c_{ik}$ represents the prediction
$(F_j, G_k) = \arg\max\left(\sum w_{PM}\ P(F_j, G_k, PM)\right)$
For each parameter in $PM$:
if $P_{M_{max}}(F_j, G_k, PM)$ < $P_{M_{current}}(F_j, G_k, PM)$
Set $M_{max}$ = $M_{current}$
Return $M_{max}$
End For
```

4. RESULTS AND DISCUSSION

The results of the proposed methodology are extensively discussed herein. The implementation was conducted using Jupyter Notebook, optimized for GPU utilization to enhance computational efficiency. The entire model was then ported on a Raspberry Pi 4B (8 GB) board. Following the training of classifiers on 80% of the dataset images, testing was conducted on the remaining 20%. The evaluation of each model among the 28 combinations was performed based on key metrics such as accuracy, precision, F1 score, and recall. Noteworthy is the model utilizing the combination of LBP and XGBoost, which demonstrated the highest accuracy, achieving 81.48% when retaining 90% of information post PCA (case 1), and 83.5% when retaining 95% of information (case 2).

In case 1, with 90% data retention post-PCA, SIFT with RF achieved a testing accuracy of 73.8%, with class 0 (car) precision, F1, and recall at 75.03%, 74.0%, 72.05%, and class 1 (road) scores at 73.9%, 76.1%, and 78.02%. ORB with RF attained a testing accuracy of 78.98%, with class 0 scores of 79.2%, 78.6%, and 79.0%, and class 1 scores of 78.0%, 78.1%, and 78.6%. FREAK exploited with KNN achieved a testing accuracy of 75.09%, with class 0 scores at 76.4%, 75.0%, and 75.6%, and class 1 scores at 74.8%, 76.1%, and 76.2%. LBP with XGBoost achieved the highest accuracy at 81.48%, with class 0 scores of 82.1%, 81.1%, and 80.2%, and class 1 scores at 81.0%, 82.3%, and 82.5%. In case 2, with 95% data retention, scores across all parameters improved. Among 28 models, SIFT and FREAK with RF and SVM outperformed other SIFT-based models, while ORB with SVM and LR yielded higher scores compared to other ORB models. LBP, combined with RF and XGBoost, delivered the best overall results. Case 2 results are presented in Table 3.

Table 3. Illustrates the results of the models when 95% of the information is retained (case 2)

Feature descriptor+classifier	Testing accuracy (%)	Label	Precision (%)	F1 (%)	Recall (%)
SIFT+RF	78.05	0	79.2	77.4	76.3
		1	77.3	79.1	81.6
SIFT+SVM	77.22	0	81.0	76.8	71.4
		1	74.0	79.3	84.2
ORB+SVM	80.46	0	82.0	79.1	76.4
		1	78.0	81.9	84.8
ORB+LR	79.53	0	79.4	79.7	80.0
		1	79.1	79.2	78.6
FREAK+RF	76.57	0	77.2	76.5	76.0
		1	76.8	77.8	77.0
FREAK+SVM	75.5	0	78.4	74.5	71.2
		1	73.6	77.0	80.0
LBP+RF	82.77	0	81.0	83.7	86.3
		1	85.1	82.2	80.1
LBP+XGBoost	83.5	0	82.1	83.5	85.0
		1	85.0	84.0	81.4

In both case 1 and 2, it was observed that LBP features, in combination with ensemble classifiers, achieved superior scores for both class labels (0 and 1). Notably, LBP outperformed other classifiers by having the highest testing accuracy ranging from 76.14% to 83.57%, thus indicating its ability to capture essential feature values for real-time obstacle detection. In Case 1, a fusion of LBP and XGBoost resulted in an F1 score of 85% for car detection (class 0), demonstrating balance between precision and recall. SIFT performed well for class 1 (“Not Car on Road”), but it was less accurate in car detection (class 0) compared to LBP and ORB, suggesting that SIFT’s features may be less efficient at spotting cars in diverse backgrounds. Yet, SIFT performed well when categorizing roads, attesting to its capacity for capturing nuances in ‘Not Car on Road’ elements. ORB was also able to achieve an accuracy of 80.46% with SVM. On the other hand, FREAK underperformed across all these measures consistently implying that it fails to capture intricate patterns in the dataset. Notable shifts were observed among the most accurate models of each feature descriptor when there was a change in amount of information retained post PCA. Specifically, concerning SIFT, there was a notable decline in the results `when only 90% of the information was retained. For SIFT, KNN relatively demonstrated superior performance when a lesser amount of information was retained. As for ORB, RF exhibited better performance compared to RBF-SVM in case 1. Interestingly, ORB exhibited resilience to changes in the retained information by producing slightly lower scores across the metrics. Contrarily, for FREAK, KNN, and LR emerged as FREAK’s most accurate models, replacing its combination with RF and SVM in case 2. On the other hand, LBP maintained its effectiveness with ensemble-based classifiers such as XGBoost and RF by delivering comparatively superior results across all the parameters. It was observed that the performance of different models of feature descriptors and classifiers was influenced by the percentage of information retained post PCA in dimensionality reduction. Notably, all models encountered relatively more challenges in predicting cars (class 0) compared to road (class 1), perhaps owing to the complexity involved in distinguishing cars from various distinct and diverse environmental factors.

Balancing computational efficiency and accuracy is crucial for selecting models suitable for real-time applications. By assessing the time complexities of feature descriptors and classifiers, including their training and prediction phases, we can estimate the model's overall time complexity. SIFT emerges as the most computationally intensive, while ORB and FREAK are observed to be significantly more efficient [32]. LBP stands out for its computational efficiency and strong performance in human detection tasks [33]. After analyzing the time complexity across the 28 models incorporating feature descriptors and classifiers, it was observed that combinations implicating LBP achieved lower time complexities than other feature descriptors. The pairing of LBP with the GNB classifier resulted in the least time complexity of [O(n*d)], where 'n' signifies the samples in the training dataset and 'd' denotes the dimensionality or the number of features in each sample. However, despite this efficiency, the achieved accuracy stood at 76.8%, which was relatively lower compared to other LBP-based models. Furthermore, the KNN model exhibited a time complexity of [O(n*d)], similar to GNB but with negligible training time complexity. However, it exhibited a lower accuracy rate of 76.1%, positioning it as one of the least accurate among the LBP-based models. The model exhibiting the worst-case time complexity ranging from [O(d*n^2) to O(d*n^3)] was observed when SIFT was coupled with the RBF-SVM classifier. Conversely, when LBP was combined with ensemble-based classifiers, superior performance across all parameters was observed. However, this came at the cost of increased time complexity. With RF, the time complexity reached [O(K*d*nlog(n))], and with XGBoost, the time complexity rose to [O(K*d*|x| log n)], where K is the number of trees, d is the height of the trees, n represents the total number of data points used to train the model, and |x| is the number of non-missing entries in the training data [34]. These complexities notably surpass those achieved with KNN, GNB, and LR. The LBP-based model demonstrated a worst-case time complexity of [O (d*n^2)], where n is the number of data points in the training set and d is the number of features in the data when coupled with the SVM classifier. Contrarily, models employing the GNB classifier achieved the lowest time complexity, albeit with inferior scores across all parameters. Despite KNN and LR classifiers performing well in certain scenarios, their overall performance was surpassed by ensemble-based classifiers and SVM. This observation indicated the trade-off between accuracy and time complexity delivered by the model in this particular case. The receiver operating characteristic (ROC) curve and area under the ROC curve (AUC) for the XGBoost classifier are illustrated in Figure 4. Notably, the AUC for both the "Car on Road" class with label 0 and the "Not Car on Road" class with label 1 was observed to be 91%.

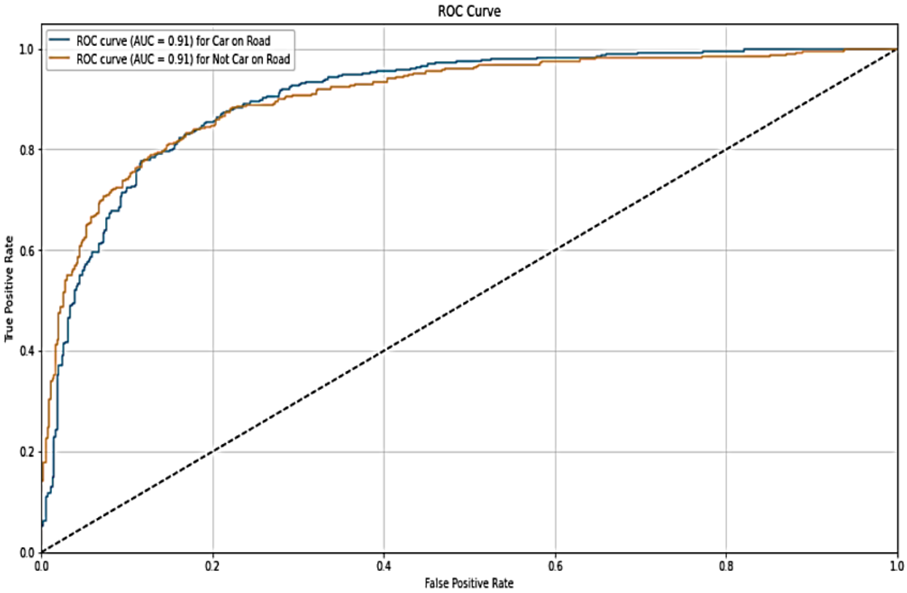

Figure 4. ROC curve

5. CONCLUSION

This study proposed a solution aimed at detecting and classifying cars and roads using handcrafted features extracted from traditional feature descriptors and employing seven classifiers. A total of 28 models were created and assessed based on accuracy, precision, F1, and recall scores, alongside considerations of time complexity. Among these models, the one leveraging LBP and XGBoost attained a testing accuracy of 83.57% and outperformed others across all parameters. The adoption of classical machine learning techniques rendered the system computationally efficient. Furthermore, a novel dimension reduction technique integrating K Means and PCA was employed. The implemented solution was deployed on a Raspberry Pi 4B (8 GB) board, demonstrating satisfactory performance in real-world scenarios. While the system effectively detected cars and roads, it encountered misclassification under low illumination conditions. To address this limitation, the authors intend to explore a deep learning-based approach capable of handling multiple images.

REFERENCES

[1] H. Li, J. Li, Z. Su, X. Wang, and J. Luo, "Research on active obstacle avoidance control strategy for intelligent vehicle based on active safety collaborative control," in *IEEE Access*, vol. 8, pp. 183736-183748, 2020, doi: 10.1109/ACCESS.2020.3029042.

[2] H. Bilal, B. Yin, J. Khan, L. Wang, J. Zhang, and A. Kumar, "Real-time lane detection and tracking for advanced driver assistance systems," *2019 Chinese Control Conference (CCC)*, Guangzhou, China, 2019, pp. 6772-6777, doi: 10.23919/ChiCC.2019.8866334.

[3] Y. Torres-Berru and P. Torres-Carrion, "Development of machine learning model for mobile advanced driver assistance (ADA)," *2019 International Conference on Information Systems and Software Technologies (ICI2ST)*, Quito, Ecuador, 2019, pp. 162-167, doi: 10.1109/ICI2ST.2019.00030.

[4] S. V. Mahadevkar *et al*., "A review on machine learning styles in computer vision—techniques and future directions," in *IEEE Access*, vol. 10, pp. 107293-107329, 2022, doi: 10.1109/ACCESS.2022.3209825.

[5] Z. -Q. Zhao, P. Zheng, S. -T. Xu, and X. Wu, "Object detection with deep learning: a review," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[6] J. Leng, Y. Liu, D. Du, T. Zhang, and P. Quan, "Robust obstacle detection and recognition for driver assistance systems," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1560-1571, Apr. 2020, doi: 10.1109/TITS.2019.2909275.

[7] A. Dairi, F. Harrou, Y. Sun, and M. Senouci, "Obstacle detection for intelligent transportation systems using deep stacked autoencoder and k -nearest neighbor scheme," in *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5122-5132, Jun. 15, 2018, doi: 10.1109/JSEN.2018.2831082.

[8] D. P. Carrasco, H. A. Rashwan, M. Á. García, and D. Puig, "T-YOLO: tiny vehicle detection based on YOLO and multi-scale convolutional neural networks," in *IEEE Access*, vol. 11, pp. 22430-22440, 2023 doi: 10.1109/ACCESS.2021.3137638.

[9] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297-309, Feb. 2015, doi: 10.1109/TITS.2014.2331353.

[10] S. Bhatlawande, A. Patil, S. Shilaskar, K. Patil, and V. Pingat, "A monocular vision based system for detection of drivable road," *2023 First International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV)*, Hyderabad, India, 2023 pp. 1-5, doi: 10.1109/ICPEEV58650.2023.10391905.

[11] J. Dai-Hong, D. Lei, L. Dan, and Z. San-You, "Moving-object tracking algorithm based on PCA-SIFT and optimization for underground coal mines," in *IEEE Access*, vol. 7, pp. 35556-35563, 2019, doi: 10.1109/ACCESS.2019.2899362.

[12] Q. Wu and Y. Zhou, "Real-time object detection based on unmanned aerial vehicle," *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, Dali, China, 2019, pp. 574-579, doi: 10.1109/DDCLS.2019.8908984.

[13] A. Mammeri, D. Zhou, and A. Boukerche, "Animal-vehicle collision mitigation system for automated vehicles," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1287-1299, Sep. 2016 doi: 10.1109/TSMC.2015.2497235.

[14] X. Gu, J. Yu, Y. Han, M. Han, and L. Wei, "Vehicle lane change decision model based on random forest," *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China, 2019, pp. 115-120, doi: 10.1109/ICPICS47731.2019.8942520.

[15] A. Mhalla, T. Chateau, S. Gazzah, and N. E. B. Amara, "An embedded computer-vision system for multi-object detection in traffic surveillance," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4006-4018, Nov. 2019, doi: 10.1109/TITS.2018.2876614.

[16] W. Zhang, "Combination of SIFT and canny edge detection for registration between SAR and optical images," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, doi: 10.1109/LGRS.2020.3043025.

[17] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," in *IEEE Access*, vol. 3, pp. 599-609, 2015, doi: 10.1109/ACCESS.2015.2432455.

[18] L. Guo, D. Zhou, J. Zhou, S. Kimura, and S. Goto, "Lossy compression for embedded computer vision systems," in *IEEE Access*, vol. 6, pp. 39385-39397, 2018, doi: 10.1109/ACCESS.2018.2852809.

[19] Y. Wang, W. -L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: bridging the Gap in 3D object detection for autonomous driving," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 8445-8453.

[20] F. Pourkamali-Anaraki and S. Becker, "Preconditioned data sparsification for big data with applications to PCA and K-Means," in *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954-2974, May 2017, doi: 10.1109/TIT.2017.2672725.

[21] S. Shilaskar, V. Patil, S. Pawar, A. Poke, R. Jambhulkar, and S. Bhatlawande, "Driver safety system using microcontroller and image processing," *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 340-344, doi: 10.1109/IDCIoT56793.2023.10053512.

[22] S. Bhatlawande *et al*., "AI based handheld electronic travel aid for visually impaired people," *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, Mumbai, India, 2022, pp. 1-5, doi: 10.1109/I2CT54291.2022.9823962.

[23] K. M. Masal, S. Bhatlawande, and S. D. Shingade, "Development of a visual to audio and tactile substitution system for mobility and orientation of visually impaired people: a review," *Multimedia Tools and Applications*, vol. 83, pp. 20387–20427, 2024, doi: 10.1007/s11042-023-16355-0.

[24] J. Madake, S. Bhatlawande, S. Purandare, S. Shilaskar, and Y. Nikhare, "Dense video captioning using BiLSTM encoder," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824569.

[25] S. Bhatlawande, D. Khapre, M. Kinge, and T. Khairnar, "Vision based assistive system for fall detection," *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022, pp. 1-7, doi: 10.1109/CONIT55038.2022.9847697.

[26] S. Bhatlawande, S. Shilaskar, T. Gadad, S. Ghulaxe, and R. Gaikwad, "Smart home security monitoring system based on face recognition and android application," *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 222-227, doi: 10.1109/IDCIoT56793.2023.10053558

[27] T. Yawen and G. Jinxu, "Research on vehicle detection technology based on SIFT feature," *2018 8th International Conference on Electronics Information and Emergency Communication (ICIEC)*, Beijing, China, 2018, pp. 274-278, doi: 10.1109/ICEIEC.2018.8473575.

[28] C. Yao, H. Zhang, J. Zhu, D. Fan, Y. Fang, and L. Tang, "ORB feature matching algorithm based on multi-scale feature description fusion and feature point mapping error correction," in *IEEE Access*, vol. 11, pp. 63808-63820, 2023, doi: 10.1109/ACCESS.2023.3288594.

[29] Y. Li, "A novel fast retina keypoint extraction algorithm for multispectral images using geometric Algebra," in *IEEE Access*, vol. 7, pp. 167895-167903, 2019, doi: 10.1109/ACCESS.2019.2954081.

[30] Y. Ding, Q. Zhao, B. Li and X. Yuan, "Facial expression recognition from image sequence based on LBP and Taylor expansion," in *IEEE Access*, vol. 5, pp. 19409-19419, 2017, doi: 10.1109/ACCESS.2017.2737821.

[31] F. Liu and Y. Deng, "Determine the number of unknown targets in the open world based on Elbow method," in *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 986-995, May 2021, doi: 10.1109/TFUZZ.2020.2966182.

[32] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," *2013 18th International Conference on Digital Signal Processing (DSP)*, Fira, Greece, 2013, pp. 1-7, doi: 10.1109/ICDSP.2013.6622757.

[33] J. Xu, Q. Wu, J. Zhang, and Z. Tang, "Fast and accurate human detection using a cascade of boosted MS-LBP features," in *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 676-679, Oct. 2012, doi: 10.1109/LSP.2012.2210870.

[34] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.

BIOGRAPHIES OF AUTHORS

Vikas J. Nandeshwar Assistant Professor at Vishwakarma Institute of Technology, Pune, specializes in sensors and applications, robotics, and mechatronics. His expertise includes data science, machine learning, deep learning, and natural language processing (NLP). His work is marked by contributions to conferences and journals, emphasizing his commitment to advancing technology in engineering disciplines. He can be contacted at email: vikas.nandeshwar1@vit.edu.

Sarvadnya Bhatlawande is a first-year engineering student at Vishwakarma Institute of Technology, Pune, with a deep passion for machine learning and computer vision. He actively explores advanced techniques in artificial intelligence and machine learning, aiming to expand his expertise in these dynamic fields. Despite being in the early stages of his academic journey, Bhatlawande exhibits a strong commitment to machine learning and artificial intelligence, driven by a desire to contribute significantly to their advancement. For further correspondence. He can be contacted at email: sarvadnya.bhatlawande23@vit.edu.

Anjali Solanke is currently working as an Associate Professor in the Department of Electronics and Telecommunication Engineering, Marathwada Mitra Mandal's College of Engineering, Pune, India. She is having 23 years of teaching experience. Her area of research is medical image processing and machine learning. She can be contacted at email: anjali.solanke1@gmail.com.

Harsh Sathe is a first-year B.Tech. student majoring in information technology (IT). He eagerly anticipates his inaugural work opportunity. He contributes a burgeoning perspective and a proactive approach to learning and research. They aspire to make meaningful contributions to the field of machine learning through collaborative efforts. He can be contacted at email: harsh.sathe23@vit.edu.

Shivanand Satao is currently a first-year engineering student at Vishwakarma Institute of Technology, Pune. He is committed to continuous learning, participating in workshops and seminars that enhance their understanding of computer vision. His involvement underscores a proactive approach in acquiring essential knowledge and skills for future research endeavors. He can be contacted at email: shivanand.satao23@vit.edu.

Safalya Satpute is currently a first-year engineering student at Vishwakarma Institute of Technology, Pune. Embarking on his academic journey, they are eager to explore the intricacies of machine learning and robotics. He actively engages with foundational concepts in artificial intelligence and machine learning and exhibits enthusiasm towards contributing to research initiatives. He can be contacted at email: safalya.satputee23@vit.edu.

Atharv Saste possesses proficiency in programming and automation. His primary interests encompass artificial intelligence (AI), machine learning (ML), and robotics, fields in which he aspires to make meaningful contributions. He can be contacted at email: atharv.saste22@vit.edu.