Journal of Scientific & Industrial Research
Vol. 82, February 2023, pp. 185-191
DOI: 10.56042/jsir.v82i2.69926

Object Sub-Categorization and Common Framework Method using Iterative AdaBoost for Rapid Detection of Multiple Objects

B Narendra Kumar Rao[1]*, R Ranjana[2], Nagendra Panini Challa[3], S Sreenivasa Chakravarthi[4] & J Vellingiri[5]

[1]School of Computing, Mohan Babu University, Tirupati 517 102, Tamil Nadu, India
[2]Department of Information Technology, Sri Sairam Engineering College, Chennai 600 044, Tamil Nadu, India
[3]School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati 522 237, Andhra Pradesh, India
[4]Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidayapeetham
Chennai 601 103, Tamil Nadu, India
[5]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632 014, Tamil Nadu, India

Received 03 August 2022; revised 27 September 2022; accepted 07 October 2022

Object detection and tracking in real time has numerous applications and benefits in various fields like survey, crime detection etc. The idea of gaining useful information from real time scenes on the roads is called as Traffic Scene Perception (TSP). TSP actually consists of three subtasks namely, detecting things of interest, recognizing the discovered objects and tracking of the moving objects. Normally the results obtained could be of value in object recognition and tracking, however the detection of a particular object of interest is of higher value in any real time scenario. The prevalent systems focus on developing unique detectors for each of the above-mentioned subtasks and they work upon utilizing different features. This obviously is time consuming and involves multiple redundant operations. Hence in this paper a common framework using the enhanced AdaBoost algorithm is proposed which will examine all dense characteristics only once thereby increasing the detection speed substantially. An object sub-categorization strategy is proposed to capture the intra-class variance of objects in order to boost generalisation performance even more. We use three detection applications to demonstrate the efficiency of the proposed framework: traffic sign detection, car detection, and bike detection. On numerous benchmark data sets, the proposed framework delivers competitive performance using state-of-the-art techniques.

Keywords: Computer vision, Segmentation, Supervised learning, Traffic scene perception

Introduction

One of several rapidly-emerging fields in the intelligent transportation system is Traffic Scene Perception (TSP) based upon vision. Over the last decade, this field has been extensively researched.[1] The three steps of TSP include detection, recognition, and tracking of diverse objects that we are trying to detect. Detection is segregating features relevant to an object from the entire scene, example extracting number plate of a vehicle. Recognition is identifying the object against a standard type, example finding whether an automobile is a car or bus. Tracing is, knowing the location or path taken by an object.

We concentrate on three types of objects: traffic signs, automobiles, and bikers. The traffic sign detection notifies the driver regarding the changes in traffic. The goal is to correctly locate and recognise road signals in a variety of driving situations. For traffic sign detection some approaches use signal and colour information. These methods, on the other hand, are not adaptable when considered for adverse weather and illumination situations. Additionally, the visibility of traffic signs can vary over time as a result of weather and accident damage.[2] Most modern approaches use gradient features, like Local Binary Patterns (LBP) and Histograms of Directed Gradients (HOG) instead of colour and form features. Although the features are relatively resistant to distortion and changes in image lighting, they still cannot deal with extreme deformation.

Automobile detection, taken up as the second task, is a challenging task just because of the variations available in the various varieties of automobiles and their viewpoints. Initial implementation of adapting sliding window methods that work well for face detection, seems to be promising in automobile detection, however, they fail when the viewpoints vary.[3,4] The Deformed parts model, could be easily adapted to detect automobiles which use the concept

*Author for Correspondence
E-mail: narendrakumarraob@gmail.com

of subcategorization.[5] The third task of the proposed framework is detecting bikers, which is novel and the most challenging in any TSP scenario, again due to its versatility and varied features and viewpoints. Earlier works have tried with some success in modifying the existing pedestrian tracking apps to track cyclists. Detecting bikers is not reported in the literature. The existing work considers each of the detection and recognition as separate problems and extracts features accordingly for further processing. The proposed common framework therefore aims at using the same set of features for detecting three different classes of objects.

Problem Definition

The aim of the proposed work is the development of a framework that can recognise objects from a live webcam feed is proposed. Most of the previous research has tried to design specific detectors using specific features for specific objects. The novelty of the proposed work is designing a common framework that can identify multiple objects that are of different types. Using webcam and Live Stream, the System determines what all the objects present in the feed are. The major objectives of the proposed work are as follows.

- This work aims to build a framework that performs object detection.
- To design the framework in such a way that all the classes are addressed.
- To focus on all three classes namely detecting traffic signs, automobiles and bikers in the same framework.
- To enhance the speed of the detection of objects.

The major areas where the proposed work can be applied is

- Traffic places
- Shopping malls
- Hospitals etc.

The objects can be identified and necessary actions can be taken accordingly. The next Section discusses the existing techniques that are available in object detection and their relative merits and demerits.

Background and Motivation

The topic of recognising objects is very early and began with still grey-scale photographs. Until recently progress has been made and there are several studies in the literature that focus on object detection, however, they insist on using a specific framework for a particular type of object. Based on the application wherein it is used, the object detection techniques vary substantially. The next Section discusses a few such applications and the challenges faced in object detection.[6–8]

One of the major fields where object detection is vital is autonomous driving. It requires a wide range of sensory abilities in both angle and distance, such as traffic merges, four-way stops, overtaking, and other maneuvers. The primary tasks involved in any vision-based vehicle guidance system are road detection, obstacle detection, and sign identification. The first two have been studied for many years and have produced numerous positive results, but traffic sign identification has received less attention[9] Drivers may get a lot of useful information about the road from traffic signs, which makes driving easy and safe. According to the authors, traffic indicators will most likely serve the same purpose for autonomous vehicles. Their design is so that it can be identified by human drivers, owing to their distinct colour and shape from natural surroundings.

Another study discusses about the vehicle detection and tracking system that can be deployed to analyse scenes captured from a car- mounted camera. It is designed to work well in any weather condition. On a PC equipped with an IMAP-VISION real-time image processing board, the system runs at a rate of 15 frames per second.[10]

Over the previous decade, numerous traffic sign detectors have been presented, each with its own set of rigorous benchmarks. The majority of existing traffic sign detectors is focused on their look. These detectors are classified based on Color, shape, texture, and hybrid approaches.[9,11] A two-stage process is commonly used in colour-based systems. Segmentation is the method of dividing an image for further processing. Segmentation begins with a thresholding procedure in a single colour space. Shape detection is then applied to the segmented sections. As the RGB colour space is delicate to changes in lighting, various methods are used to convert it to the Hue-Saturation-Intensity (HSI) colour space, which is somewhat light- independent. Other approaches use normalised RGB space segmentation, which has been demonstrated to perform better than the HSI. RGB space and normalised HSI can reduce the detrimental effects of lighting changes, yet they still fall short in several cases. Canny edge detectors or their derivatives are used in shape-based techniques to detect edges or corners from raw pictures. These detectors are not affected by changes in illumination, but they have high memory and computational requirement for huge images. Texture-based techniques begin by extracting hand-crafted features computed from image texture, which are then used to train a classifier. Histograms of Oriented Gradients (HOGs) and Local Binary Patterns (LBPs), Aggregate Channel Features (ACF) and other hand-crafted features are popular. Some approaches combine HOG and SVM features, while others combine ACF and an AdaBoost classifier.

A Convolutional Neural Network (CNN) is used to detect traffic signs in addition to the methods mentioned above, and it produces excellent results. Hybrid approaches combine the previously mentioned approaches.

Many of the existing car detectors rely on vision. They focus on vision-based automobile detectors that use monocular information. There are mainly three types of detectors and they are based on DPM, sub-categorization, and motion approaches. The foundation for DPM-based techniques is the deformable model (DPM), which was successfully used in car recognition. In a DPM variation, the number of automobile orientations is discretized, and each component of the mixture model matches to single orientation. Occlusion patterns are commonly used for training DPM and to reason about the interactions between cars and hindrances in order to detect them. DPM frequently employs sub categorization-based algorithms to detect automobiles from numerous perspectives. Locally linear embedding method with applied HOG features, sub categorization of automobiles is used to learn the car orientation.[12] A semi-supervised clustering algorithm utilizing ACF characteristics is used to group cars with comparable perspectives, occlusions, and truncation scenarios. As Monocular images cannot provide any 3D or depth information[6], motion-based techniques are used to frequently leverage appearance cues in monocular vision.[13] Cars are detected using an adaptive background model based on the motion that distinguishes them from the background. An adaptive background model is used in modelling the area from the camera's view-field where overtaking cars scenario happens.

From the above mentioned discussion on literature, it can be observed that traffic sign detection and automobile detection have evolved independently and use varied techniques to improve their effectiveness. A common framework is not reported in the above-mentioned literature. It can be mentioned that all the discussed works are designed for identifying or tracking a specific object or for a specific application. So there is a need for a common framework that can be used for detecting any object irrespective of the application. The proposed framework helps in achieving the same.

Proposed System

In the proposed method, the objects are identified by performing object sub-categorization. In order to effectively sub-categorize, we need to extract features. Object features can be classified as visual and geometric. Examples of visual features are colour and texture and geometric features are orientation, width and height. The extraction of these features requires separate methodologies as discussed in the following Sections.

Visual Feature Extraction

To extract the visual features, a clustering algorithm is used. To perform this clustering, approaches like HOG and ACF are used. HOG can be used to capture the shapes of objects but it does not consider the colour of the object. ACF uses both shape and colour information. It can perform better than HOG. For clustering, a total of ten feature channels are used: LUV colour channels (3 channels), a histogram of directed gradients at 6 bins (6 channels), and normalized magnitude gradient (6 channels) (1 channel). All training samples are converted to the same median size before the training.

Extraction of Geometrical Features

Apart from using visual features, we use certain geometrical features in order to detect objects. The geometrical features used to identify the objects in this framework are as follows:

3D Orientation –represents how an object is placed and appears as the cameras viewpoint shifts

Aspect Ratio – width and height of the object and is very much needed for identification

Truncation Level – decides which part of the object is within a defined boundary and what needs to be removed for being outside it.

Occlusion Index- Occlusion is a condition wherein one object may hide another and the detection algorithm may identify it as a new object once the hiding object is removed. In object tracking, accounting for occlusion is very important else it may result in poor performance. An occlusion index is given to categorize the impact of occlusion and is normally ranged from being not occluded, slightly occluded to extensively occluded.

In order to extract geometric features, clustering is employed in the subcategories obtained. The number of clusters in the sub categorization approach is varied during the design to see how the number of clusters affects the performance of the model. The increased number of subcategories affects the geometrical and aesthetic aspects of the image. In spectral clustering, the geometrical features outperform visual features. As the number of subcategories increases, the detection performance improves. However, using a large number of subcategories can degrade performance since the average items of data in the view subcategory is insufficient to train a useful model. We consider the above factors while modelling the design of the system.

Iterative Enhanced AdaBoost Algorithm

The most challenging part of the proposed framework is object detection and identification in a dynamic real- time environment. It is challenging because it needs extraction and processing of several varied and common features. So, standard object detection algorithms use AdaBoost (Adaptive Boosting) training to reduce the computational load. AdaBoost works primarily by identifying the relevant features of all the available features in a scenario. All these features together are called as weak classifiers and their weighted sum is called a strong classifier. In the proposed work, we modify the AdaBoost classifier for better efficiency. The modified classifier is a shrinkage version of the standard classifier. In the proposed classifier, bootstrapping procedure is used to train the classifier. Bootstrapping is a standard technique to estimate the accuracy of the known classifier over a new data and is normally used to train the classifier. Hard negative samples are collected and the classifier is retrained. However, when object subcategorization is used on an object class, one classifier is required to be trained for each subcategory to identify the objects.

In order to train the model, training data is considered as the weighted distributions and the weak learners are identified from the model. The error rate, defined in terms of the total wrong predictions, is computed for the weak learners and the weights are updated by using the error rate and shrinkage factor. Shrinkage factor is the weighted coefficient that is added to regularize the boosting process. This process is repeated for every element in the training data. The final classifier is constructed after analyzing all the elements present in the training data.

Bootstrapping: After the training phase, three bootstrapping iterations are executed to increase the performance of the learnt classifier. Random negative samples from training images with positive slices cut off during the initial training phase, and further bootstrapping iterations add extra negatives to the training set.

- Weighted distribution of the training data is considered during the training process.
- Weak learners are identified and their error rate is computed and weights are added for the weak learners using the shrinkage factor.

The above steps are computed for all the elements of the training data and the final classifier is constructed after performing the bootstrap iterations as well.

The steps involved are described below:

Iterative Enhanced AdaBoost algorithm	
STEP – 1:	Assign equal weights to all records in data set
STEP – 2:	Weak learners are identified from the model.
STEP – 3:	The error rate is computed for the weak learners and the weights are updated by using the error rate and shrinkage factor
STEP – 4:	The normalization factor is computed. Repeat STEP - 1 to 3 for each element in the data set.
STEP – 5:	Compute and adjust the weights
STEP – 6:	Perform three iterations of bootstrapping to increase the performance of the classifier
STEP – 7:	Provide the final detection of objects

The proposed algorithm aims to implement the following functional and non-functional requirements

Functional Requirements

- Capture the feed from the real time scene.
- Detect the objects from the captured feed.
- Provide the percentage to which the detection is estimated to be correct.

Non-Functional Requirements

- Security: The captured feed must not be shared with any other third party.
- Reliability: The objects detected must be with maximum accuracy.
- Performance: The performance of the framework must be really good.
- Response Time: Response must be really fast.
- Maintainability: The framework must be easy to maintain.

The implementation and evaluation consist of three steps namely training the model, feature extraction and post -processing.

Experimental Setup

This section elaborates on the experimental setup and evaluation of the proposed framework. The dataset used contains 7481 images used in the model training and 7518 test images, totalling over 80 thousand identified objects. The dataset contains a vast number of objects of various sizes, view angles, occlusion patterns, and truncation circumstances, as we can see. Based on the obvious variety of the current objects, the dataset is divided into three subsets in terms of object size, occlusion, and truncation complexity (Easy, Moderate, and Hard). There are 15710 objects in the moderate group and heights from 25 to 270 pix and aspect-ratio ranging from 0.9 to 4.0. The training images are divided into:

- Training set (primary 4000 images)
- Validation set (secondary 4000 images and residual 3481 images).

In order to evaluate the algorithm, a sample of the real time camera feed is taken and processed. A button is provided which when pressed will start the video capture. A predefined time interval is set by the user and the video is captured until the desired duration. When the button is pressed, a function is called and the model is loaded into the system. Then the permission for accessing the camera is checked. If the permissions are given, the live video is captured by the system using the webcam and the model identifies the objects present in the system.

The model uses the object subcategorization method and the feature extraction method in order to identify the objects present in the feed obtained by the webcam. Once the model is loaded, it subcategorizes the objects present in the feed by using feature extraction methods. The model tries to identify the objects present in the feed by analysing the features extracted from the feed data. Once all the objects are identified, an array of names of the objects and the confidence scores are sent back to the function that is calling the model.

Now that all the results are fetched by the function in the form of an array, the array is traversed to display all the names of the objects identified by the model and the confidence scores that indicate the extent to which the prediction is correct. If there are any overlapping objects, the boxes are drawn in such a way that they do not overlap. The details of the test data set are given in Table 1.

Condition for true positive: if a match is within 25% for a single boundary

Table 1 — Details of test data set

Parameter	Details
Resolution	40 × 100 pixels
Positive samples	550
Negative samples	500
Single scale test photos	170 images with 200 objects
Multi-scale test photos	108 images with 139 objects

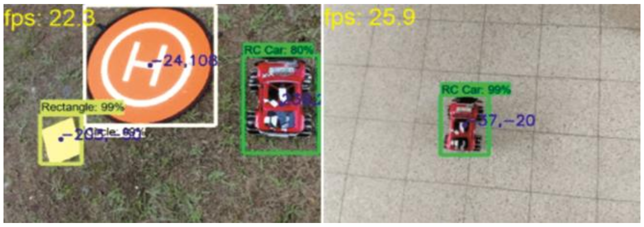

Fig. 1 — Testing object detection by the model

If more than one boundary matches, one with a more confidence score is considered.

Different objects are detected by the system along with the confidence score with which the objects are identified and Fig. 1 represents the same.

Performance Evaluation and Discussion

To evaluate the performance of the model, the model is tested on different datasets namely

1. UIUC- Car image database captured at University of Illinois – Urbana Campaign (UIUC)[14]
2. GTSDB – The German Traffic Sign Detection Benchmark[15]
3. KITTI – these data sets from the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) vision benchmark suite[16] captures data by driving around the city of Karlsruhe and nearly fifteen cars and thirty walkers are visible per image. It is used to evaluate experiments in autonomous driving

The performance is compared and tabulated as follows based on the results obtained. Observed values when the traffic sign detection was done is presented in Table 2. The GTSDB dataset categorizes the signs as mandatory, dangerous and prohibitory. So the experiment was evaluated for all the three categories with combined features[17] as given below

ACF = Aggregated channel features

LUV = Colour channels with luminance L and Color values U and V

Sp-LBP = Spatially pooled local binary patterns

Sp-CoV = Spatially pooled co variance features

The various possible combinations are

COM1-ACF, COM2-sp-LBP and ACF, COM3-sp-CoV and LUV, COM4-sp-COV and ACF, COM5-Sp-CoV and Sp-LBP and ACF[17]

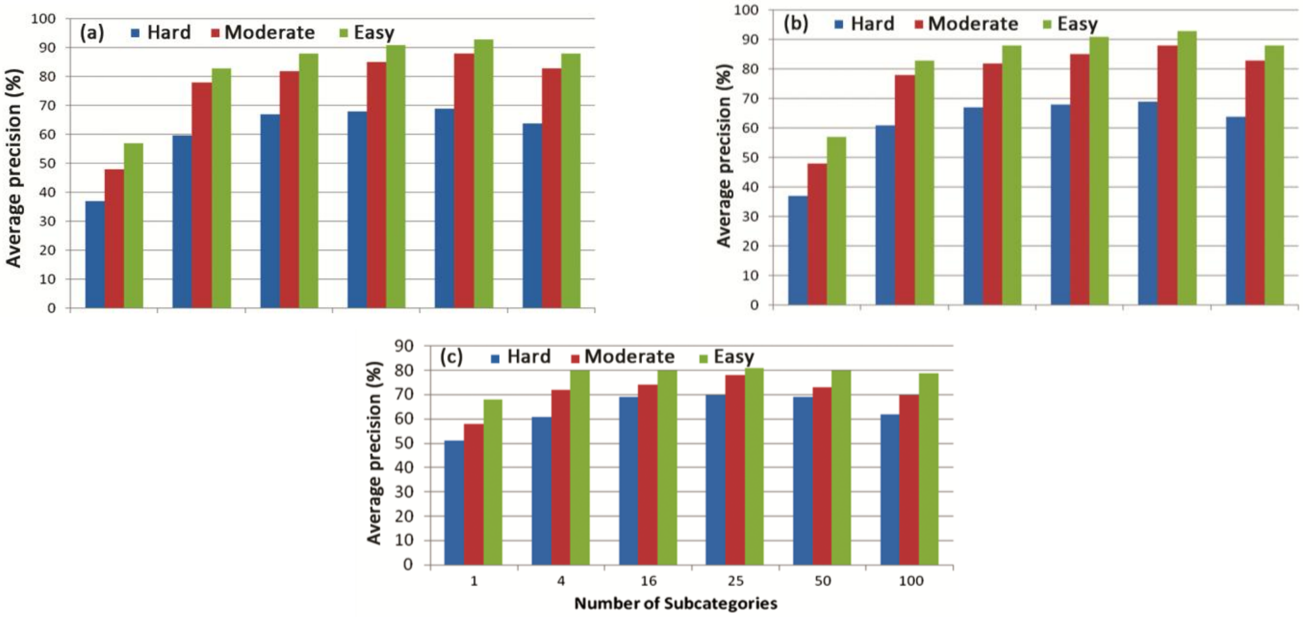

Fig. 2 — Detection of automobiles: (a) Spectral clustering + geometrical features, (b) Spectral clustering + visual features, (c) Spectral clustering + aspect-ratios

Table 2 — Performance with various feature combinations

Feature-combination	Mandate	Danger	Prohibit (%)	Average
COM1	92.55	94.48	98.62	95.22
COM2	96.02	95.06	99.99	97.02
COM3	95.46	96.57	99.20	97.08
COM4	95.51	95.13	98.63	96.42
COM5	97.47	98.00	100	98.49

Table 3 — Performance comparison of various detectors

Method	F-Measure%	Det. rate%	No. of false Positive
Current Method	98.6	99.18	3
Pruning	98.6	97.6	1
AdaBoost	98.6	98.4	2
AdaBoost + LDA	98.6	97.6	1
CS-AdaBoost	95.3	95.3	9

From the Table 3 we can observe that the detection rate is better for this algorithm when compared to the other existing algorithms. A false positive means the number of objects wrongly identified by the model. Even though the false positives are higher than some other algorithms, it is better because of the good rate.

Experimental Results

This application enables the user to interact with the system and get the required results of object detection. When spectral clustering was used with geometrical features, visual features, and aspect ratios of the objects in automobile detection, the results are observed as depicted in Fig. 2(a), Fig. 2(b), and Fig. 2(c) respectively.

From all the three plots it is evident that as the number of sub categories is increased, we get a better efficiency, however it drops after a threshold, beyond which increasing the sub categories does not yield a improved detection rate.

Conclusions

Object detection and identification in a live stream of real-time traffic are challenging. The real time live feed may contain varied objects from automobiles and bikers to pedestrians and even sign boards. Each of these objects has unique features and has been handled as separate entity in most of the literature. In the proposed work, a common framework for identifying three different classes of objects namely automobiles, traffic signs and bikers were proposed. The framework used the object subcategorization technique and the iterative AdaBoost algorithm for object detection. The proposed framework was evaluated using standard algorithms and the results obtained were compared against standard literature. The proposed method was found to be accurate. In future research, the context information may be included to better detect objects and Convolution neural networks with appropriate weights can be deployed for increased accuracy.

References

1 Agarwal S, Awan A & Roth D, Learning to detect objects in images via a sparse, part-based representation, *IEEE Trans Pattern Anal Mach Intell*, **26(11)** (2004) 1475–1490.

2 Kyo S, Koga T, Sakurai K & Okazaki S, A robust vehicle detecting and tracking system for wet weather conditions using the IMAP-vision image processing board, in *Proc IEEE Int Conf Intel Transport Syst (Cat. No.99TH8383)* 1999, 423–428, doi: 10.1109/ITSC.1999.821095.

3 Dalal N & Triggs B, Histograms of oriented gradients for human detection, in *Proc. IEEE Comput Soc Conf Comput Vis Pattern Recognit*, (CVPR'05) **1**, 2005, 886–893, doi: 10.1109/CVPR.2005.177

4 Viola P & Jones M J, Robust real-time face detection, *Int J Comput Vis*, **57(2)** (2004) 137–154.

5 Felzenszwalb P F, Girshick R B, McAllester D A & Ramanan D, Object detection with discriminatively trained part-based models, *IEEE Trans Pattern Anal Mach Intell*, **32(9)** (2009) 1627–1645.

6 Geiger A, Wojek C & Urtasun R, Joint 3D estimation of objects and scene layout, in *Proc Adv Neural Inf Process Syst*, **24** (2011).

7 Ahonen T, Hadid A & Pietikainen M, Face description with local binary patterns: application to face recognition, *IEEE Trans Pattern Anal Mach Intell*, **28(12)** (2006) 2037–2041.

8 Wu J, Charles B S, Mullin M D & Rehg J M, Fast asymmetric learning for cascade face detection, *IEEE Trans Pattern Anal Mach Intell*, **30(3)** (2008) 369–382.

9 Markus M, Radu T, Rodrigo B & Gool L V, Traffic sign recognition-how far are we from the solution, in *Proc Int Jt Conf Neural Netw* (IJCNN) 2013, 1–8.

10 Alberto B, Andrea C, Stefano C & Paolo Z, Lateral vehicles detection using monocular high resolution cameras on terramax, in *Proc IEEE Intell Veh Symp* 2008, 1143–1148.

11 De la Escalera A, Moreno L E, Salichs M A & Armingol J M, Road traffic sign detection and classification, *IEEE Trans Ind Electron*, **44(6)** (1997) 848–859.

12 Jianzhu C, Fuqiang L, Zhi-peng L & Zhen J, Vehicle localisation using a single camera, in Proc *IEEE Intel Veh Symp* 2010, 871–876.

13 Caraffi C & Cattani S, VisLab at the Grand Challenge, *IEEE Computer*, **39(12)** (2006) 36–37.

14 Nagarajan B & Devendran V, Vehicle classification under cluttered background and mild occlusion using Zernike features, *Procedia Eng*, **30** (2012) 201–209.

15 Houben S, Stallkamp J, Salmen J, Schlipsing M & Christian I, Detection of traffic signs in real-world images: the German traffic sign detection benchmark, *Int Jt Conf Neural Netw*, 2013, 1–8, DOI:10.1109/IJCNN.2013.6706807.

16 Geiger A, Lenz P & Urtasun R, Are we ready for autonomous driving?, the kittivision benchmark suite, *Conf Comput Vis Pattern Recognit* (IEEE) 2012, 3354–3361.

17 Qichang H, Paisitkriangkrai S, Chunhua S, Hengel A V D & Fatih P, Fast detection of multiple objects in traffic scenes with a common detection framework, *IEEE Trans Intell Trans Syst*, **17(4)** (2016) 1002–1014, doi: 10.1109/TITS.2015.2496795.