D2TFRS: An Object Recognition Method for Autonomous Vehicles Based on RGB and Spatial Values of Pixels

Furqan Alam[1(✉)], Rashid Mehmood[2], and Iyad Katib[1]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
fmohammed0026@stu.kau.edu.sa, iakatib@kau.edu.sa
[2] High Performance Computing Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia
RMehmood@kau.edu.sa

Abstract. Autonomous driving is now near future reality which will transform our world due to its numerous benefits. The foremost challenge to this task is to correctly identify the objects in the driving environment. In this work, we propose an object recognition method known as Decision Tree and Decision Fusion based Recognition System (D2TFRS) for autonomous driving. We combined two separate feature sets, which are RGB pixel values and spatial points X,Y of each pixel to form our dataset. The D2TFRS is based on our intuition that reclassification of pre-identified misclassified objects in a driving environment can give better prediction accuracy. Results showed that D2TFRS outperformed AdaBoost classifier and performed better than C5.0 classifier in terms of the classification accuracy and Kappa. In terms of speed, C5.0 outperforms both AdaBoost and D2TFRS. However, D2TFRS outperformed AdaBoost with respect to speed. We strongly believe that D2TFRS will have better parallelization performance compared to the other two methods and it will be investigated in our future work.

Keywords: Autonomous driving · Autonomous vehicles · Object recognition
Decision tree · Decision fusion · Deep learning · Majority voting
C5.0 · SVM

1 Introduction

An autonomous vehicle (AVs) is one that can accelerate, increase and decrease speeds, put and release brakes and steer, itself avoiding any sort of accidents. Such technology has long been part of Hollywood sci-fi quixotic vision of the future. This is due to the fact that AVs will free drivers from boring side of driving during travel and reduce accident rates by providing breathtaking control over vehicles. In past, many attempts have been made but subjected to the limitation of available technologies. However, in recent years with technological advancements, the dream of AVs come very close to reality. Now we are able to manufacture them nevertheless they are in their testing phase. AVs have the potential to change how we look at our surroundings.

R. Mehmood et al. (Eds.): SCITA 2017, LNICST 224, pp. 155–168, 2018.
https://doi.org/10.1007/978-3-319-94180-6_16

The Autonomous Driving (AD) is getting lots of attention and popularity due to its various benefits [1] and assumed to be an on-road reality soon. Most of the major industry titans which include Google, Tesla, Ford, Volvo, BMW, Microsoft, Apple, and others, are making huge investments in developing technologies which will enable AD. A new forecast by Intel and Strategy Analytics research firm estimated that AVs will be a 7$ trillion market by 2050 [2]. The competition of which company will bring its AVs first on the road to common public getting so tough, resulted in various perk luring practices to get skilled engineers from the rival companies and stealing AVs technologies from the competitors [3–5]. The core of these developments revolves around the critical question, how to perceive driving environment with higher certainties.

The key technology on which success of AVs depends is how accurately, they are able to perceive the driving environment. The initial step in this quest is to recognize the static and dynamic objects around the vehicles with higher accuracies. In a driving environment this object recognition problem in more complex due to the fact that it is multi-class problem and given the dynamic nature of the driving environment which add further complexities to it. AVs consist of several on-board and off-board sensors such as cameras, LIDAR, Radar and GPS as illustrated in Fig. 1.

Fig. 1. General view of the autonomous driving environment.

The aim of any object recognition system is to predict with the highest degree of certainty for the given task. The result evaluation of different classification schemes can be different in terms of classification accuracies. One classifier tends to produce better predictions for a particular class, though its overall accuracy can be lower as compared to the other. The sets of patterns of rightly classified or misclassified data instances by the distinct classifiers would not certainly coincide, thus this form the basis to acquire better classification accuracies through decision fusion of predictions from various classifiers.

Supervised machine learning algorithms learn using a training dataset which contains independent variables and their response variables. They keep on learning until the minimum possible classification error achieved. In this work, our focus is to use supervised learning in a way that it enhances the object classification accuracy in a driving environment which will enable an auto-pilot to take better driving decisions.

1.1 Contributions

In this work, our main aim is to develop a methodology to achieve higher object classification accuracy by integrating supervised learning and decision fusion. The main contributions of this work are:

- We manually labeled images from a subset of KITTI city dataset [6] by using free-form selection (polygon) rather than a box or rectangular selection. This means, highly accurate pixel labeling is achieved by carefully selecting only the area of interest to enhance training of the algorithm.
- We used two feature sets together for training purposes. The first feature set consist of RGB values whereas the second feature set consist of the spatial location of each pixel i.e. x and y coordinates. The use of two feature sets increased the training accuracy considerably.
- We tried to demonstrate how pre-identification of worst data instances which are hardest to classify correctly, improves the accuracy of a classifier system.

1.2 Paper Structure

The paper is divided into seven sections. Section 2, contains literature review and in Sect. 3 we explained dataset and data preparation for this work. Further is Sect. 4, the classifiers are discussed which are used in this work, whereas in Sect. 5, the proposed method has been explained in detail. We represent results in Sect. 6 and finally, conclusions are drawn in Sect. 7.

2 Literature Review

Machine learning is mighty artificial intelligence (AI) tool which helps us to understand the complicated world around us by learning. Nowadays machine learning applied in almost every field such as biomedical, education, business, security, robotics, networking and much more [7–9]. Machine learning which eventually uses to develop AI for autonomous vehicles (AVs), ranges from infotainment systems to advanced driver assistance systems (ADAS) and further to complete self-driving auto-pilots. With machine learning, AI systems continuously learn from experience by their ability to foresee and identify the happenings in their surroundings, which is promising to be highly constructive when integrated into a software architecture of AVs. Search Engine giant Google and Tesla have been doing considerable research and development for developing the AI capabilities for their autonomous cars, albeit in a more vocal manner than their counterparts. Perceiving driving environment is the key problem for facilitating safe and smooth autonomous driving. The problem starts from recognizing static objects (road, speed breakers, traffic light, buildings) and dynamic objects (cars, cycles, trucks) around AVs. All the different objects must be classified by an object recognition system which is a multi-class problem for AVs and has been well studied in [10–12].

Identifying, tracking, and avoiding human beings is a pivotal capability of AVs. Pedestrian recognition must guarantee the safety of humans walking on footpaths and crossing the roads while auto-pilots are driving AVs which is studied in [13–15].

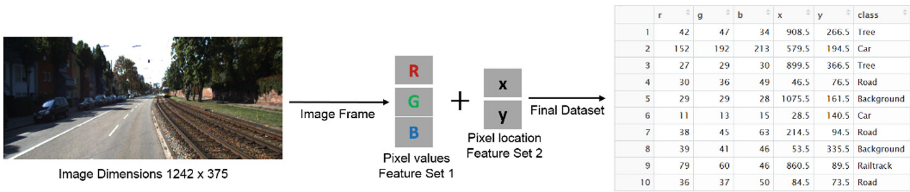
At Google, research scientist, Anelia Angelova introduced a novel pedestrian detection system that only requires video images [16]. Similarly to [16] in [17], the deep learning based video-only pedestrian detection system is presented which is under development at the University of California, San Diego. Works like [16, 17], could make human detection systems for AVs, to pinpoint humans using low-cost sensors like cameras alone without using expensive Lidar units which can reduce the cost of AVs considerably with high reliability. The developments in [16, 17], also support the arguments of Tesla CEO Elon Musk against using expensive Lidar technology for self-driving cars. A realistic situation can arise when AVs will have a sudden encounter with a pedestrian, to save life and avoid collision with a pedestrian is a crucial and complex problem. In [18] paper, author studies the problem of detecting sudden pedestrian encounters to aid drivers to avert any sort of accident. Road detection is a crucial problem for AVs as it decides how much space is available for driving and turning to ensure safe and smooth driving. In recent years a lot of development has been seen in this area [19–21]. For this purpose in [22], authors proposed a road detection technique using SVM which automatically updates the training data to minimize classification error. Similarly, in [23], linear SVM is used for Segment-Based Fractal Texture Analysis (SFTA) and compared with the multi-layer convolutional neural network (CNN). Both linear SVM and CNN produced very high classification accuracies. However, CNN showed slightly better specificity.

Another way to perceive driving environment is to combine multiple decisions or multiple sensor data for deducing the driving environment. This can also be defined as Data fusion which is well studied from various perspectives in one of the latest and comprehensive surveys [24]. The paper review mathematical methods for data fusion, specific sensor environments. Further authors discussed the emerging trends which would be benefited from data fusion [24]. For example combining GPS and camera images to predict safe driving distance to another vehicle on the road. Combining the multiple inputs or features into a single output is a complex problem but the outcome tends to show more certainty than single sensor data analytics as achieved in proceeding literature. For example in [25] authors fuse cameras images and LIDAR for deducing driving environment by labeling segments of images whereas in [26] object grid maps are created by combining camera images and laser. In literature such as [13, 14], the single feature set is used to identify humans. Solving the same problem, though using multi-sensor data, a smoothing-based depth up-sampling method for human detection is proposed in [27] which fuses camera images and LIDAR data. Furthermore in [28] authors uses knowledge of object classes to recognize humans, car obstacles, and bicyclists. A multi-layer perceptron (MLP) classifier is used in [29] to recognize, interpret and track autonomous moving objects. Blend of stereo vision, LIDAR and stereo vision data is used and supplied to MLP in [29] as input. Hane et, al use images from cameras with wheel odometry for drawing out static obstacles [30] whereas in [31] Dempster Shafer theory of evidence is used to integrate sensors data to classify the obstacles.

Combining results of multiple classifiers tend to produce better results, this is a well-proven concept. This sort of combination is known as Decision Fusion (DF). However, it is important to select a combination of right classifiers in order to take benefits from DF. In one of such work [32], authors critically examine the use of the ρ-correlation as a way to quantify the classifier diversity for selecting classifiers for fusion. DF methods are used successfully for image classification problems. A scheme to aggregate the results of different classifiers is proposed in [33]. Situations where the classifiers disagree with each other in [33], are solved by computing the pointwise accuracy and finding the global reliability [34]. Traditional methods for hyperspectral image classification typically use raw spectral signatures without considering spatial characteristics. In work [35], a classification algorithm based on Gabor features and decision fusion is proposed. First, the adjacent and high correlated spectral bands are intelligently grouped by coefficient correlation matrix. Following that, Gabor features in each group are extracted in PCA-projected subspaces to quantify local orientation and scale characteristics. Afterwards, locality preserving non-negative matrix factorization is incorporated to reduce the dimensionalities of these feature subspaces. Finally, the classification results from Gaussian-mixture-model classifiers are merged by a decision fusion rule. Experimental results show that the proposed algorithms substantially outperforms the traditional and state-of-the-art methods. Majority of AVs researches are based on binary classification problems and less attention has been given to challenging multi-class problems.

3 Dataset and Data Preparation

We used KITTI datasets [6] for this work. We have used two feature sets which are (R, G,B) values of the pixels and spatial values of each pixel in the image frames of dimension 1242 × 375 as depicted in Fig. 2. We create a dataset which has six attributes, namely r, g, b, x, y, class. We used Raster package [36] in R, to compute pixel values and location of each pixel in the image frame.



	r	g	b	x	y	class
1	42	47	34	908.5	266.5	Tree
2	152	192	213	579.5	194.5	Car
3	27	29	30	899.5	366.5	Tree
4	30	36	49	46.5	76.5	Road
5	29	29	28	1075.5	161.5	Background
6	11	13	15	28.5	140.5	Car
7	38	45	63	214.5	94.5	Road
8	39	41	46	53.5	335.5	Background
9	79	60	46	860.5	89.5	Railtrack
10	36	37	50	84.5	73.5	Road

Fig. 2. Data preparation process.

Further, we manually labeled the images from a subset of KITTI city dataset [6] by using free-form selection (polygon) rather than a box or rectangular selection which is further depicted in Fig. 3. This means highly accurate pixel labeling is achieved by carefully selecting only the area of interest to enhance training of the machine learning algorithm. After selecting pixels of a particular object, we manually labeled every object pixel and spatial values to make final dataset. Our dataset contains 380000 rows and six attributes and we divided the datasets into two parts which are training 60% and 40% testing. Further, we used SMOTE algorithm on training data to overcome class imbalance problem which is discussed in proceeding section.

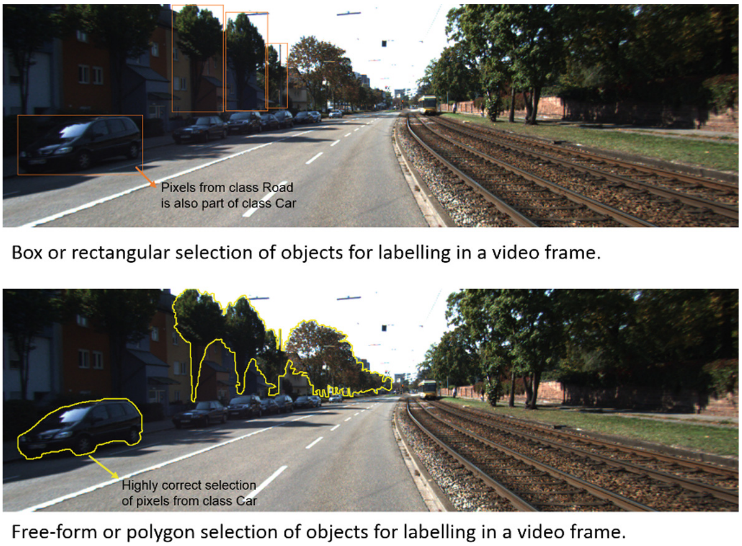


Fig. 3. Object labeling process. The image is taken from KITTI dataset [6].

4 Algorithms

In this work, we used several supervised machine learning algorithms based on their prediction accuracy, execution time and scalability for classification and decision fusion through majority voting.

4.1 Decision Tree

C4.5 is a supervised learning algorithm which builds a decision tree using the concept of information entropy proposed by Quinlan [37]. It can handle both continuous and discrete attributes. In this work we used C5.0 which is an extension of C4.5, is also commercially sold by Ross Quinlan. The reason to use C5.0 for this work lies in the fact that it is extremely fast, several folds faster than its predecessor C4.5. It can take benefits of multi-core and multiple CPU [38]. Further, it has better memory management, which is needed because of a significant amount of data processing is required particularly in RGB image classification. It can give similar or better results to C4.5 and forms significantly smaller decision trees.

4.2 Support Vector Machine

Support vector machine (SVM) is one of the most accurate classifiers and have a sound theoretical foundation. SVM constructs hyperplane or a set of hyperplanes for performing classification and regression [39]. It can compete with far more complex modern-day classifiers in terms of accuracies and it is considered as one of the best classifiers which are listed among top 10 machine learning algorithm [38].

4.3 Deep Learning

Deep learning (DL) mimics a neural system of humans for performing learning task. It belongs to the family of artificial neural networks. It digs deep into the data and finds out the complex relationships among data elements. Widely used in image recognition, natural language processing, speech recognition and bioinformatics due to its quality of producing highly accurate predictions, though DL is computationally expensive. To develop a further understanding of various deep learning architectures, models, and their mathematical formulations in a more comprehensive manner, work such as [40–42] can be investigated.

5 Proposed Method

In this work, our prime focus is to identify data instances which are most difficult to classify for the given supervised machine learning algorithm, prior to classifying them and to reclassify the predicted misclassified data. We divide our main method into two phases. The first phase in which we carefully train our models and generate data for the training of proceeding stage because, from stage-2 onwards, machine learning algorithms need to be trained with the data specific to that stage. In the second phase, in which we test our whole method to predict its accuracy. All the experimentations are performed on R statistical machine learning platform and H2O [43], SVM [44], C5.0 [45] and Caret [46] libraries are used.

5.1 Training

Formally we can define our training process as, for the given training set (X_i, Y_i), we want to generate a classifier function to predict Y_i labels for new $X_i = (r_{i1}, g_{i2}, b_{i3}, x_{i4}, y_{i5})$. In our work, the training process is very critical and the core of the work. It serves two purposes. Firstly, identify accurate machine learning models and secondly, generate dataset for next stage. For method depicted in Fig. 4, as input we used $data_1$ to train C5.0 classifiers and to make data for training of the next stage. We predicted class labels using $data_2$.

Misclassified data instances are only 2.71% of whole data. Training C5.0 classifier for predicting misclassified (miss) and rightly classified (hit) data labels produced results with high accuracy. However, the prediction accuracy of misclassified data instances is below 50%. This is due to imbalance dataset problem. To counter this, we used SMOTE algorithm [47], to generate balanced and massive data of 1.5 million rows for training C5.0 classifier for predicting miss and hit and update $data_2$ accordingly. Then we separated miss and hit data. Further miss dataset (D_{miss_1}) is used to train classifiers for majority voting. Same steps are repeated to train classifiers n number of times. Class labels which are predicted at different stages are combined together in P_{final} based on row indexes of original input data.

Input: $data_1, data_2$
Output: $Trained\ Models$
1. $\text{ƒ} = \{\ Train\ M_1 = C5.0_{main}(data_1)$
2. $Predict\ P_1 \leftarrow M_1(data_2)$
3. $For(i\ to\ nrow(data_2))$
4. $\{$
5. $If(data_2[class, i]\ != P_1[i])$
6. $\{data_2[status, i] = "miss"\}$
7. $\}$
8. $MJ_1 \leftarrow [C5.0_1(D_{miss_1}), SVM_1(D_{miss_1}), DL_1(D_{miss_1})]$
9. # Accuracy Table of each classifier and Majority Vote
10. $T_1 \leftarrow [P(MJ_1), P(C5.0_1), P(SVM_1), P(DL_1)]$
11. # Function to select classifier or Majority vote based
#on highest classification accuracy and named as M_2
12. $M_2 \leftarrow maxAcc(T_1)$
13. $Predict\ P_2 \leftarrow M_2(D_{miss_1})$
14. $For(i\ to\ nrow(D_{miss_1}))$
15. $\{$
16. $If(D_{miss_1}[class, i]\ != P_2[i])$
17. $\{D_{miss_2} = D_{miss_1}[i]\}$
18. $\}$
20. Repeat from line 8 to 16, n times, incrementally and save
each trained classifier so to use them during testing
21. $Predict\ P_{final} \leftarrow MergeRowsIndexwise(P_1, P_2, P_3, \ldots . P_n)\}$

Fig. 4. Training method.

5.2 Testing

The testing process is explained in Fig. 5, which is self-explanatory in nature. We used classier function, obtained from training process, to predict Y_i label for new $X_i = (r_{i1}, g_{i2}, b_{i3}, x_{i4}, y_{i5})$. All trained classifiers are used in the testing phase. In Fig. 5, from stage-2 all the step repeated n number of times. In this work we used n = 2, however it can be more but will reduce the prediction speed.

6 Results and Analysis

To evaluate our results, we compared D2TFRS method to C5.0 and AdaBoost classifiers. We used confusion matrix, sensitivity, and specificity as the benchmarks for results evaluation.

6.1 Confusion Matrix

A confusion matrix (CM) is a table which shows actual versus predicted data labels. The sum of diagonal (SoD) of CM represents the correctly classified data label, thus can be used to compute classifier accuracy too which can be given as:

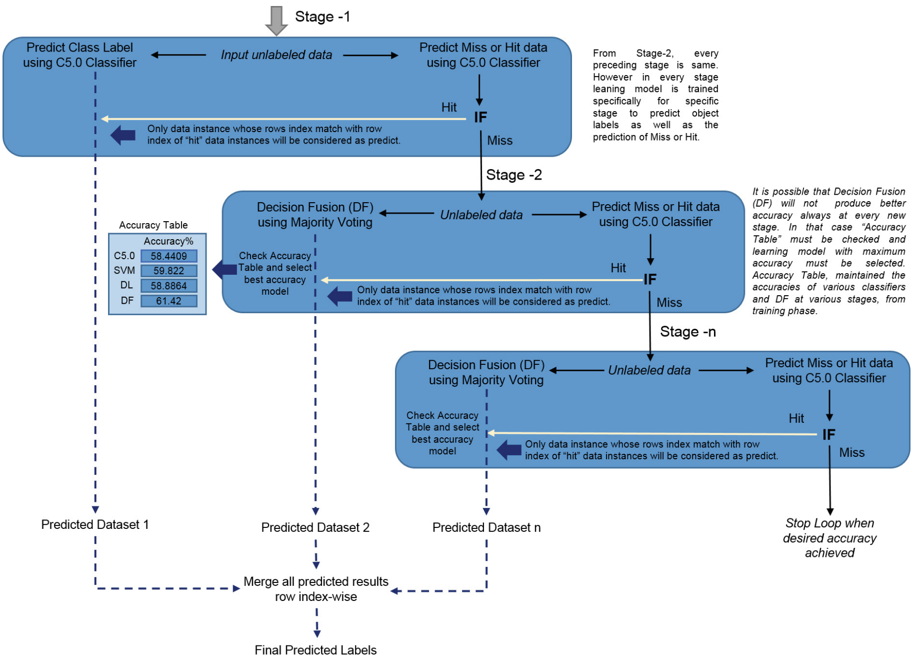


Fig. 5. Block diagram of proposed D2TFRS.

$$Accuracy\% = (SoD/Sum\ of\ all\ cells\ of\ CM) * 100 \tag{1}$$

In Fig. 6, we visualize the CM of the D2TFRS method, C5.0 and AdaBoost classifiers. SoD which is the green color cells in Fig. 6, for each classifier represent rightly classified data labels. D2TFRS outperformed AdaBoost classifier by getting 6.48% better classification accuracy. D2TFRS performed better than C5.0 classifier which produces classification accuracy of 97.29% which is 1.33% less than classification accuracy of D2TFRS.

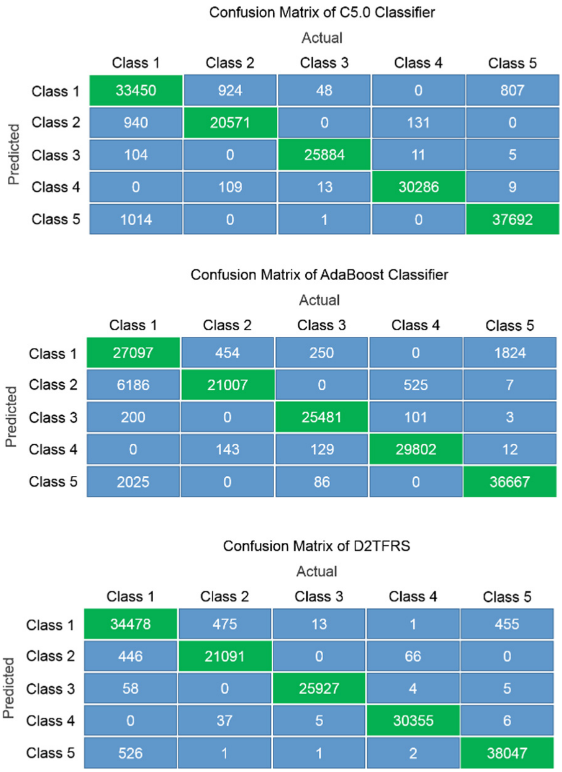
6.2 Sensitivity and Specificity

Sensitivity can be defined as the proportion of actual class labels which are correctly predicted by the classifier. Whereas Specificity is the ability of the classifier to identify negative results. Important terms used to calculate sensitivity and specificity are a number of true positive (TP), number of true negatives (TN), number of false positive (FP) and number of false negatives (FN) respectively.

Mathematically, these can be expressed as:

$$Sensitivity = TP/(TP + FN) \tag{2}$$

$$Specificity = TN/(TN + FP) \tag{3}$$



Confusion Matrix of C5.0 Classifier

Predicted \ Actual	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	33450	924	48	0	807
Class 2	940	20571	0	131	0
Class 3	104	0	25884	11	5
Class 4	0	109	13	30286	9
Class 5	1014	0	1	0	37692

Confusion Matrix of AdaBoost Classifier

Predicted \ Actual	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	27097	454	250	0	1824
Class 2	6186	21007	0	525	7
Class 3	200	0	25481	101	3
Class 4	0	143	129	29802	12
Class 5	2025	0	86	0	36667

Confusion Matrix of D2TFRS

Predicted \ Actual	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	34478	475	13	1	455
Class 2	446	21091	0	66	0
Class 3	58	0	25927	4	5
Class 4	0	37	5	30355	6
Class 5	526	1	1	2	38047

Fig. 6. Confusion Matrix of C5.0, AdaBoost, and D2TFRS.

In terms of sensitivity and specificity, D2TFRS performed better than C5.0 and AdaBoost classifiers for all classes as depicted in Figs. 7 and 8. AdaBoost performed worst among the three whereas C5.0 performed better than AdaBoost but lacks slightly behind proposed D2TFRS. Further, a graphical comparison of sensitivities and specificities are given in Figs. 7 and 8.

6.3 Kappa and Speed

Kappa (κ), is an index that considers an observed agreement with respect to a baseline agreement [48]. κ is a statistical benchmark to measure classification. There are no universal acceptability criteria on how to interpret κ. However first of its kind guidelines are given by Landis and Koch. Value of κ nearer to 1, means substantial or almost perfect agreement. Whereas the value of κ farther from 1 means no agreement or slight agreement. For more detail, characterization of κ can be found in [49].

Mathematically, κ can be expressed as:

$$\kappa = (p_o - p_e)/(1 - p_e) \tag{4}$$

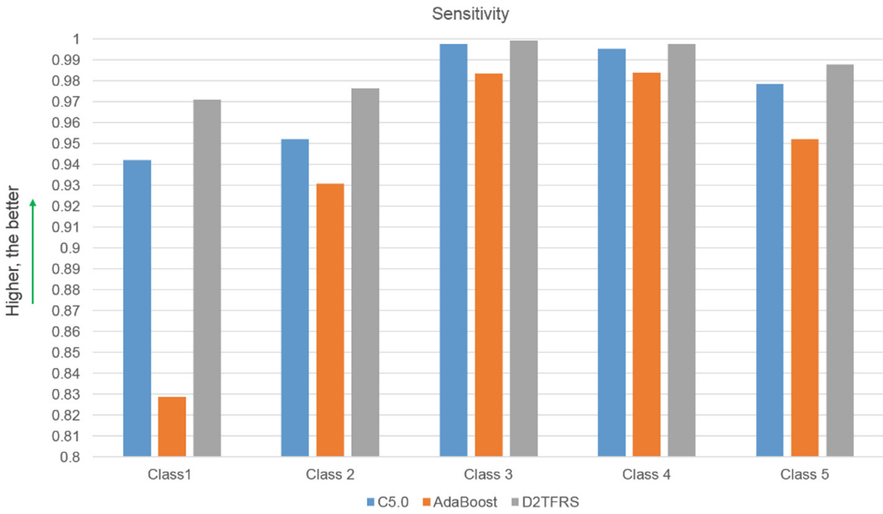


Fig. 7. Sensitivities measurement of C5.0, AdaBoost, and D2TFRS.

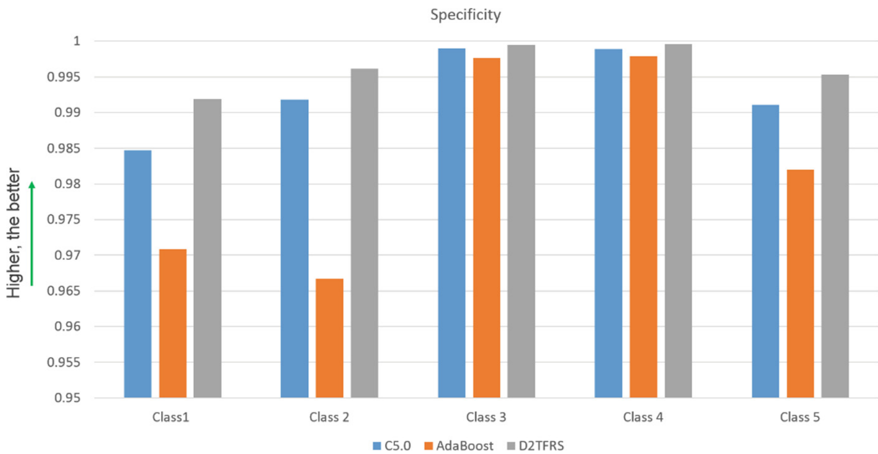


Fig. 8. Specificity measurement of C5.0, AdaBoost, and D2TFRS.

Where an observed agreement is given as p_o, and expected agreement is given as p_e. The value of κ is always ≤ 1. Values of κ are given in Table 1. Proposed method, D2TFRS, has an almost perfect agreement which is nearest to 1 as compared to C5.0 and AdaBoost classifiers.

In terms of speed, C5.0 took 5.11 s which is almost five times faster than D2TFRS which took 24.09 s and AdaBoost took 125 s with 20 iterations for which boosting is run. We strongly believe parallelization can increase the speed of D2TFRS by several magnitudes, as in this work D2TFRS implemented sequentially not in parallel. Further details of accuracy, Kappa and speeds can be found in Table 1.

Table 1. Classification statistics.

	Accuracy%	Kappa	Speed in sec
C5.0	97.29	0.9658	5.11
AdaBoost	92.14	0.9012	125
D2TFRS	98.62	0.9825	24.9

7 Conclusion

Autonomous driving is a near future reality and object recognition will play an important role in it. In this work, we proposed D2TFRS which recognize objects with higher accuracy as compared to C5.0 and AdaBoost. There are two reasons which are responsible for D2TFRS better classification accuracy. Firstly we predicted misclassified data instances prior to classification. Secondly, decision fusion through majority voting is done for reclassifying predicted misclassified data instances only. In this work, D2TFRS is implemented sequentially as a result, it is slower than C5.0. In the future, its speed can also be maximized by several magnitudes by the parallel implementation. We used a small dataset to test D2TFRS which can be considered as a drawback of this work. Therefore further investigation needed to confirm the steady performance of D2TFRS on much larger datasets. In future, we planned to further optimize D2TFRS to make it much faster with higher accuracy by training with massive datasets.

Acknowledgements. The authors acknowledge with thanks the technical and financial support from the Deanship of Scientific Research (DSR) at the King Abdulaziz University (KAU), Jeddah, Saudi Arabia, under the grant number G-661-611-38. The work carried out in this paper is supported by the HPC Center at KAU.

References

1. Litman, T.: Autonomous vehicle implementation predictions implications for transport planning. Transp. Res. Board Ann. Meet. **42**, 36–42 (2015)
2. Morris, D.Z.: Driverless cars will be part of a $7 Trillion Market by 2050 (2017). http://fortune.com/2017/06/03/autonomous-vehicles-market/
3. McGoogan, C.: Uber fires driverless car boss accused of stealing Google's trade secrets (2017). http://www.telegraph.co.uk/technology/2017/05/31/uber-fires-driverless-car-boss-failing-assist-google-lawsuit/
4. Kharpal, A.: Apple has reportedly hired ex-NASA and Tesla staffers to boost its self-driving car effort. https://www.cnbc.com/2017/04/25/apple-driverless-cars-hires-nasa-tesla.html
5. Feris, R.: Tesla sues former Autopilot director for allegedly stealing secrets, poaching coworkers. https://www.cnbc.com/2017/01/26/tesla-sues-former-exec-for-allegedly-stealing-secrets-poaching-workers.html
6. Geiger, A., Lenz, P.: Vision meets robotics: the KITTI dataset. Int. J. Rob. Res. **14**(3), 195–210 (2013)
7. Mehmood, R., Alam, F., Albogami, N.N., Katib, I., Albeshri, A., Altowaijri, S.: UTiLearn: a personalised ubiquitous teaching and learning system for smart societies. IEEE Access. **3536**, 1–22 (2017)

8. Alam, F., Mehmood, R., Katib, I., Albeshri, A.: Analysis of eight data mining algorithms for smarter Internet of Things (IoT). Int. Work. Data Min. IoT Syst. (DaMIS 2016) **98**, 437–442 (2016)
9. Alam, F., Thayananthan, V., Katib, I.: Analysis of round-robin load-balancing algorithm with adaptive and predictive approaches. In: Proceedings of 11th International Conference Control (2016)
10. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
11. Petrovskaya, A., Thrun, S.: Model based vehicle detection and tracking for autonomous urban driving. Auton. Rob. **26**, 123–139 (2009)
12. Wu, B.O., Nevatia, R.A.M.: Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. Int. J. Comput. Vis. **75**, 247–266 (2007)
13. Tsukada, A., Background, A.: Road structure based scene understanding for intelligent vehicle systems. In: Proceedings of 2010 IEEE/RSJ International Conference on Intelligence Robotics System, pp. 5557–5562 (2010)
14. Hu, Q., Wang, P., Shen, C., Porikli, F.: Pushing the limits of deep CNNs for pedestrian detection. Comput. Vis. Pattern Recognit. **28**(6), 1358–1368 (2018)
15. Navarro, P.J., Fernández, C., Borraz, R., Alonso, D.: A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data. Sensors **17**(1), 18 (2017)
16. Harris, M.: New pedestrian detector from google could make self-driving cars cheaper. http://spectrum.ieee.org/cars-that-think/transportation/self-driving/new-pedestrian-detector-from-google-could-make-selfdriving-cars-cheaper
17. Hsu, J.: Deep learning makes driverless cars better at spotting pedestrians. IEEE Spectrum: Technology, Engineering, and Science News. https://spectrum.ieee.org/cars-that-think/transportation/advanced-cars/deep-learning-makes-driverless-cars-better-at-spotting-pedestrians. Accessed 3 Jul 2018
18. Xu, Y., Xu, D., Lin, S., Han, T.X.: Detection of sudden pedestrian crossings for driving assistance systems. IEEE Trans. Syst. Man Cybern. Syst. **42**, 729–739 (2012)
19. Peterson, K., Ziglar, J., Rybski, P.E.: Fast feature detection and stochastic parameter estimation of road shape using multiple LIDAR. In: IEEE/RSJ International Conference on Intelligence Robotics System, pp. 22–26 (2008)
20. Beyeler, M., Mirus, F., Verl, A.: Vision-based robust road lane detection in urban environments. In: Proceedings of 2014 IEEE International Conference on Robotics Automation, pp. 4920–4925 (2014)
21. Felisa, M., Zani, P., Dipartimento, V.: Robust monocular lane detection in urban environments. In: Proceedings of 2010 IEEE Intelligent Vehicles Symposium, pp. 591–596 (2010)
22. Zhou, S., Gong, J., Xiong, G., Chen, H., Iagnemma, K.: Road detection using support vector machine based on online learning and evaluation. In: Proceedings of 2010 IEEE Intelligent Vehicles Symposium, pp. 256–261 (2010)
23. Nair, V., Parthasarathy, N.: Supervised learning methods for vision based road detection. Stanford Univ. (2012)
24. Alam, F., Mehmood, R., Member, S., Katib, I., Nasser, N.: Data fusion and IoT for smart ubiquitous environments : A Survey. IEEE Access **3536**, 1–24 (2017)
25. Xu, P., Davoine, F., Zhao, H., Denoeux, T.: Multimodal information fusion for urban scene understanding. Mach. Vis. Appl. **27**(3), 331–349 (2014)

26. Nuss, D., Thom, M., Danzer, A., Dietmayer, K.: Fusion of laser and monocular camera data in object grid maps for vehicle environment perception. In: Proceedings of 2014 17th International Conference on Intelligent Fusion (2014)
27. Premebida, C., Batista, J., Nunes, U.: Pedestrian detection combining RGB and dense LIDAR data. In: Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robotics System IROS 2014 (2014)
28. Cho, H., Seo, Y., Kumar, B.V.K.V., Rajkumar, R.R.: A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: Proceedings of 2014 IEEE International Conference on Robotics Automation, pp. 1836–1843 (2014)
29. Chumerin, N., Van Hulle, M.M.: Cue and sensor fusion for independent moving objects detection and description in driving scenes. In: Mandic, D., Golz, M., Kuh, A., Obradovic, D., Tanaka, T. (eds.) Signal Processing Techniques for Knowledge Extraction and Information Fusion, pp. 161–180. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-74367-7_9
30. Häne, C., Sattler, T., Pollefeys, M.: Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In: Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robotics System (2015)
31. Zhao, Y., Li, J., Li, L., Zhang, M., Guo, L.: Environmental perception and sensor data fusion for unmanned ground vehicle. Math. Probl. Eng. **2013**, 1–12 (2013)
32. Goebel, K., Yan, W.: Choosing classifiers for decision fusion. GE Global Research Center
33. Fauvel, M., Member, S., Chanussot, J., Member, S.: Decision fusion for the classification of urban remote sensing images. IEEE Trans. Geosci. Remote Sens. **44**, 2828–2838 (2006)
34. Yager, R.R.: A General Approach to the Fusion of Imprecise Information. Wiley, New York (1997)
35. Ye, Z., Bai, L., Tan, L.: Hyperspectral image classification based on gabor features and decision fusion. In: Proceedings of 2017 2nd International Conference on Image Vision Computing, pp. 478–482 (2017)
36. Cheng, J., et al.: Raster: geographic data analysis and modeling. CRAN (2016)
37. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
38. Wu, X., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**, 1–37 (2008)
39. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1999). https://doi.org/10.1007/978-1-4757-3264-1
40. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
41. Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. **3**, e2 (2014)
42. Bengio, Y.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**, 1–127 (2009)
43. Candel, A., Lanford, J., LeDell, E., Parmar, V., Arora, A.: Deep Learning with H2O Deep Learning with H2O (2015)
44. Rosiers, W.: A parallel-voting version of the support-vector-machine algorithm. In: CRAN (2015)
45. Kuhn, M., Weston, S., Coulter, N., Culp, M.: C5.0 decision trees and rule-based models. In: CRAN (2015)
46. Kuhn, M., et al.: Classification and regression training. In: CRAN (2017)
47. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
48. Smeeton, N.C.: Early history of the kappa statistic. Biometrics. **41**, 795 (1985)
49. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**, 159–174 (1977)