Multimedia Tools and Applications (2024) 83:71379–71406
https://doi.org/10.1007/s11042-024-18303-y

JUIVCDv1: development of a still-image based dataset for indian vehicle classification

Sourajit Maity[1] · Debam Saha[1] · Pawan Kumar Singh[2] · Ram Sarkar[1]

Received: 6 August 2023 / Revised: 28 December 2023 / Accepted: 19 January 2024 /
Published online: 7 February 2024

Abstract

An automatic vehicle classification (AVC) system designed from either still images or videos has the potential to bring significant benefits to the development of a traffic control system. On AVC, numerous articles have been published in the literature. Over the years, researchers in this domain have created and used a variety of datasets, but most often, these datasets may not reflect the exact scenarios of the Indian subcontinent due to specific peculiarities of the road conditions, road congestion nature, and vehicle types usually seen in Indian subcontinent. The primary goal of this paper is to create a new still image dataset, called JUIVCDv1, which contains 12 different local vehicle classes that are collected using mobile cameras in a different way for developing an automated vehicle management system. We have also discussed the characteristics of the current datasets, and various other factors taken into account while creating the dataset for the Indian scenario. Apart from this, we have benchmarked the results on the developed dataset using eight state-of-the-art pre-trained convolutional neural network (CNN) models, namely Xception, InceptionV3, DenseNet121, MobileNetV2, and VGG16, NasNetMobile, ResNet50 and ResNet152. Among these, the Xception, InceptionV3 and DenseNet121 models produce the best classification accuracy scores of 0.94, 0.93 and 0.92 respectively. These models are further utilized to make an ensemble model to enhance the performance of the overall categorization model. Majority voting-based ensemble, Weighted average-based ensemble, and Sum rule-based ensemble approaches are used as ensemble models that give accuracy scores of 0.95, 0.94, and 0.94, respectively.

Keywords Automatic vehicle classification · JUIVCDv1 · Still image · Indian vehicle dataset · Deep learning

Sourajit Maity and Debam Saha contributed equally to this work.

✉ Sourajit Maity
sourajit.cse.ju@gmail.com

Extended author information available on the last page of the article

1 Introduction

The need for more effective solutions to the traffic congestion issue has grown in recent years due to the increase in vehicle traffic. Automatic vehicle classification (AVC) systems are needed for real-time traffic management and monitoring to deal with the ever-increasing volume of traffic. The rising number of vehicles on the road has been a concern for researchers towards the betterment of the road traffic scenario. Many studies on the traffic management system have been published in areas such as vehicle categorization [1], detection [2, 3], make and model recognition [4–6], segmentation, lane detection, pedestrian detection, etc. Autonomous driving will play a huge role in vehicle-related research in the future. For this reason, research on AVC systems is much needed in today's scenario. Such systems can also be used to collect information about vehicle make and model details that are necessary for security reasons. Working on such problems using real-world traffic scenarios is difficult in terms of training, testing, and model validation. Huge amounts of realistic data are required for this purpose. Deep learning algorithms [7, 8] give better results in the case of real-time competitive performance in comparison to other machine learning algorithms and conventional approaches in various applications, and deep learning models strive to outperform previous results in a given domain. During the study of this domain, it has been found that there are not many datasets available for research, and many of them are based on speculative situations. Moreover, well-known AVC datasets are mostly paid, and the datasets with a decent number of images lack appropriate annotations, thus making it challenging to use them for research purpose. On the other hand, sufficient samples are needed to create an effective supervised learning-based model that is accurate as well as capable of functioning in real-life scenarios.

For vehicle localization, there are a few datasets available, but the number of datasets available for AVC is limited. Furthermore, existing AVC datasets do not mostly capture real-life scenarios adequately. For example, images taken in the Indian subcontinent frequently show multiple vehicles overlapping in a single frame due to heavy traffic congestion. This issue makes the classification, localization, detection, and segmentation processes extremely challenging. These challenges are relevant not only in India but also in Bangladesh, Pakistan, Sri Lanka, and many other South-Asian nations. The available AVC models which were trained on well-managed traffic scenarios, might not be applicable to the datasets collected from these nations. However, Bhattacharya et al. [9] developed a dataset, called JUVDsi, for vehicle detection in Indian road scenarios. Ali et al. [10] published a vehicle detection dataset, called IRUVD, which includes 14 vehicle classes typically based on the Indian roads scenario.

1.1 Research motivation and contributions

In this sub-section, the motivation behind this research work and its contributions have been discussed. For vehicle detection, a large number of datasets are found in the literature. However, only a few datasets are available for vehicle classification tasks. Not all datasets accurately reflect real-world situations. Similar to the Indian subcontinent, images taken in crowded traffic conditions sometime show many automobiles overlapped in a single image. This problem makes the classification, localization, detection, and segmentation processes very challenging. Also, there is no more room for research on some AVC datasets because researchers have already achieved almost 100% accuracy on those datasets. To reduce road accidents, it is also necessary to recognize vehicles [11], pedestrians [12] and re-identification

[13–15] of vehicles. There are several potential applications of AVC in various sectors, such as smart cities, transportation, law enforcement, and car industry. Security, traffic management, along with customer experiences are all significantly enhanced by accurate vehicle identification and categorization. AVC may be used in a wide range of real-world situations, including intelligent parking systems, autonomous cars, toll payment systems, traffic monitoring systems, and vehicle identification at crime scenes. At transportation hubs, robust AVC techniques are frequently employed, as a result of which identifying vehicle while entering restricted areas can help with security-related difficulties. To train the model for this problem, a large number of real-world vehicle images are needed. Vehicle re-identification, which attempts to match the same vehicle image captured by many cameras, is vital to video surveillance for public safety since vehicles are an indispensable part of human existence. In this instance, the problem of vehicle re-identification demands the highest level of accuracy from AVC approaches. Considering the aforementioned details, in this paper, we have introduced a new still-image based dataset, called JUIVCDv1, for AVC. The following is a list of this paper's main contributions.

1. This dataset offers a realistic image representation of the traffic situation in India, which is very different from that of other developed countries. Vehicle images captured in various scenarios are considered. A total number of 6335 vehicle images can be found in this dataset.
2. Researchers may take this dataset to evaluate the effectiveness of their methods for autonomous vehicle localization and categorization.
3. This dataset includes images of vehicles captured at night time, which makes the categorization task more challenging. As a result, the developed model is also capable of handling data collected in low-light conditions.
4. The vehicle images in the collection are taken in different weather conditions. Therefore, the model is resilient enough to handle data collected in a variety of meteorological scenarios.
5. Detailed annotations are provided for the performance evaluation of either new or existing methods developed on this dataset.
6. Initially, we have executed eight CNN models, namely Xception, InceptionV3, DenseNet121, MobileNetV2, VGG16, NasNetMobile, ResNet50, and ResNet152 on this dataset for AVC. We have also applied three ensemble models on this dataset, namely Majority Voting-based Ensemble (MVE), Weighted Average-based Ensemble (WAE), and Sum Rule-based Ensemble (SRE). Finally, we have benchmarked this dataset using an MVE classifier combination approach which achieves 95% accuracy.

2 Literature review

In this section, we have discussed three aspects of the research problem related to AVC. This section summarizes the AVC datasets that are available publicly for development and validation, performance comparison of different types of AVC methods, and some recent state-of-the-art approaches already available for AVC. During this study, it was found that the datasets related to AVC are quite expensive and the number of freely available AVC datasets is very few. Moreover, the datasets that are freely available to the research community have been used so frequently over the years that these attain an accuracy of almost 100%. Therefore, there is a need for a new dataset with more challenges portraying real-life scenarios. Also, a country like India has a different scenario in this domain due to the road conditions and the presence of multiple vehicles within a single frame. If we use other datasets that are not similar to the Indian road scenario, then the model may not be working properly in real-life conditions. Table 1 provides a summary of the freely available AVC datasets commonly used by researchers to date with state-of-the-art accuracies achieved on these datasets.

To the best of our knowledge, we have gathered the following shortcomings of the existing datasets. The BIT car dataset comprises only 6 classes, and the images in the dataset are captured under a single weather condition, posing a challenge for building a robust AVC model. The online images in the CompCars dataset are gathered from vehicle forums, public websites, and search engines. In contrast, in our developed dataset, images from real-world traffic scenarios are captured. Notably, this dataset includes 12 distinct vehicle classes. The images in our developed dataset are sourced from various weather conditions, including rainy, sunny, and cold seasons, as well as nighttime and daytime settings. We have partitioned the dataset into train and test sets using a 70:30 ratio, which help the future researchers to evaluate their methods. There are many studies related to vehicle detection, but the research articles related to AVC is found to be a few. Maity et al. [26] surveyed this topic in the year 2021 considering all the works done for AVC in the last decade. In this section, we will also discuss some significant AVC methods. Sun et al. [27] proposed a novel vehicle-type classification system, which uses a lightweight CNN with feature optimization and a joint learning strategy. The first step was to create a lightweight CNN with feature optimization, called LWCNN-FO. To minimize the parameters of the network, they employed depth-wise separable convolution. Additionally, the SENet module was included to automatically determine the significance of each channel with features using self-learning. Silva et al. [28] proposed an AVC system with a computer vision solution to solve the problem of the vehicles' make and model classification. By observing the car's attributes and contrasting them with those in the membership, a camera was set up to authenticate the vehicle. They concentrate on constructing a fine-grained AVC system that uses the system's multi-camera composition to fuel a CNN with many views of the vehicle. The evaluations presented indicate that incorporating data from multiple perspectives of a vehicle enhances the classification accuracy of its make and model, particularly in difficult tolling situations. Ni et al. [29] proposed a vehicle attribute recognition system by appearance. The study covered both coarse-grained (vehicle type) and fine-grained (car manufacturer and model) components in its review of existing vehicle component identification methods. This paper aimed to perform vehicle

Table 1 List of freely available image datasets used for AVC till date

Dataset	No. of classes	No. of images	Accuracy (%)
MIO TCD [16]	11	648959	97.95
FG3DCar [17]	30	300	95.30
Stanford Cars [18]	196	16185	96.80
BIT vehicle [19]	6	9850	96.10
BoxCars [20]	27	63750	86.57
CompCars [21]	163	136726	99.00
Poribohon BD [22]	15	9058	98.70
Deshi BD [23]	13	10440	98.00
Frontal-103 [24]	103	65433	91.00
LSUN+Stanford [25]	196	2067710	99.00
Ours (JUIVCDv1)	12	6335	95.00

type recognition by categorizing vehicles into broad classifications based on their sizes or intended usage, such as sedans, buses, and trucks. This study also conducted an analysis of Vehicle Make Recognition, which involves the classification of vehicles based on their respective manufacturers such as Ford, Toyota, and Chevrolet. Silva et al. [30] presented a subscription/membership function for an automated toll-collecting (ATC) system using computer vision. This application system established a one-to-one correspondence between a distinct identifier (ID), a tangible automobile, and a membership. A camera-based system was implemented to authenticate that every transaction aligned with the factual membership data by cross-verifying the vehicle and ID information. The visual system employed various algorithms to extract distinct features of a vehicle such as a number plate, make, model, color, number of axles, and so on. The system performed a comparison between the extracted characteristics and those present in the membership. The authors concentrated on addressing the vehicle's make classification problem and suggested a detailed vehicle classification approach that leverages the system's multi-camera configuration. Sahin et al. [31] proposed the utilization of Light Detection and Ranging (LiDAR) sensor data to differentiate between distinct categories of truck trailers, surpassing the capabilities of conventional classification sensors such as inductive loop detectors and piezoelectric sensors. The present study demonstrates the processing of point-cloud data obtained from a 16-beam LiDAR sensor to extract valuable information and features. The outcomes indicate that the Support Vector Machine (SVM) model can effectively differentiate various caravan body types with a remarkably high degree of precision, ranging from 85% to 98%. Liu et al. [32] proposed a new end-to-end CNN architecture that can simultaneously detect and remove adversarial perturbations by utilizing denoising techniques. This approach was referred to as Denoising Detection and Denoising Adversarial Perturbations (DDAP). The DDAP denoiser utilized the DDAP detector's adversarial examples to eliminate adversarial perturbations. The method being proposed can be considered a pre-processing measure. It does not necessitate any alterations to the configuration of the vehicle classification model and has minimal impact on the classification outcomes of clear images. To validate the capabilities of DDAP, they conducted testing on public datasets such as BIT-Vehicle. Butt et al. [33] presented an AVC system that utilized a CNN to enhance the resilience of vehicle classification in real-time scenarios. The authors provide a dataset of vehicles consisting of 10,000 images that were classified into six distinct vehicle classes. The dataset was designed to account for challenging lighting conditions to enhance the reliability of vehicle classification systems in real time. The study involved fine-tuning pre-trained models such as GoogleNet, Inception-v3, VGG, AlexNet, and ResNet on a self-constructed vehicle dataset to assess their accuracy and convergence capabilities. To achieve generalization, the network was fine-tuned on the VeRi dataset, a publicly available collection of 50,000 images that have been classified into six distinct vehicle categories. Guo et al. [34] proposed a semi-supervised approach for vehicle type classification in Intelligent Transportation Systems (ITS) using broad ensemble learning. The methodology outlined comprised two primary components. The initial phase involves training a set of base Broad Learning System (BLS) classifiers using semi-supervised learning techniques to mitigate the growing burden of unlabeled samples and reduce the duration of the training process. In the second phase, a dynamic ensemble architecture is created using trained classifiers that possess distinct characteristics. The authors utilized the publicly available BIT-Vehicle dataset and MIOTCD dataset to conduct experiments and showcased that their proposed method has better performance in terms of effectiveness and efficiency when compared to a single BLS classifier and other commonly used methods. Mohine et al. [35] introduced a hybrid deep 1D CNN-bidirectional long short-term memory (CNN-BiLSTM) approach that utilized acoustic modality to move vehicle categorization into two-wheeler, low, medium, and heavyweight groups, as well as noise analysis. Furthermore, it underwent testing on the reference dataset, SITEX02, to validate its performance thus achieving an accuracy rate of 96%. The comparative analysis of the 1D CNN-BiLSTM model's performance was conducted against traditional classifiers such as ANN, CNN, SVM, and CNN-LSTM models. Based on the empirical findings, it has been observed that the CNN-BiLSTM model has achieved a superior classification accuracy of 92% in comparison to traditional classifiers.

3 Dataset preparation

In this section, the specifics of creating the JUIVCDv1 dataset have been discussed in detail. Here, we have covered dataset nomenclature, methods of collecting vehicle videos, the process of creating images from videos, and the process of annotations.

3.1 Dataset nomenclature

We have named our developed dataset JUIVCDv1, where JUIVCD stands for 'Jadavpur University Indian Vehicle Classification Dataset'. The dataset has 12 different vehicle classes namely, 'Bicycle', 'Van', 'Car', 'Bus', 'Ambassador, 'Autorickshaw', 'Rickshaw', 'Motorized2Wheeler', 'Motorvan', 'Toto', 'Truck' and 'Minitruck'. Figure 1 illustrates each of the vehicle classes, class names, and class labels.

3.2 Collection of raw data preparation

Images have been collected from highways in Kolkata, an Indian metropolitan city, and some rural locations around Kolkata. We made every effort to compile as many real-time traffic scenarios as possible. Videos are first taken, and then we used labelImg [36] to extract the frames and generate still images. JUIVCDv1 dataset includes images from both fixed positions of the camera as well as from a moving vehicle. We collected data both during the daytime and nighttime. We have also provided bounding boxes of the vehicles in our dataset. On Indian urban streets, the most realistic traffic situations have been adopted.
To capture videos and still images, we mostly used two different camera phones:

1. Redmi Note 9Pro (1280x720p)
2. Honor - HRY-AL00 (1080x2340p)

To make it easier for the image processing algorithms to analyze each video, we have created image frames from each video and saved the images into the JPEG format. The steps of this procedure are discussed below:

1. Specific image frames have been chosen such that they can be easily distinguished from each other in the set of chosen frames, and are not too fuzzy.
2. All still images that have been transformed from videos to image frames have been divided into a training set and a test set. The first 70% image frames are taken into the training set and the remaining is taken into the test set.

3.3 Annotation of processed data

If we want to apply supervised learning algorithms, an accurate annotation is a crucial necessity for any developed dataset. But, sometimes it takes too much time to annotate proper

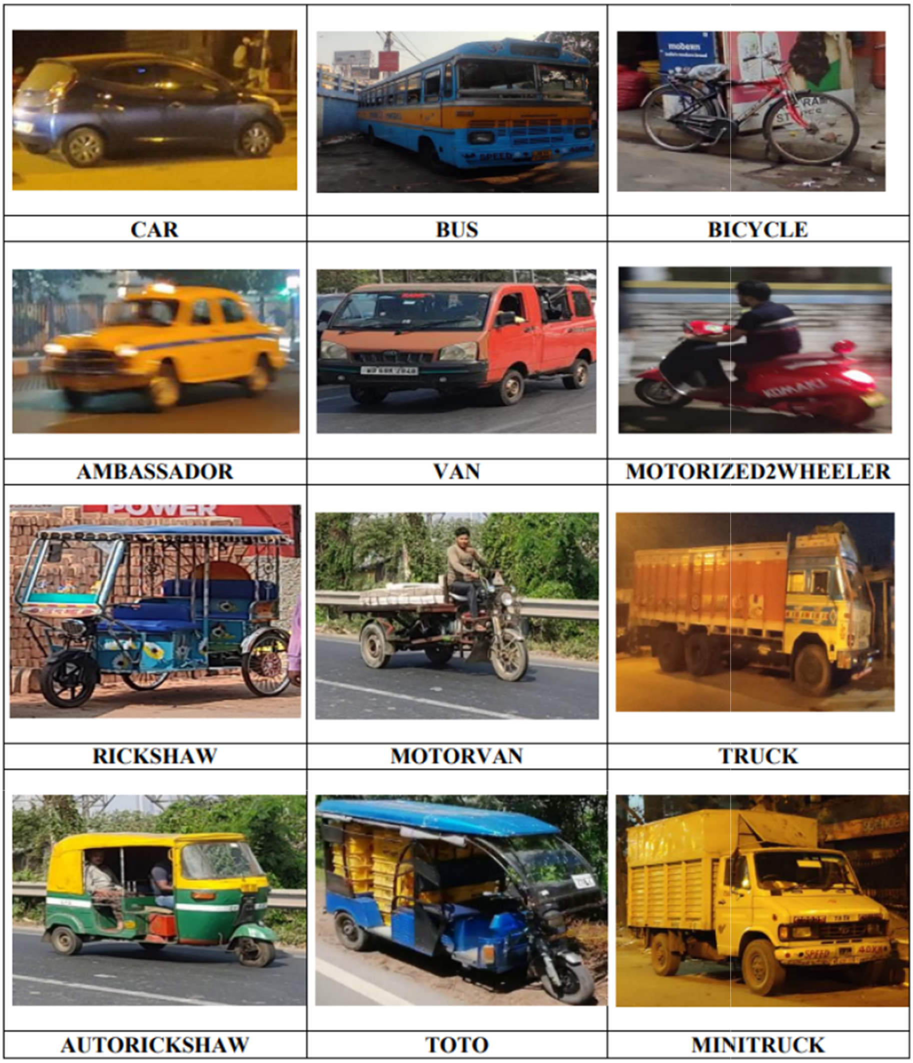

Fig. 1 Sample images of the vehicle classes that are considered in the JUIVCDv1 dataset. (the class labels are denoted by digits before the name of the vehicle classes)

data [37]. Having annotations in the test data is also beneficial for the researchers to evaluate performance while developing a new algorithm. This dataset's annotations are given in both TXT and XML formats. We have used a standard tool, called labelImg tool for the annotation. Figure 2 represents the annotation format of a sample image taken from the JUIVCDv1 dataset. Table 2 shows the annotation on a sample image using the said tool. The bounding boxes of the objects are described as:-bx, by, where the x and y coordinates represent the center of the box. The bh and bw, are the height and width of the bounding boxes respectively relative to the entire image and c represents the class of the object. In TXT format, '0' is defined as a class of the object and the next values are x_center, y_center, width, and height, respectively. In JSON format, the annotation information is represented in the following

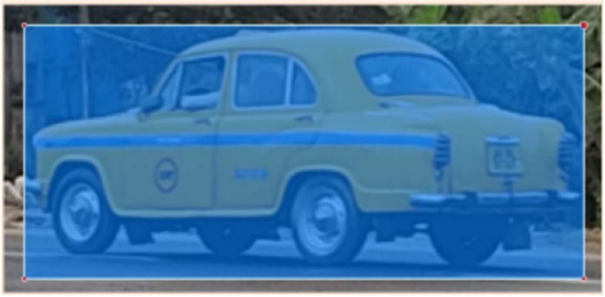

Fig. 2 Annotation of a sample image taken from JUIVCDv1

order: the image name first, then the class of the vehicle, the x and y coordinates of the bounding box, followed by the width and height of the bounding box.

4 Details of JUIVCDv1 dataset

The dataset contains images that can be utilized to create a realistic AVC system focusing mainly on typical Indian road conditions as well as traffic scenarios. The images are taken at various times of the day and night to accommodate every possible diversity of the typical Indian road scenes. Images contain a single object in a single frame. The videos are recorded from the sidewalks on both the sides of the road, and while riding on a moving vehicle. This provides a diversity of images and will help to strengthen the developed models by researchers. To make the model robust and to operate the same in varied situations, the dataset is intentionally kept unbalanced.

4.1 Train set

There are 12 folders in the train set of the JUIVCDv1 dataset. The folders consist of namely ‘0_Car’, ‘1_Bus’, ‘2_Bicycle’, ‘3_Ambassador’, ‘4_Van’, ‘5_Motorized2wheeler’, ‘6_Rickshaw’, ‘7_Motorvan’, ‘8_Truck’, ‘9_Autorickshaw’, ‘10_Toto’, ‘11_MiniTruck’. The ‘0_Car’ folder has 560 number of images, ‘1_Bus’ folder has 560 number of images, ‘2_Bicycle’ folder has 120 images, ‘3_Ambassador’ folder has 480 images, ‘4_Van’, ‘5_Motorized2wheeler’ and ‘6_Rickshaw’folder has 560 number of images, ‘7_Motorvan’ has only 33 images, ‘8_Truck’ has 140 images, ‘9_Autorickshaw’ has 564 images, ‘10_Toto’ has 36 images and ‘11_MiniTruck’ has 181 images. A total of 4300 vehicle images are pro-

Table 2 Annotation format of JUIVCDv1 dataset still image

Annotation format in JSON	Annotation format in TXT
[“Image:“3(106).png"],	
“annotation:[“label":“ambassador_taxi"],	
“coordinates":“x":204.5,“y":167.45,	
“width":395.0,“height":228.0	0 0.508 0.491 0.982 0.670

vided in the training set of the JUIVCDv1 dataset. Sample images are already shown in Fig. 1. The number of objects present in each vehicle class in the training data has been shown in Fig. 3 using a bar graph. Here, the Y-axis is the number of vehicle images, while the X-axis denotes the number of samples present in each vehicle class.

4.2 Test set

similar to the train set, there exist 12 folders in the test set of the JUIVCDv1 dataset. The folders contain namely ‘0_Car’, ‘1_Bus’, ‘2_Bicycle’, ‘3_Ambassador’, ‘4_Van’, ‘5_Motorized2wheeler’, ‘6_Rickshaw’, ‘7_Motorvan’, ‘8_Truck’, ‘9_Autorickshaw’, ‘10_Toto’, ‘11_MiniTruck’. In ‘0_Car’ folder, there are 240 number of images, ‘1_Bus’ folder has 240 number of images, ‘2_Bicycle’ folder has 80 images, ‘3_Ambassador’ folder has 320 images, ‘4_Van’, ‘5_Motorized2wheeler’ and ‘6_Rickshaw’folders have 240 number of images, ‘7_Motorvan’ has only 11 images, ‘8_Truck’ have 59 images, ‘9_Autorickshaw’ has 240 images, ‘10_Toto’ has 23 images and ‘11_MiniTruck’ has 122 images. A total of 2035 vehicle images are given in the test data of the JUIVCDv1 dataset. Sample images are already shown in Fig. 1. The number of objects present in each vehicle class in the test data is shown as a bar graph in Fig. 4. Here, the Y-axis represents the number of images, while the X-axis represents the number of items in a vehicle class.

5 Benchmarking JUIVCDv1 dataset

To categorize the automobiles in our dataset, we have also considered some state-of-the-art pre-trained deep learning models.

5.1 Xception

Chollet et al. [38] proposed a CNN model based on depth-wise separable convolution layers. They assert that it is possible to completely dissociate the cross-channel mapping and feature maps of spatial correlations. This hypothesis is the extreme version of Inception architecture. For this reason, the authors proposed the architectural name Xception, which means Extreme Inception. The Xception architecture consists of 36 convolutional layers for feature extraction. The Xception architecture is just a linear stack of residually connected depth-separable convolution layers. Figure 5 shows the architecture of the Xception model.

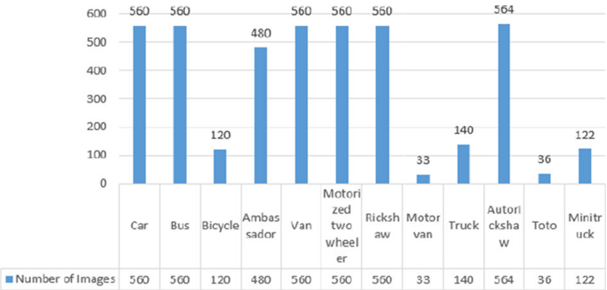

Fig. 3 Bar graph showing the number of images present per vehicle class in the train set of the JUIVCDv1 dataset

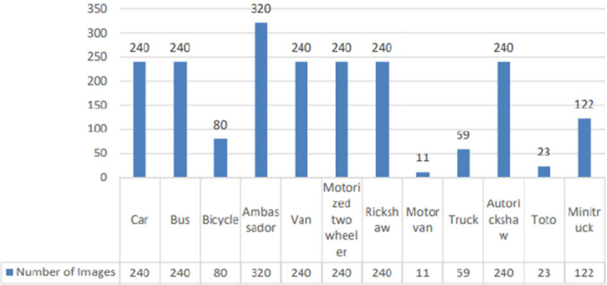

Fig. 4 Bar graph showing the number of images present per vehicle class in the test set of the JUIVCDv1 dataset

5.2 InceptionV3

In 2016, Szegedy et al. [39] proposed a novel model for classification, called InceptionV3. The InceptionV3 is a CNN model belonging to the Inception family. It incorporates various enhancements such as the utilization of label smoothing, factorized 7x7 convolutions, and an auxiliary classifier to disseminate label information to lower network layers. Additionally, batch normalization is employed for layers in the side head. In Fig. 6, the overall architecture of the InceptionV3 model is shown.

5.3 DenseNet121

DenseNet, [40] is a CNN architecture, which was recently presented, with an intriguing connection pattern. DenseNet architecture connects layers with a dense block, promoting feature

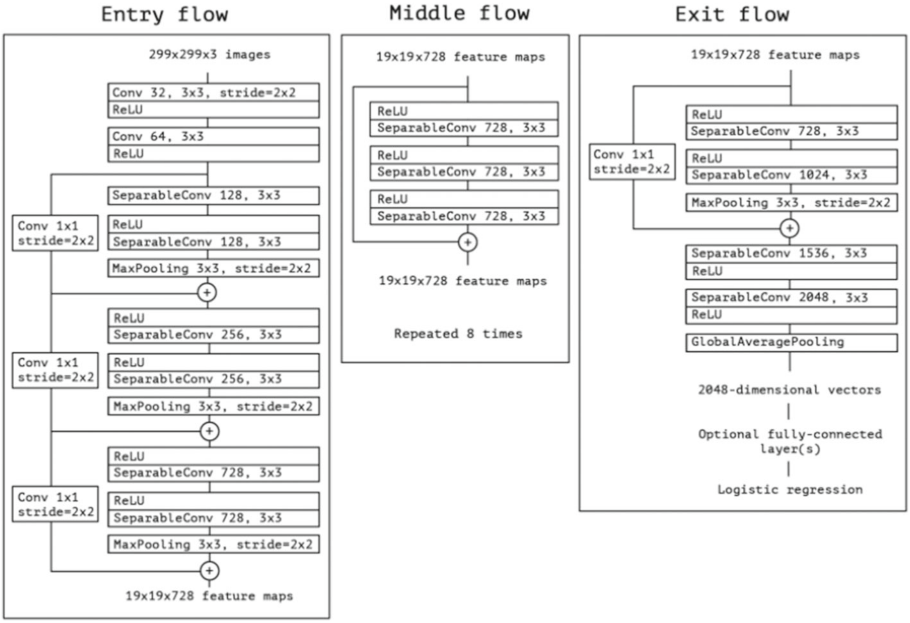

Fig. 5 Architecture of the Xception model

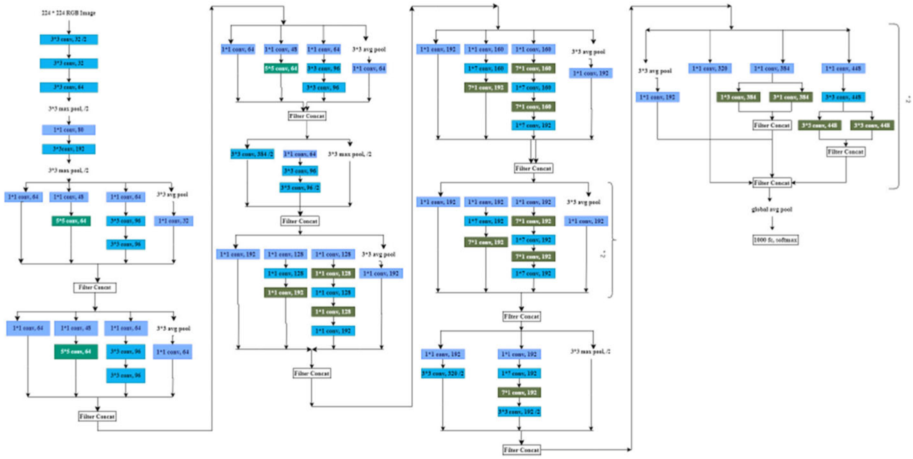

Fig. 6 Architecture of the InceptionV3 model [39]

reuse and reduced overfitting. Each layer accesses predecessor's feature maps, reducing overfitting. Direct supervision from the loss function and shortcut pathways contribute to implicit deep supervision. This results in a dense model, reducing overfitting and improving computational and memory efficiencies. Concatenation of layers enhances network compactness and growth rate, reducing channel count. In Fig. 7, the block diagram of the DenseNet model is shown.

5.4 MobileNetV2

An effective model for mobile and embedded vision applications is provided in MobileNet [41]. It is a simplified architecture that builds lightweight deep CNN models using depth-wise separable convolutions. Each input channel of MobileNet receives a single filter applied to depth-wise convolution. Two layers, one for combining and one for filtering, are segregated from this by the depth-wise separable convolution. The result of this factorization is a significant decrease in computation and model size. Modern object detection systems can potentially use MobileNet as an efficient base network. The schematic diagram of the MobileNet model is shown in Fig. 8.

5.5 VGG16

In 2014, the Visual Geometry Group (VGG) at the University of Oxford developed VGG16 [42], which is a popular CNN model. It consists of a total of 16 layers, including 16 layers of

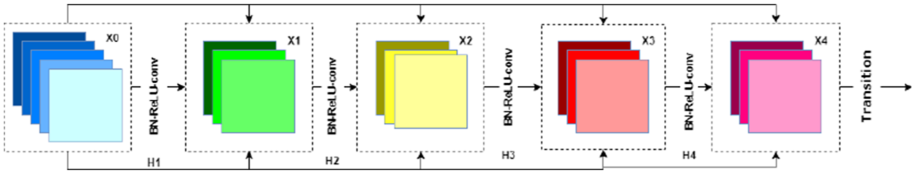

Fig. 7 Architecture of the DenseNet121 model

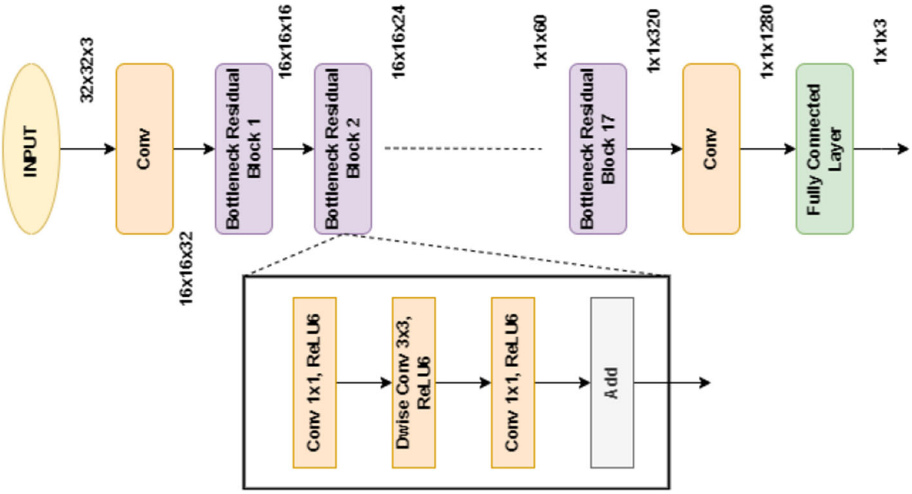

Fig. 8 Architecture of the MobileNetV2 model

convolutional processing, and three levels of fully linked processing. In the initial layers of the network, convolutional layers consisting of 3*3 filters have been used. The convolutional layers are then followed by max-pooling layers with 2*2 filters. These 2*2 filters can reduce the spatial size of the output of the convolutional layers in half. Of almost 16 convolutional layers, the first 13 layers employ 3*3 filters, and the remaining three use 1*1 filters. When smaller filters are used, a deeper network can be constructed using fewer parameters. The first convolutional layer starts with 64 filters and then works its way up to 512 filters in the final layer. Each of the fully linked layers that make up the final stage of the network consists of 4096 neurons. A Softmax layer serves as the last layer of the network and is responsible for producing the class probabilities. The architecture of the VGG16 model is represented in Fig. 9.

5.6 ResNet50

A well-known CNN model, called ResNet-50 [43], a member of the ResNet (Residual Network) family, was introduced by He and colleagues. This model uses a standard input picture size of 224 by 224 pixels. In this model, a max-pooling layer comes after the first layer, which is a typical convolutional layer. ResNet-50 is made up of four stages and sixteen residual blocks. The number of residual blocks changes from stage to stage, as does the number of filters inside each block. The architecture of the Resnet50 model is presented in Fig. 10.

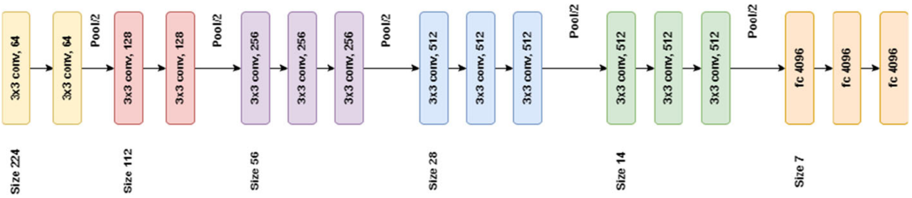

Fig. 9 Architecture of the VGG16 model

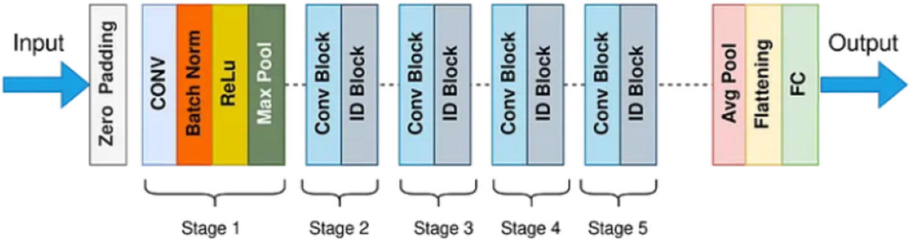

Fig. 10 Architecture of the Resnet50 model

5.7 NasNetMobile

NasNet is a scalable CNN architecture (built for neural architecture search), which consists of fundamental building blocks refined through reinforcement learning. It was trained on over a million of images from the ImageNet database [44]. A cell is made up of only a few processes (a few separable convolutions and pooling) and is repeated several times to meet the network's capacity requirements. NasNetMobile is a mobile version, which consists of 12 cells with 5.3 million parameters and 564 million multiply-accumulates (MACs). The element-wise addition method is used by NASNet, which is far more intuitive than vector-wise operations. When utilizing a feature map as an input, two types of convolutional cells are employed. The input picture size of the network is 224*224. The architecture of the NasNetMobile model is presented in Fig. 11.

5.8 Majority voting-based ensemble

The MVE is one of the most popular and commonly used classifier combination approaches. In a majority voting rule [46–48], each classifier casts a vote for one class, and the class with the maximum number of votes wins. In terms of statistics, the target label anticipated by the ensemble is the mode of distribution of the individual predictions of labels. For example, suppose three classifiers are used in the ensemble (C1, C2, and C3), and the class labels are A and B. If both C1 and C2 predict the result as A, and C3 predicts the result as B, according to the MVE approach, the result will be A. The voting procedures are based on

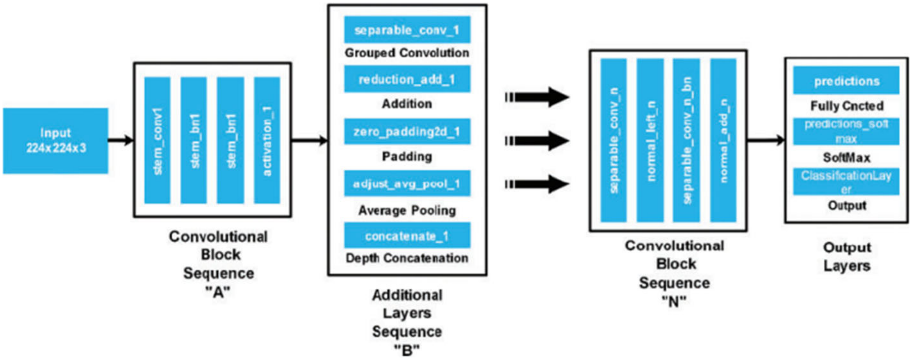

Fig. 11 Architecture of the NasNetMobile model [45]

a democratic (weighted) mechanism that aggregates the forecasts from the categorization models. These categorization models have been separately calibrated using many analytical sources. The simplest and most natural technique relies on the MVE rule, which designates a sample based on the most common class assignment (the "loose" method). In the event of a tie, the sample is not categorized. When all of the models under consideration have completed the forecast agreement, voting by strict majority depicts this situation. We have opted for the MVE strategy since it generally yields superior results [46]. For easy understanding, a pictorial illustration of the MVE approach is shown in Fig. 12.

5.9 Sum rule-based ensemble

In machine learning [49], one popular idea related to ensemble learning is the SRE. In ensemble learning, many models are combined to produce a prediction model that itself is more powerful and reliable than any of the individual models. In particular, the term "SRE" describes a technique that adds predictions from several models by adding up each one's unique forecasts. Next, a summary of each model's predictions is provided. In classification tasks, weights may be provided to the class labels, and these weights are taken into account while calculating the total. The total is added together and is used to determine the final projection. When it comes to classification challenges, the ensemble prediction might be the class with the largest sum or weighted sum.

5.10 Weighted average-based ensemble

The WAE [50] is one of the most popular and commonly used classifier combination approaches. It combines the predictions of several models to increase prediction accuracy. This approach creates a weighted average of the predictions by giving weights to various models according to their performance on a validation set. For easy understanding, a pictorial illustration of the WAE approach is shown in Fig. 13.

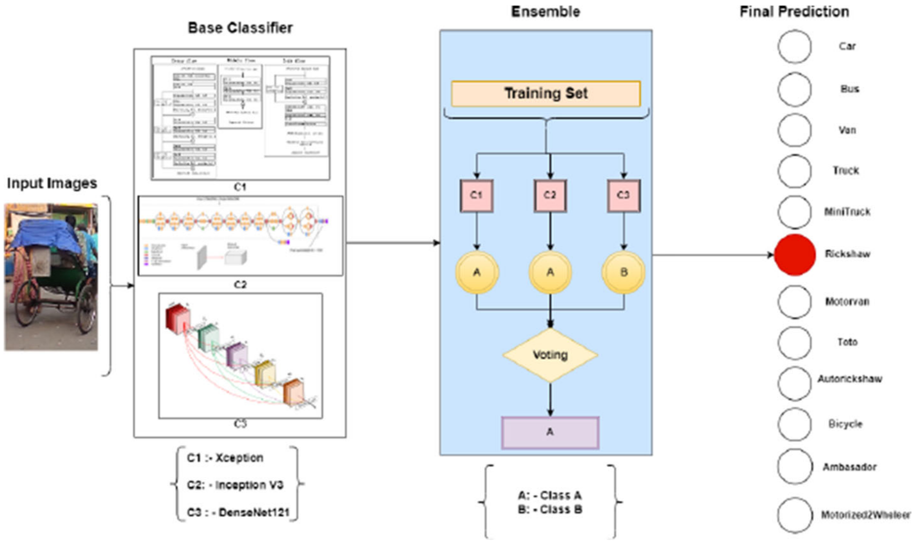

Fig. 12 A pictorial illustration of the MVE approach used to benchmark the JUIVCDv1 dataset

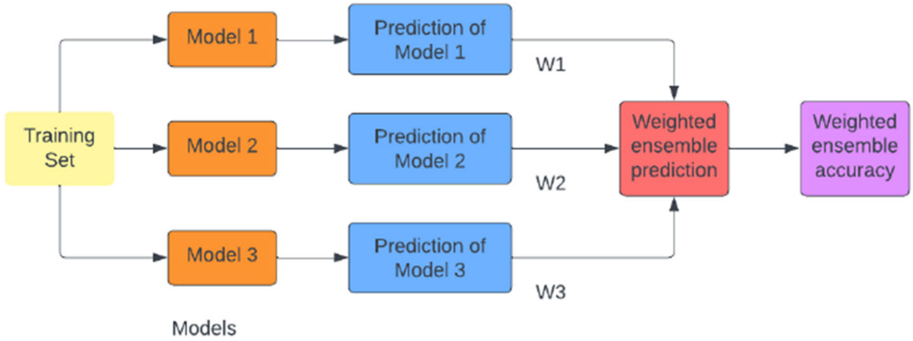

Fig. 13 A pictorial illustration of the WAE approach used to benchmark the JUIVCDv1 dataset

6 Results and discussion

The present work includes the development of a still image dataset called JUIVCDv1 for vehicle classification and benchmarking the results on the same. We have trained and tested eight different CNN models namely Xception, InceptionV3, DenseNet121, MobileNetV2, VGG16, NasNetmobile, ResNet50, and Resnet152 on our developed dataset. Finally, three popular classifier combination approaches such as MVE, SRE and WAE are used to enhance the overall classification performance. In the following subsections, we have discussed the results obtained with their corresponding analysis.

6.1 Model evaluation

Some standard evaluation metrics such as classification accuracy, precision [51], recall [51], F1-score [52], and confusion matrices [53] are used to measure the performance of the CNN-based models on our developed dataset. These evaluation metrics are previously defined in [9]. We have achieved a 0.94 accuracy score on the Xception model, while InceptionV3 has achieved an accuracy score of 0.93, DenseNet121 and MobileNetV2 models have achieved accuracy scores of 0.92 and 0.90 respectively, NasNetmobile have achieved 0.88, the VGG16 model has achieved 0.85 accuracy score and ResNet50 have achieved 0.56 and Resnet152 have achieved 0.55. After, analyzing the results, three base CNN models, namely Xception, InceptionV3, and DenseNet121 are chosen for the MVE, SRE, and WAE, techniques due to their performance for the said task. The MVE method has achieved an accuracy score of 0.95, whereas both the WAE and SRE approaches attained an accuracy score of 0.94.

6.2 Results obtained by CNN models

The outcomes of the eight pre-trained CNN models as well as three popular ensemble approaches have been examined in this section. The outcomes of the base learners have been displayed graphically. When it comes to identifying certain vehicle classes, some models have been observed to be more accurate than others. A graphical comparison of the test accuracies provided by eight pre-trained CNN models along with three ensemble approaches is shown in Fig. 14. In addition, a report detailing the accuracies of the classification scores of the three best performing CNN models has been provided as a classification report, and the confusion matrices for the three ensemble models are also provided for observing the accurate and inaccurate classifications made by each of the models.

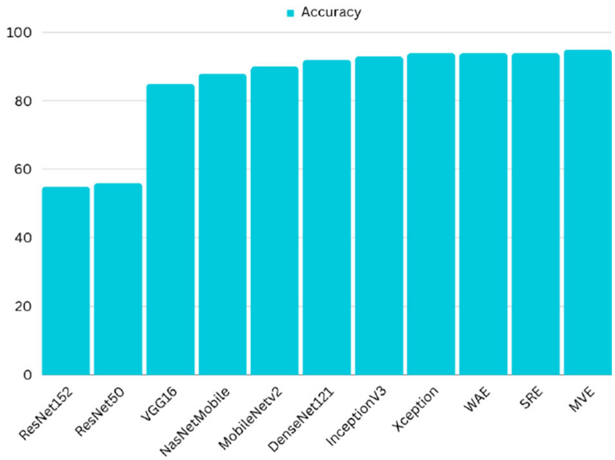

Fig. 14 Performance comparison of test accuracies produced by eight pre-trained CNN models along with three ensemble techniques used for AVC on the proposed JUIVCDv1 dataset

6.3 Performance comparison: classification report

The performance of AVC on the test set of the JUIVCDv1 dataset using the Xception model has been presented in Table 3. After evaluating Xception on our dataset, we have observed that the highest precision value of 0.99 is achieved by the class 4_Van and the class namely 5_Motorized2wheeler, and the lowest precision value of 0.69 achieved by 10_Toto. The class 7_Motorvan has achieved the highest recall value of 1.00 and the class 1_Bus has the lowest recall value of 0.57. The highest F1-Score has been achieved by the model for two classes, namely 5_Motorvan and 8_Truck i.e., 0.98. The Xception model has achieved an overall accuracy of 0.94.

In Table 4, we provide how AVC performs using the InceptionV3 model on the JUIVCDv1 test set. After analyzing InceptionV3 using our dataset, we found that class 2_Bicycle had

Table 3 Classification report showing the values of Precision, Recall, and F1-score for each of the vehicle classes produced by the Xception model on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-score
0_Car	0.86	0.98	0.92
1_Bus	0.87	0.57	0.68
2_Bicycle	0.88	0.75	0.81
3_Ambassador	0.97	0.95	0.96
4_Van	0.99	0.90	0.94
5_Motorized2wheeler	0.99	0.96	0.97
6_Rickshaw	0.94	0.90	0.92
7_Motorvan	0.96	1.00	0.98
8_Truck	0.98	0.98	0.98
9_Autorickshaw	0.77	0.91	0.83
10_Toto	0.69	0.78	0.73
11_Minitruck	0.97	0.98	0.98

(Accuracy score: 0.94)

Table 4 Classification report showing the values of Precision, Recall, and F1-score for each of the vehicle classes produced by the InceptionV3 model on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-Score
0_Car	0.86	1.00	0.92
1_Bus	0.96	0.95	0.95
2_Bicycle	1.00	0.88	0.93
3_Ambassador	0.99	0.97	0.98
4_Van	0.97	0.90	0.93
5_Motorized2wheeler	0.98	0.97	0.98
6_Rickshaw	0.95	0.98	0.97
7_Motorvan	0.58	1.00	0.73
8_Truck	0.60	0.95	0.74
9_Autorickshaw	0.97	0.97	0.97
10_Toto	0.73	0.48	0.58
11_Minitruck	0.94	0.61	0.74

(Accuracy score: 0.93)

the model's greatest precision value of 1.00, while class 7_Motorvan had the model's lowest precision value of 0.58. The model for two classes, 0_Car and 7_Motorvan, has the maximum recall value of 1.00, and 10_Toto has the lowest Recall value of 0.48. The model has the highest F1-Score of 0.98 for the two classes of 3_Ambassador and 5_Motorized2wheeler. The total accuracy score of the InceptionV3 model is 0.93.

The effectiveness of AVC utilizing the DenseNet121 model on the JUIVCDv1 test set is shown in Table 5. The model attained the highest precision of 1.00 for the classes 3_Ambassador and 7_Motorvan, and the lowest precision of 0.57 for the class 8_Truck, according to our evaluation of DenseNet121 on our dataset. In the model, the recall value of 0.43 for the class 10_Toto is the lowest, and the classes "0_Car" and "5_Motorized2wheeler" have the maximum recall value of 1.00. The model for class 5_Motorized2wheeler has the highest F1-Score, which is 0.98. An accuracy score of 0.92 is attained with the DenseNet121 model.

Table 5 Classification report representing the results of Precision, Recall, and F1-score for each of the vehicle classes produced by the DenseNet121 model on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-score
0_Car	0.88	1.00	0.93
1_Bus	0.91	0.94	0.93
2_Bicycle	0.93	0.84	0.88
3_Ambassador	1.00	0.94	0.97
4_Van	0.95	0.88	0.92
5_Motorized2wheeler	0.97	1.00	0.98
6_Rickshaw	0.95	0.97	0.96
7_Motorvan	1.00	0.91	0.95
8_Truck	0.57	0.93	0.71
9_Autorickshaw	0.95	0.95	0.95
10_Toto	0.67	0.43	0.53
11_Minitruck	0.86	0.61	0.72

(Accuracy score: 0.92)

Table 6 Classification report showing the values of Precision, Recall, and F1-score for each of the vehicle classes produced by the MVE approach on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-score
0_Car	0.92	1.00	0.96
1_Bus	0.88	0.61	0.72
2_Bicycle	0.93	0.70	0.80
3_Ambassador	0.98	0.97	0.97
4_Van	1.00	0.86	0.93
5_Motorized2wheeler	0.99	0.96	0.98
6_Rickshaw	0.97	0.96	0.96
7_Motorvan	0.98	1.00	0.99
8_Truck	0.96	0.98	0.97
9_Autorickshaw	0.98	1.00	0.99
10_Toto	0.67	0.88	0.76
11_Minitruck	0.96	1.00	0.98

(Accuracy score: 0.95)

6.4 Results obtained by ensemble approaches

This section discusses the results provided by three popular state-of-the-art ensemble methods MVE, SRE and WAE. In Table 6, we show the performance of AVC using the MVE model on the JUIVCDv1 test set. We have chosen three CNN models for the majority voting technique since they are more accurate than the other CNN models utilized in this case: Xception, InceptionV3, and DenseNet121. After analyzing our dataset, we have found that the model attained the maximum accuracy value of 1.00 for class 4_Van and the lowest precision value of 0.48 for class 9_Autorickshaw. Class 0_Car has the model's highest recall value of 1.00, while class 1_Bus has the lowest recall value of 0.61. The class 7_Motorvan model has the highest F1-Score, which is 0.99. The final accuracy of the MVE technique is 0.95.

In Table 7, we have demonstrated the performance of AVC on the JUIVCDv1 test set using the SRE model. We have chosen three CNN models, Xception, InceptionV3, and

Table 7 Classification report showing the values of Precision, Recall, and F1-score for each of the vehicle classes produced by the SRE approach on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-score
0_Car	0.90	1.00	0.95
1_Bus	0.97	0.96	0.97
2_Bicycle	1.00	0.84	0.91
3_Ambassador	0.99	0.96	0.97
4_Van	0.94	0.95	0.95
5_Motorized2wheeler	0.98	1.00	0.99
6_Rickshaw	0.96	0.98	0.97
7_Motorvan	0.73	1.00	0.85
8_Truck	0.68	0.85	0.75
9_Autorickshaw	0.97	0.98	0.98
10_Toto	0.68	0.85	0.75
11_Minitruck	0.89	0.70	0.78

(Accuracy score: 0.94)

Table 8 Classification report showing the values of Precision, Recall, and F1-score for each of the vehicle classes produced by the WAE approach on the JUIVCDv1 dataset

Vehicle class	Precision	Recall	F1-score
0_Car	0.90	1.00	0.95
1_Bus	0.97	0.96	0.97
2_Bicycle	1.00	0.84	0.91
3_Ambassador	0.99	0.96	0.97
4_Van	0.94	0.95	0.95
5_Motorized2wheeler	0.98	1.00	0.99
6_Rickshaw	0.96	0.98	0.97
7_Motorvan	0.73	1.00	0.85
8_Truck	0.68	0.85	0.75
9_Autorickshaw	0.97	0.98	0.98
10_Toto	0.80	0.52	0.63
11_Minitruck	0.89	0.70	0.63

(Accuracy score: 0.94)

DenseNet121, for the WAE approach since they are more accurate than the other CNN models used in this example. We have discovered that the model achieved the highest accuracy value of 1.00 for class 2_Bicycle and the lowest precision value of 0.68 for class 8_Truck after evaluating our dataset. The model's best recall value is 1.00 for classes Class 0_Car, 7_Motorvan, and 5_Motorized2wheeler, while the lowest recall value is 0.52 for class 10_Toto. The model in class 5_Motorized2wheeler has the highest F1-Score, which is 0.99. The SRE technique has a total accuracy of 0.94.

In Table 8, we have shown the performance of AVC using the WAE model on the JUIVCDv1 test set. We have chosen the best three performing CNN models, namely Xception, InceptionV3, and DenseNet121, for the WAE technique since they are more accurate than the other CNN models that are utilized in this case. After analyzing our dataset, we have found that the model attained the maximum precision value of 1.00 for class 2_Bicycle and the lowest precision value of 0.68 for class 8_Truck. Class 0_Car, 7_Motorvan, and 5_Motorized2wheeler have the model's highest recall value of 1.00, while class 10_Toto has the lowest recall value of 0.52. The class 5_Motorized2wheeler model has the highest F1-Score, which is 0.99. The total accuracy of the WAE technique is 0.94.

6.5 Performance comparison: accuracy vs epoch

In this section, curves related to training and validation accuracies of eight different base models have been shown. In Fig. 15, the training accuracy (TA) and validation accuracy (VA)

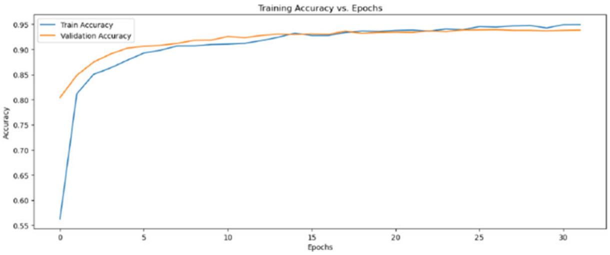

Fig. 15 Training accuracy and validation accuracy curve achieved through Xception model

curves of the Xception model are given concerning the number of epochs, whereas Fig. 16 denotes the same for the InceptionV3 model. After analyzing base CNN models on JUIV-CDv1 dataset, we have found that the Xception model achieves the best accuracy in Indian road scenario, and it shows a accuracy of 0.94 where InceptionV3 and DenseNet121 achieves the accuracy score of 0.93 and 0.92 respectively. Finally, Fig. 17 shows the DenseNet121 model's TA and VA curves vs the number of epochs.

6.6 Confusion matrices

In Fig. 18(a) shows the confusion matrix for the MVE approach model, Fig. 18(b) shows the confusion matrix for the SRE approach model and Fig. 18(c) shows the confusion matrix for the WAE approach model. In Fig. 19, a sample of the class 10_Toto is wrongly classified as a sample of the class 7_Motorvan, whereas in Fig. 20, a sample of the class 11_Minitruck is wrongly classified as a sample of the class 0_Car. This wrong classification may be attributed to the fact that for some images, there are intra-class similarities among the sample images belonging to different vehicle classes. Moreover, as there are very few still images of 10_Toto in our collected data, and 10_Toto as well as 6_Rickshaw have a significant amount of similarity in their appearances, several samples of 10_Toto have been incorrectly designated by the models. Adding more samples of 10_Toto to the training set will help the models learn better to distinguish 6_Rickshaw and 10_Toto, and to address the problem. Again, the images of the front side of 11_Minitruck are quite similar in appearance to some of the 0_Car. This also leads to the wrong classification of the 11_Minitruck images.

6.7 Data visualization

In the year 2019, Selvaraju et al. [54] proposed a visual explanation algorithm namely Gradient-Weighted Class Activation Mapping (Grad-CAM), that creates a coarse localization map, which highlights the significant areas in the image for prediction/classification by using the gradients of any target concept. Grad-CAM [55] may be used with several CNN models, like VGG which has fully connected layers, visual question-answering CNNs for multimodal tasks, or CNNs for reinforcement learning. Grad-CAM may be seen as one of the first steps in the bigger scheme of interpretable or explainable AI since the visualizations provide insights into failure and aid in the identification of bias while surpassing previous standards. The backpropagation technique's issues with upsampling and downsampling relevance maps to create coarse relevance heatmaps are also successfully avoided by this extension of the CAM algorithm. Figure 21 shows the Grad-CAMs generated on some sample vehicle images by

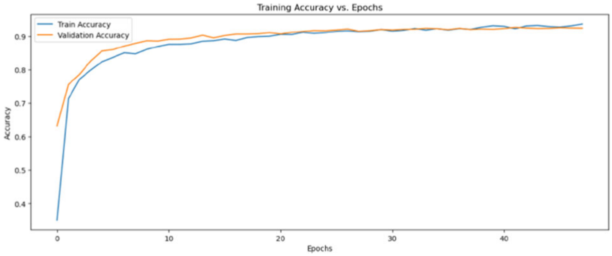

Fig. 16 Training accuracy and validation accuracy curve achieved through Xception model

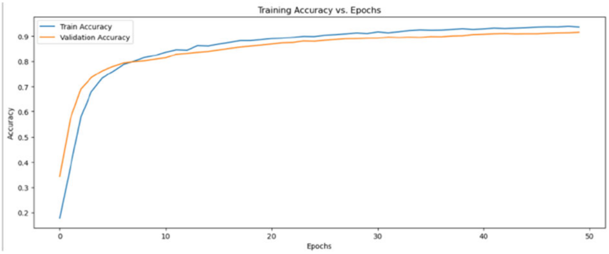

Fig. 17 Training accuracy and validation accuracy curve achieved through Xception model

best five different CNN models used in the present work. In Fig. 21, the second and third columns depict the original vehicle image and Grad-CAMs generated by Xception, InceptionV3, DenseNet121, MobileNetV2 and NasNetMobile models respectively.

6.8 Results on poribohonBD dataset

Tabassum et al. [56] created the PoribohonBD dataset for vehicle categorization based on vehicle images of Bangladesh. Sample vehicle images of this dataset are accessible at: https://data.mendeley.com/datasets/pwyyg8zmk5/2here. We have considered this dataset for experimentation since both the countries, India and Bangladesh have very similar road scenarios. Images of vehicles are collected from two sources: a) social media and b) smartphone cameras. The collection includes 9058 tagged and annotated photos of 15 native vehicles, that are commonly found on roads of Bangladesh, including bus, three-wheeler rickshaw, motorcycle, truck, and wheelbarrow. In this dataset, data augmentation techniques are also used to maintain the amount of images comparable for each type of vehicle. Initially, we have chosen eight base CNN models for the primary study. After analyzing the results, we have observed that the DenseNet121 model attains the maximum accuracy value of 0.94, the ResNet152v2 model attains an accuracy of 0.91, the MobileNet model attains an accuracy of 0.90, the Xception model attains an accuracy of 0.88, InceptionV3 model attains the accuracy of 0.85, VGG16 model attains the accuracy of 0.67, ResNet50 model attains the accuracy of 0.65 and NASNetMobile model attains the accuracy of 0.60. Among them, three best performing base CNN models such as Xception, InceptionV3, and DenseNet121 are chosen for implementing the MVE, WAE, and SRE techniques. After considering the ensemble methods, the MVE approach has achieved the highest accuracy score of 0.96, the SRE method has attained an

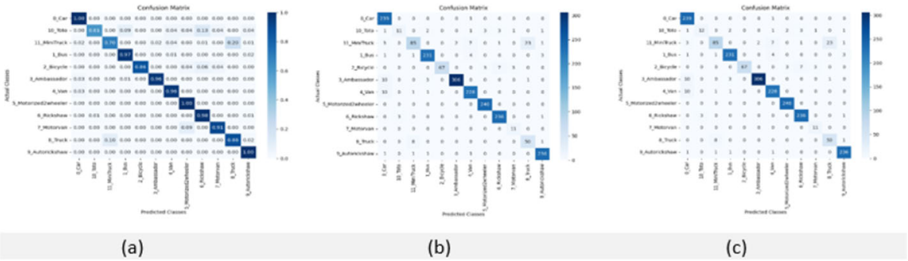

Fig. 18 Confusion matrices produced by MVE, SRE, and WAE techniques on our developed JUIVCDv1 dataset

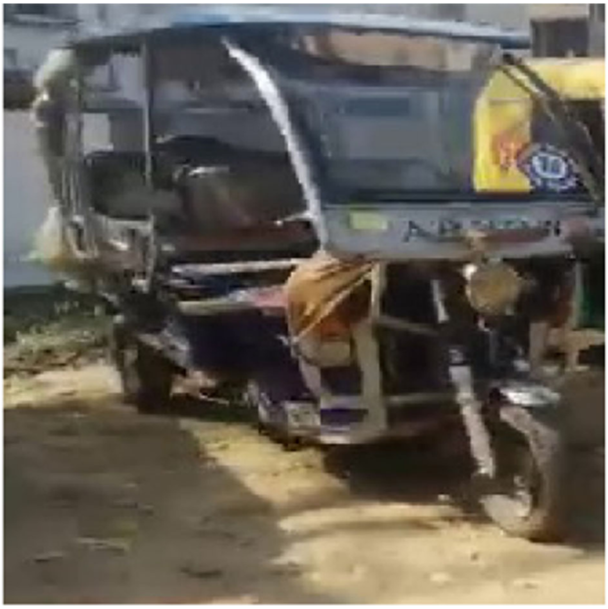

Fig. 19 Missclassified Image 1

accuracy score of 0.95, and the WAE method has attained an accuracy score of 0.95. Table 9 presents the performance of AVC given by both base CNN models as well as three ensemble models on the PoribohonBD dataset.

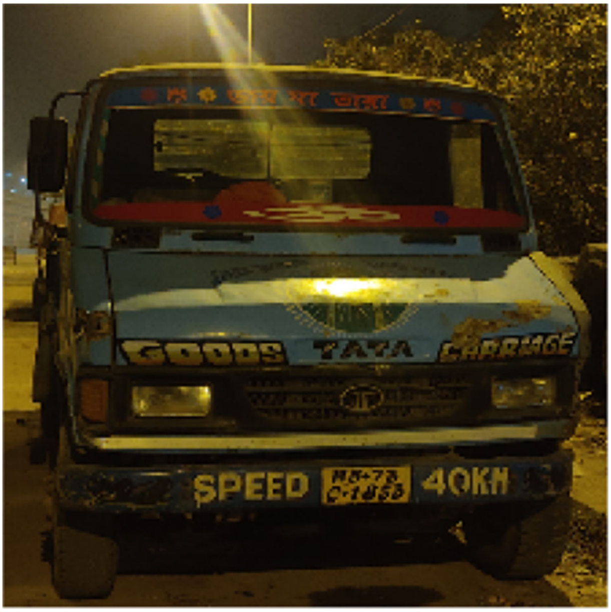

Fig. 20 Missclassified Image 2

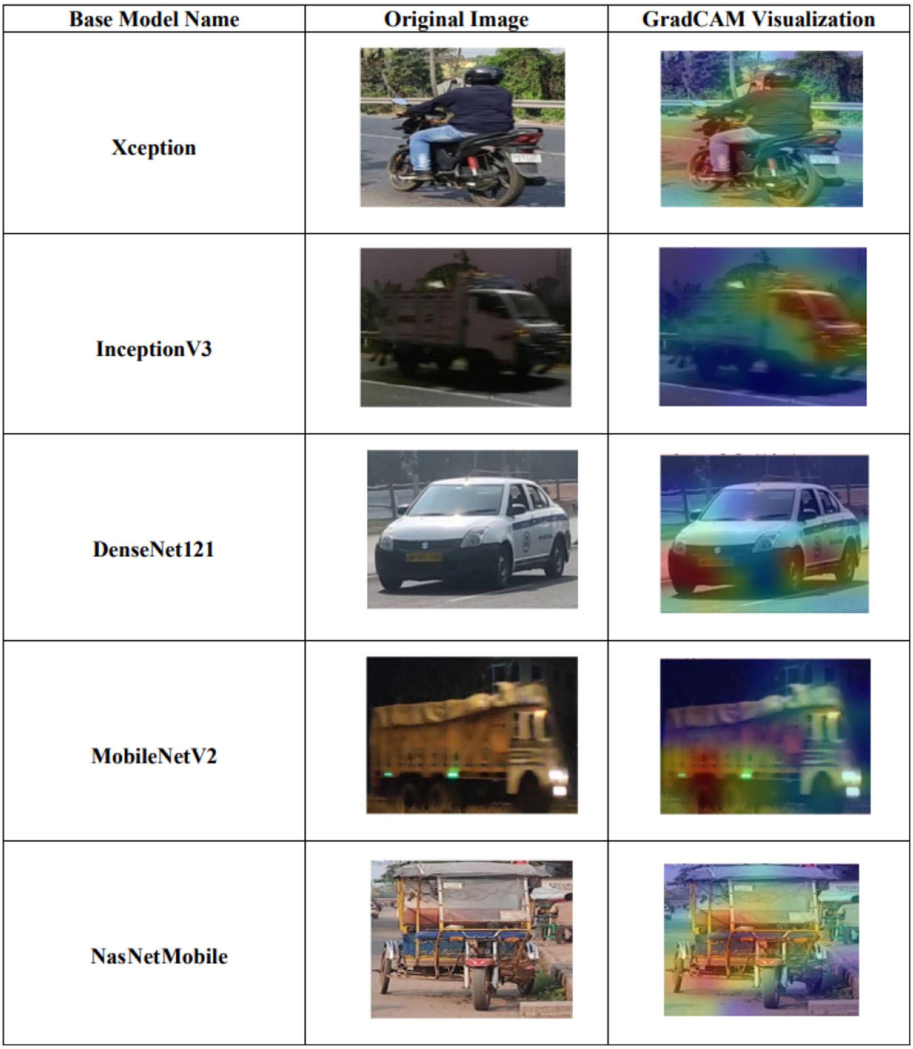

Fig. 21 GradCAM-based data visualization using best five base CNN models on the proposed JUIVCDv1 dataset

6.9 Limitations of JUIVCDv1 Dataset

Further analysis of the results gives some ideas about the complexities of the developed dataset. It would also help future researchers to work on this dataset and develop more advanced methods to deal with associated problems. Some major issues are as follows:

- The training set of the dataset has a class imbalance, which is a major problem. A large proportion of the still images in this set include cars and motorcycles, whereas only a small fraction contains images of totes and bicycles. We have employed data augmentation techniques for such situations, where the samples are comparatively less.

Table 9 Classification report showing the values of Precision, Recall and F1-score for eight base CNN models and three ensemble models applied on the PoribohonBD dataset

Model	Accuracy	F1-score	Recall
DenseNet121	0.95	0.95	0.94
ResNet152	0.56	0.55	0.55
ResNet50	0.57	0.56	0.56
MobileNet	0.90	0.90	0.90
Xception	0.88	0.88	0.88
InceptionV3	0.87	0.85	0.85
VGG16	0.68	0.67	0.67
NasNetMobile	0.60	0.60	0.60
MVE	**0.97**	**0.96**	**0.96**
WAE	0.95	0.95	0.95
SRE	0.95	0.95	0.95

Bold marks are used to highlight highest scores

- There are 33 Motorvan in the training set, which may not be sufficient for the models to accurately learn about the vehicle classes. The 11 vehicles in the validation set allow for a more precise model and easier vehicle classification. However, adequate data is not available for the CNN models to properly learn bicycle class images. With less inequality across classes in terms of the number of images, we may have observed far better classification accuracy.
- In our dataset, several Totos are misclassified due to the scarcity of a significant amount of sample images in the dataset.
- All of the roads in nations like India, Bangladesh, or Pakistan are not as good as those in the developed nations of either Europe or America. For the former case, traffic congestion as well as breaking of traffic rules are quite common in these countries. These issues add inherent complexities to the image quality.
- Sometimes, it becomes difficult to precisely characterize the vehicles in images taken in various weather conditions, such as those taken at night when illumination is compromised or those taken in rainy conditions.

7 Conclusion

Nowadays, there is a large number of vehicles on the roads, and hence the need for AVC systems has become more significant for managing real-time traffic, especially in highly populated cities. A realistic image/video dataset portraying a traffic condition is essential for this purpose. Researchers may utilize this dataset to assess the efficiencies of their approaches for both automated localization and classification of vehicles. Though there are plenty of datasets available for vehicle localization, only a few of them can be used for the classification task. Additionally, very few available datasets can accurately reflect real-world situations. For example, images taken on the Indian subcontinent, frequently show two or more vehicles overlapping in a single frame leading to traffic congestion. Therefore, researchers have faced difficulties in using this information because of the distinctive features of Indian roads, such as the high volume of traffic, clogged highways, the poor state of the roads, and traffic congestion. In this study, we have created an image dataset suitable for AVC keeping Indian roads in mind to overcome this research vacuum. We have included the necessary annotation for the assessment of the AVC algorithms. This dataset is free to use for the research communities only. We have used eight deep learning models, namely Xception, DenseNet121, InceptionV3, MobileNetV2, VGG16, NasNetmobile, ResNet50, and Resnet152 for benchmarking our dataset. Additionally, we have applied three popular state-of-the-art ensemble models such as MVE, WAE, and SRE to enhance the accuracy of the developed dataset. We have achieved the best accuracy of 0.95 by using the MVE approach, which is satisfactory given the complexity of the images.

7.1 Future scope

- Version 1 of the dataset has about 6k images, which might not be sufficient to train CNN models properly. Therefore, we would like to continue gathering images or videos for the upgradation of the dataset.
- We will take steps in the future to maintain an equivalent number of data in each vehicle class by collecting more images for classes such as totos, vans, and rickshaws.
- We are planning to collect images in various weather conditions including foggy, nighttime, rainy, etc.
- We will attempt to include more vehicle classes that are commonly found on Indian roads.
- Multi-view or multimodal datasets are not available for the classification of vehicles. Lots of research and data are required to make a practical solution for AVC. So, we will plan to capture images for the multimodal dataset.

Acknowledgements The authors acknowledge resources and support provided by the *CMATER* Laboratory, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.

Data Availability Some sample images of the dataset are uploaded in the GitHub repository JUVCsi. The entire dataset will be freely available for research purposes upon positive responses from the reviewers.

Declarations

Conflicts of interest There is no conflict of interest, according to the authors.

References

1. Islam A, Mallik S, Roy A, Agrebi M, Singh PK (2023) A filter-based feature selection framework for vehicle/non-vehicle classification. In: Measurements and instrumentation for machine vision, pp 677–684 . Taylor
2. Bhattacharya D, Bhattacharyya A, Agrebi M, Roy A, Singh P (2022) Dfe-avd: deep feature ensemble for automatic vehicle detection. In: Proceedings of international conference on intelligence computing systems and applications (ICICSA 2022)
3. Maity S, Chakraborty A, Singh PK, Sarkar R (2023) Performance comparison of various yolo models for vehicle detection: An experimental study. In: International conference on data analytics & management, pp 677–684. Springer
4. Zha Z, Tang H, Sun Y, Tang J (2023) Boosting few-shot fine-grained recognition with background suppression and foreground alignment. IEEE Transactions on Circuits and Systems for Video Technology
5. Tang H, Yuan C, Li Z (2022) Tang J Learning attention-guided pyramidal features for few-shot fine-grained recognition. Pattern Recognit 130:108792
6. Gayen S, Maity S, Singh PK, Geem ZW, Sarkar R (2023) Two decades of vehicle make and model recognition–survey, challenges and future directions. Journal of King Saud University-Computer and Information Sciences, pp 101885

7. Li Z, Tang H, Peng Z, Qi G-J, Tang J (2023) Knowledge-guided semantic transfer network for few-shot image recognition. IEEE Transactions on Neural Networks and Learning Systems
8. Tang H, Li Z, Peng Z, Tang J (2020) Blockmix: meta regularization and self-calibrated inference for metric-based meta-learning. In: Proceedings of the 28th ACM international conference on multimedia, pp 610–618
9. Bhattacharyya A, Bhattacharya A, Maity S, Singh PK, Sarkar R (2023) Juvdsi v1: developing and benchmarking a new still image database in indian scenario for automatic vehicle detection. Multimedia Tools and Applications, pp 1–33
10. Ali A, Sarkar R, Das DK (2023) Iruvd: a new still-image based dataset for automatic vehicle detection. Multimedia Tools and Applications, pp 1–27
11. Dong N, Yan S, Tang H, Tang J, Zhang L (2023) Multi-view information integration and propagation for occluded person re-identification. arXiv:2311.03828
12. Yan S, Tang H, Zhang L, Tang J (2023) Image-specific information suppression and implicit local alignment for text-based person search. IEEE Transactions on Neural Networks and Learning Systems
13. Yan S, Dong N, Zhang L, Tang J (2023) Clip-driven fine-grained text-image person re-identification. IEEE Transactions on Image Processing
14. Yan S, Zhang Y, Xie M, Zhang D (2022) Yu Z Cross-domain person re-identification with pose-invariant feature decomposition and hypergraph structure alignment. Neurocomputing 467:229–241
15. Yan S, Dong N, Liu J, Zhang L, Tang J (2023) Learning comprehensive representations with richer self for text-to-image person re-identification. In: Proceedings of the 31st ACM international conference on multimedia, pp 6202–6211
16. Luo Z, Branchaud-Charron F, Lemaire C, Konrad J, Li S, Mishra A, Achkar A, Eichel J, Jodoin P-M (2018) Mio-tcd: a new benchmark dataset for vehicle classification and localization. IEEE Trans Image Process 27(10):5129–5141. https://doi.org/10.1109/TIP.2018.2848705
17. Lin Y-L, Morariu VI, Hsu W, Davis LS (2014) Jointly optimizing 3d model fitting and fine-grained classification. In: Computer Vision–ECCV 2014: 13th european conference. Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, pp 466–480. Springer
18. Krause J, Stark M, Deng J, Fei-Fei L (2013) 3d object representations for fine-grained categorization. In: 2013 IEEE international conference on computer vision workshops, pp 554–561 . https://doi.org/10.1109/ICCVW.2013.77
19. Dong Z, Wu Y, Pei M (2015) Jia Y Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Trans Syst 16(4):2247–2256. https://doi.org/10.1109/TITS.2015.2402438
20. Sochor J, Herout A, Havel J (2016) Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 3006–3015. https://doi.org/10.1109/CVPR.2016.328
21. Yang L, Luo P, Loy CC, Tang X (2015) A large-scale car dataset for fine-grained categorization and verification. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3973–3981. https://doi.org/10.1109/CVPR.2015.7299023
22. Tabassum S, Ullah S, Al-nur NH, Shatabda S (2020) Poribohon-bd: Bangladeshi local vehicle image dataset with annotation for classification. Data in Brief 33:106465. https://doi.org/10.1016/j.dib.2020.106465
23. Hasan MM, Wang Z, Hussain MAI, Fatima K (2021) Bangladeshi native vehicle classification based on transfer learning with deep convolutional neural network. Sensors 21(22):7545
24. Lu L, Wang P (2020) Huang H A large-scale frontal vehicle image dataset for fine-grained vehicle categorization. IEEE Trans Intell Trans Syst 23(3):1818–1828
25. Kramberger T (2020) Potočnik B Lsun-stanford car dataset: enhancing large-scale car image datasets using deep learning for usage in gan training. Appl Sci 10(14):4913
26. Maity S, Bhattacharyya A, Singh PK, Kumar M, Sarkar R (2022) Last decade in vehicle detection and classification: A comprehensive survey. Archives of Computational Methods in Engineering, pp 1–38
27. Sun W, Zhang G, Zhang X, Zhang X (2021) Ge N Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. Multimed Tools Appl 80:30803–30816
28. Silva B, Barbosa-Anda FR (2022) Batista J Exploring multi-loss learning for multi-view fine-grained vehicle classification. J Intell Robot Syst 105(2):43
29. Elkerdawy S, Ray N, Zhang H (2018) Fine-grained vehicle classification with unsupervised parts co-occurrence learning. In: Proceedings of the european conference on computer vision (ECCV) Workshops, pp 0–0
30. Silva B, Oliveira R, Barbosa-Anda FR, Batista J (2021) Multi-view and multi-scale fine-grained vehicle classification with channel convolution feature fusion. In: 2021 IEEE international intelligent transportation systems conference (ITSC), pp 3018–3025. IEEE

31. Sahin O, Nezafat RV (2021) Cetin M Methods for classification of truck trailers using side-fire light detection and ranging (lidar) data. J Intell Trans Syst 26(1):1–13
32. Liu P, Fu H (2021) Ma H An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification. Comput Vis Media 7:217–227
33. Butt MA, Khattak AM, Shafique S, Hayat B, Abid S, Kim K-I, Ayub MW, Sajid A (2021) Adnan A Convolutional neural network based vehicle classification in adverse illuminous conditions for intelligent transportation systems. Complexity 2021:1–11
34. Guo L, Li R (2021) Jiang B An ensemble broad learning scheme for semisupervised vehicle type classification. EEE Trans Neural Netw Learn Syst 32(12):5287–5297
35. Mohine S, Bansod BS, Bhalla R (2022) Basra A Acoustic modality based hybrid deep 1d cnn-bilstm algorithm for moving vehicle classification. IEEE Trans Intell Trans Syst 23(9):16206–16216
36. Tzutalin D (2022) Labelimg is a graphical image annotation tool and label object bounding boxes in images. https://github.com/tzutalin/labelImg
37. Tang H, Liu J, Yan S, Yan R, Li Z, Tang J (2023) M3net: multi-view encoding, matching, and fusion for few-shot fine-grained action recognition. In: Proceedings of the 31st ACM international conference on multimedia, pp 1719–1728
38. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
39. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
40. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
41. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
43. Mascarenhas S, Agarwal M (2021) A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In: 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON), vol 1, pp 96–99. IEEE
44. Naskinova I (2023) Transfer learning with nasnet-mobile for pneumonia x-ray classification. Asian-Eur J Math 16(01):2250240
45. Shah FA, Khan MA Sharif M, Tariq U, Khan A, Kadry S, Thinnukool O (2022) A cascaded design of best features selection for fruit diseases recognition. Comput Mater Contin 70:1491–1507
46. Ballabio D, Todeschini R (2019) Consonni V Recent advances in high-level fusion methods to classify multiple analytical chemical data. Data Handl Sci Technol 31:129–155
47. Dogan A, Birant D A weighted majority voting ensemble approach for classification. In: 2019 4th International conference on computer science and engineering (UBMK), pp 1–6 (2019). IEEE
48. Dey S, Roychoudhury R, Malakar S (2022) Sarkar R An optimized fuzzy ensemble of convolutional neural networks for detecting tuberculosis from chest x-ray images. Appl Soft Comput 114:108094
49. Bühlmann P (2012) Bagging, boosting and ensemble methods. Concepts and methods. Handbook of computational statistics, pp 985–1022
50. Neloy MAI, Nahar N, Hossain MS, Andersson K (2022) A weighted average ensemble technique to predict heart disease. In: Proceedings of the third international conference on trends in computational and cognitive engineering: TCCE 2021, pp 17–29. Springer
51. Buckland M (1994) Gey F The relationship between recall and precision. J Am Soc Inf Sci 45(1):12–19
52. Chicco D (2020) Jurman G The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics 21(1):1–13
53. Townsend J.T Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics 9:40–50 (1971)
54. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
55. Pramanik R, Banerjee B, Efimenko G, Kaplun D (2023) Sarkar R Monkeypox detection from skin lesion images using an amalgamation of cnn models aided with beta function-based normalization scheme. Plos one 18(4):0281815
56. Tabassum S, Ullah S, Al-Nur N.H, Shatabda S Poribohon-bd: Bangladeshi local vehicle image dataset with annotation for classification. Data in Brief 33 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Sourajit Maity[1] · **Debam Saha**[1] · **Pawan Kumar Singh**[2] · **Ram Sarkar**[1]

Debam Saha
debamsaha.cse@gmail.com

Pawan Kumar Singh
pawansingh.ju@gmail.com

Ram Sarkar
ramjucse@gmail.com

[1] Department Computer Science and Engineering, Jadavpur University, Kolkata, India

[2] Department of Information Technology, Jadavpur University, Kolkata, India