

Advanced Driver-Assistance System with Traffic Sign Recognition for Safe and Efficient Driving

Sithmini Gunasekara, Dilshan Gunarathna, Maheshi Dissanayake

Undergraduate at University of Peradeniya

Kandy, Central, Sri Lanka

e15112@eng.pdn.ac.lk

e15111@eng.pdn.ac.lk

maheshid@eng.pdn.ac.lk

Abstract—Advanced Driver-Assistance Systems (ADAS) coupled with traffic sign recognition could lead to safer driving environments. This study presents a sophisticated, yet robust and accurate traffic sign detection system using computer vision and ML, for ADAS. Unavailability of large local traffic sign datasets and the unbalances of traffic sign distribution are the key bottlenecks of this research. Hence, we choose to work with support vector machines (SVM) with a custom-built unbalance dataset, to build a lightweight model with excellent classification accuracy. The SVM model delivered optimum performance with the radial basis kernel, $C=10$, and $\gamma=0.0001$. In the proposed method, same priority was given to processing time (testing time) and accuracy, as traffic sign identification is time critical. The final accuracy obtained was 87% (with confidence interval 84%-90%) with a processing time of 0.64s (with confidence interval of 0.57s-0.67s) for correct detection at testing, which emphasizes the effectiveness of the proposed method.

Keywords-Machine learning; SVM; preprocessing; HOG

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) are the fastest growing intelligent systems utilized in automobile industries as a mechanism to reduce vehicle accidents and fatalities. Modern ADAS are active learning systems which reside inside the automobile. These systems assist the drivers by sharing vital information about traffic laws, congestion, best route and warnings to avoid accidents as well as by judging human driver's mental and physical conditions such as fatigue, sickness, distractions and driving under influence. Extremely advanced ADAS are able to share the information collected within one automobile with other nearby automobiles, and transport management systems to create seamless and safe driving environments. In the modern world, image analysis with machine learning (ML) architectures is replacing image processing based image analysis at a rapid rate. Hence, ML has become a popular area of interest in both research and technical applications for processing and analyzing visual information. More and more everyday object detection tasks are handled by ML. The key reasons for this evolution are the high accuracy of the ML models, easy implementation and analysis process, possibility of fully automating the task, advances in the electronic industry and the availability and accessibility of a high amount of data. Moreover, these algorithms can process multi-dimensional data and comprehend complex algorithms that are difficult to operate manually.

Computer vision is one of the main approaches to obtain information about the surrounding environment of the vehicle in ADAS. With the availability of compact yet improved cameras, capturing information from moving vehicles has become a simple problem. Hence, vision based traffic sign

recognition has become one of the widely sought out tasks in transport system research. Yet, most of these systems still carry room for improvement in regard to low detection accuracy, demand for higher processing power and latency in the decision making process.

In this study we present a sophisticated, yet robust and accurate traffic sign detection system using computer vision and ML, for ADAS. We adopt supervised learning architecture with labeled data collected from dash cameras of Sri Lankan vehicles to train our model. Support Vector Machines (SVM), was proposed as the ML model, owing to both fast execution rate and accuracy of detection with an unbalanced and constrained dataset.

In literature there exist few researches that have attempted to tackle this problem. A high performance multiclass traffic signs detection system based on extended sparse representation classification and cascade tree detector is proposed in [10]. The LIDAR sensor based traffic sign recognition using weakly supervised metric learning was proposed in [11]. The study [12] presents a fast computationally efficient algorithm for this task using histogram of oriented gradient (HOG) and extreme learning machine (ELM). Furthermore, HOG addresses the data unbalance problem which is common in traffic sign recognition as the distribution of signs on the roadside is not uniform. [13] presents a promising architecture based on kernel ELMs with deep perceptual features, which reaches 99.54 % accuracy. A deep neural network based architecture, where data augmentation and region proposal are used to achieve further enhanced recognition is presented in [14]. Furthermore, [15], [16], and [17] present convolutional neural network (CNN) based autonomous and real time traffic sign detection and recognition systems.

In the study presented, we are proposing a traditional machine learning algorithm with Support Vector Machines (SVM) for traffic sign recognition. We have improved the accuracy of the model by investigating the best suited pre-processing method. The algorithm proposed was tested on a database generated using traffic signs in Sri Lanka. The results concluded that, the study has designed a traffic sign recognition model for the local highway system.

II. MATERIALS AND METHODS

A. Support Vector Machine (SVM)

There exists considerable research on various types of classification applications using different architectures, such as artificial neural networks (ANN), Random Forest, k-nearest neighbors (KNN) algorithm, and deep learning architectures such as CNN and Auto encoders. However, since the number of input images per class in our dataset is limited, and the model is expected to perform in real time, we have selected SVM for the research investigated.

SVM is a widely used machine learning algorithm for classification problems. It was introduced by Boser, Guyon, and Vapnik in COLT-92 in 1992 as a linear classifier for supervised learning applications[1]. Later, SVM was re-invented for nonlinear applications with Kernel transformations. At present it is highly utilized in classification and regression problems. In general, SVM maps the input data into a multi dimensional feature space using kernel based mapping. In the feature space, an optimal hyperplane is constructed for each input class with a maximum arithmetical margin between the feature planes of two input classes. Hence, SVM is able to operate with higher predictive accuracy than other ML tools, while automatically avoiding over-fitting.

In order to find the optimum hyperplane for the task, the regularization parameters, kernel type, and the kernel coefficients should be carefully fine tuned. The regularization parameter can control the trade-off between learning error and the complexity of the model.

The kernel is employed in SVM to map the input data into a high-dimensional feature space. It is often the case that data is not necessarily linear and classes are not clearly separable. In such environments, kernels facilitate the non-linear map of data to feature space where the class separability can be handled with clarity, amid reduced computational cost. Different kernel functions such as Radial Basis Function (RBF or Gaussian), linear, polynomial, and sigmoid can be chosen according to the system requirements. Kernel coefficient is a special parameter used with the RBF kernel.

B. Preprocessing and Feature Extraction

The performance of ML based classification systems could be easily improved through pre-processing of the input data as well as through effective feature extraction techniques. Pre-processing is used for organizing and preparing raw data before feeding it to the ML model. Techniques such as

smoothing filters, up-sampling, and sharpening filters are commonly used for this purpose. These techniques enhance the image quality and facilitates efficient feature extraction process.

Another important stage in ML is feature extraction or feature selection, where features/attributes that contribute mostly for the task at hand are extracted from data. The feature selection can be manual or automated. Further, it assists to reduce the raw data into a more manageable size. This eliminates the redundancies in initial data and reduces the amount of data needed to process without losing relevant information.

C. Methodology

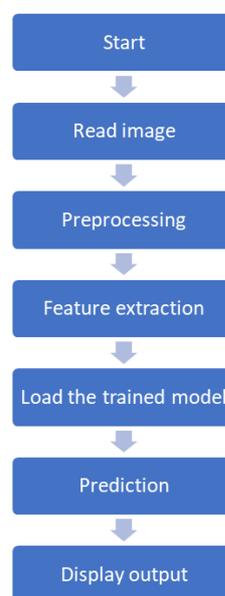


Fig. 1. System Flowchart

Our research proposes a comprehensive solution for traffic sign recognition, with focus given to local sign recognition. Hence, we first created a database consisting of Sri Lankan traffic sign images. The images were extracted from dashboard camera recordings. Considering the restrictions associated with the model training and deployment, such as processing delay and complexity as well as fewer occurrences of some traffic signs (smaller class size), SVM algorithm is utilized as our learning model. Our approach consists of three stages; firstly, localization of the traffic sign from dash camera video feed, secondly, train a model to recognize the sign, and finally deploy the trained model for detection and recognition of traffic signs. The flowchart of the proposed traffic sign recognition system at deployment stage is shown in Fig. 1.

D. Database Generation

One of the key bottlenecks emerged with this research problem is the standardized, yet generalized dataset from the

local traffic environment. As the traffic signs differ from country to country, the recognition system performs differently with varying datasets. Moreover, it is essential for the dataset to cover as many as possible different environment settings such as different illuminations, weather conditions and different perceiving angles of the sign.

The dataset generated for this study consists of images of traffic signs in Sri Lanka. Traffic signs in Sri Lanka can be broadly categorized into two classes, namely; Danger warning signs, which are signs set in yellow background with diamond shape, and Regulatory signs, mostly set in round shaped displays. The Regulatory signs consist of Prohibitory signs, Restrictive signs, Mandatory signs, and Priority signs.

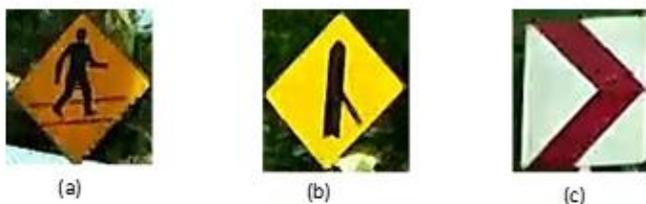


Fig. 2. Sample traffic signs from the dataset: (a) Class 1: Yellow diamond shaped pedestrian crossing sign. (b) Class 0: Yellow Diamond shaped other traffic signs. (c) Class 2: Other traffic signs

The traffic sign images in our dataset were extracted from video recordings from a dashboard camera. These recordings vary in illumination and weather conditions as well as speed setting of the vehicle. The region of interest (ROI), i.e., the traffic sign, and extraction from the video can be automated through YOLO architecture[18]. At the database generation stage some of these traffic signs were manually extracted as well.

In our experimental setup, we test the ability of the trained SVM model to detect a specific traffic sign in the danger warning signs category as this is the most frequently occurring sign type in Sri Lankan roadside. With this view, the extracted signs were categorized into three classes as follows, class 0: Images of yellow diamond shaped pedestrian crossing sign, class 1: Images of yellow diamond shaped other traffic signs, and class 2: Images of any other traffic sign. The database consists of 750 images of Sri Lankan traffic signs with 250 images in each three classes. Fig. 2 shows three images which belongs to these three signs. Note the similarities present between instances in class 0 and 1, and the subtle difference in the supplementary sign within the sign itself.

E. SVM model training, preprocessing, and feature extraction

We adopt SVM based supervised learning architecture as our prediction model and it is trained with customized, labeled, locally collected dataset. In practice, it is often the case that class 0 data is less than class 1 and 2, i.e., there exists more negative examples than positive, which could lead to inaccurate outcomes. To obtain the optimum performance with an unbalanced dataset, it is necessary to select the best hyper parameters for SVM. Two of the key hyper parameters of the SVM algorithm are C parameter and kernel function. The C parameter controls the trade-off between margin maximization

and error minimization, while the kernel parameter provides the non-linear mapping of input data into feature space.

Python sklearn library has the function GridSearchCV which enables hyperparameter optimization for a given dataset. We used this function to determine the regularization parameter, kernel, and the kernel parameter. The highest accuracy was obtained for the radial basis function (RBF) kernel, with $C=10$, and $\gamma=0.0001$. The final SVM model was trained using these best hyper parameters.

In the next step, we tried to improve the accuracy of the model using pre-processing filters and feature extraction techniques. We evaluated our system performance with up sampling, several smoothing filters and a sharpening filter used as pre-processing techniques. The types of smoothing filters used were averaging filter, Gaussian filter, median filter, and bilateral filter. Also, we have tested the model performance with HOG as a feature extraction method against absence of such method. We have refrained from using advance feature extractors such as Gabor filters and pre-processors such as super resolution techniques as these will challenge the real-time deployment of the recognition system.

F. Model testing

Any ML model once trained should be validated. If the model performance is satisfactory, it can be deployed for real time traffic sign detection. At the model testing stage, ROI with traffic sign was extracted from input video by creating a square box marking off the area where the sign is located, and key features of this ROI are extracted without human intervention. Then the unlabeled data is classified as either class 0 or 1 or 2. In practical settings, the correct classification is highly dependent on the input video quality and the ROI extraction stage.

III. RESULTS

In this analysis, accuracy is used as the key performance metric. The accuracy of the SVM classifier is defined as the ratio of the sum of true positives and true negatives to the total number of samples.

Further, the confusion matrix is also presented for the best setting of the model parameters and pre-processing methods to justify the performance. Five-fold cross validation is carried out with training and testing dataset split of 75:25.

The model performance (average accuracy) with and without different pre-processing methods as well as with and without feature extraction are tabulated in TABLE I. The average validation accuracy for the model after only applying hyperparameter optimization was 85.21%. We later applied a combination of pre-processing filters as well as feature extraction techniques to improve the performance accuracy of the proposed model. The SVM algorithm with up sampling, Gaussian smoothing filter, and HOG feature extraction achieved a higher accuracy of 87% (with confidence interval 84%-90%) Yet this approach was time consuming and took 1.64s from feeding in the input to correct classification at the output on a PC with NVIDIA GEFORCE GTX and Intel CORE i7.

TABLE I. AVERAGE ACCURACY FOR PRE-PROCESSING METHODS AND FEATURE EXTRACTION

Method		Average test set accuracy	
Without any filters		0.8521	
With preprocessing	Up sampling	0.7819	
	Smoothing/Sharpening	Averaging	0.8665
		Gaussian	0.8681
		Median	0.8585
		Bilateral	0.8548
Sharpening	0.8338		
With feature extraction	HOG	0.8374	
	HOG, Gaussian smoothing, and upsampling	0.9054	

The confusion matrix obtained without pre-processing and feature extraction is given in Fig. 3 (a), and the confusion matrix for the algorithm including up sampling, Gaussian filter, and HOG is given in Fig. 3 (b). According to Fig. 3, up-sampling, Gaussian filter and HOG improves the classification accuracy in class 0 and 1 by a considerable amount. Fig. 4-6, further illustrates the model performance in terms of classification report, classification error and ROC behavior.

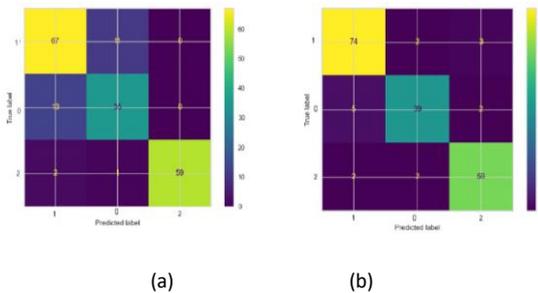


Fig. 3. Confusion matrix for SVM classification: (a) Without pre-processing and up sampling. (b) With pre-processing and up sampling

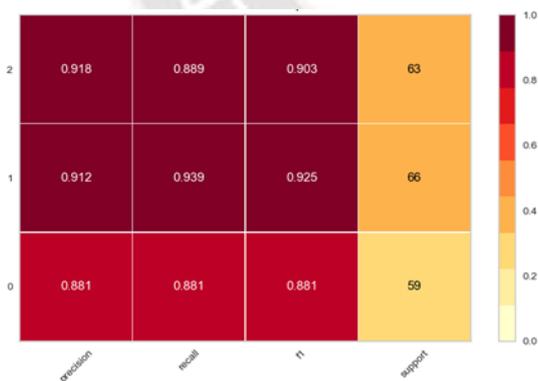


Fig. 4. Classification report for SVM with up-sampling, Gaussian filter and HOG

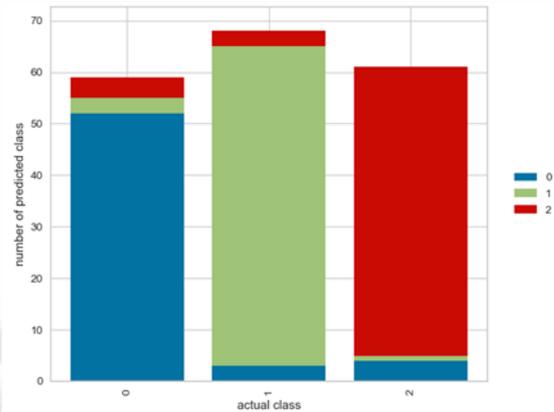


Fig.5. Classification prediction error for SVM with up-sampling, Gaussian filter and HOG

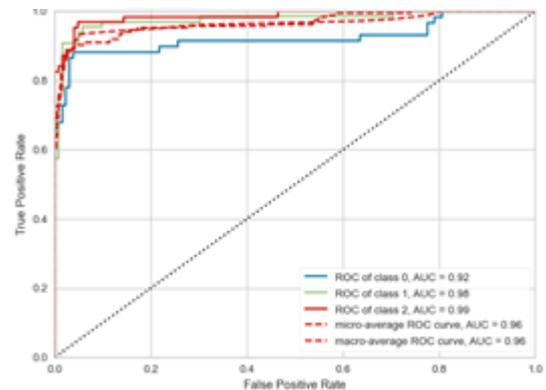


Fig. 6: ROC curves for the classification

IV. DISCUSSION

The model derived from hyperparameter optimization has given an accuracy of 85.21%. This has been improved further up to 90.54% using pre-processing techniques and feature extraction. However, the processing time has increased for this case since pre-processing and feature extraction take additional time to execute. The additional processing time is a result of improved image quality from the preprocessing methods such as upsampling and filtering. Further, by employing only the key features of the input for learning, the model is able to focus on the important features. Hence, the performance of the model is further improved by using feature extraction.

According to TABLE I, the most advantageous pre-processing technique is the Gaussian smoothing filter. HOG has also increased the accuracy. Therefore, we included them in our SVM model. However, we decided to include upsampling as well because it improved the accuracy in the model when it was used along with the Gaussian filter and HOG, even though it shows reduced accuracy in TABLE I. The reason for this could have been distortions in image boundaries when upsampling was used alone.

In Fig. 2 (a), the confusion matrix shows that the model gives a considerable number of wrong predictions regarding classes 0 and 1. These two classes have images of same shape and color. However, Fig. 2 (b) shows that the model has

improved its performance with pre-processing and feature extraction, improving the ratio of correct predictions.

V. CONCLUSIONS

In this study, we have investigated a suitable machine learning algorithm for traffic sign recognition. Our approach was based on the SVM algorithm from traditional machine learning architectures. A database consisting of Sri Lankan traffic sign images was created using the frames from dashboard camera recordings. An SVM model was trained using this data set with optimum hyperparameter values. The final accuracy obtained was 90.54% with a processing time of 0.64s.

This model can be modified in order to achieve higher accuracy in much lower processing time. The greatest challenge of the research question is providing live feedback through active recognition of traffic signs from speeding vehicles. For this purpose, the model should be further improved to complete the analysis under extreme low delay. Further modifications of the system should account for robustness under the illumination variations of images due to the time of the day and weather conditions. The distortions in the images due to speed of the vehicle and partial occlusion could also be considered in such a system. Finally, this model can be developed to use in a local traffic system as an effective driver assistant system.

REFERENCES

- [1] E. Mayoraz and E. Alpaydin, "Support Vector Machines for Multi-class Classification", Lecture Notes in Computer Science, vol. 2, pp. 833-842, 1999.
- [2] S. Lessmann, R. Stahlbock and S. Crone, "Optimizing Hyperparameters of Support Vector Machines by Genetic Algorithms", Proceedings of the 2005 International Conference on Artificial Intelligence, 2005, Las Vegas, Nevada, USA.
- [3] L. Padierna, M. Carpio, A. Rojas, H. Puga, R. Baltazar and H. Fraire, "Hyper-Parameter Tuning for Support Vector Machines by Estimation of Distribution Algorithms", Nature-Inspired Design of Hybrid Intelligent Systems, 2017.
- [4] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series", 1997.
- [5] C. S. GUREL, "Traffic Sign Recognition using multi-class SVM", 2018.
- [6] N. Hasan, T. Anzum and N. Jahan, "Traffic Sign Recognition System (TSRS): SVM and Convolutional Neural Network", 4th International Conference on Inventive Communication and Computational Technologies, 2020.
- [7] "Support Vector Machine — Introduction to Machine Learning Algorithms", Medium, 2020. [Online]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- [8] Rahul Kala, "4 - Advanced Driver Assistance Systems," Editor(s): Rahul Kala, On-Road Intelligent Vehicles, Butterworth-Heinemann, 2016, Pages 59-82.
- [9] Krishnarao S., Wang HC., Sharma A., Iqbal M. (2021) Enhancement of Advanced Driver Assistance System (Adas) Using Machine Learning. In: Yang XS., Sherratt R.S., Dey N., Joshi A. (eds) Proceedings of Fifth International Congress on Information and Communication Technology. ICICT 2020. Advances in Intelligent Systems and Computing, vol 1183. Springer, Singapore.
- [10] Liu, C., Chang, F., Chen, Z., Liu, D.: Fast traffic sign recognition via high-contrast region extraction and extended sparse representation. IEEE Trans. Intell. Transp. Syst. 17, 79–92, 2016.
- [11] Tan, M., Wang, B., Wu, Z., Wang, J., Pan, G.: Weakly supervised metric learning for traffic sign recognition in a LIDAR-equipped vehicle. IEEE Trans. Intell. Transp. Syst. 17, 1415–1427, 2016
- [12] Huang, Z., Yu, Y., Gu, J., Liu, H.: An efficient method for traffic sign recognition based on extreme learning machine. IEEE Trans. Cybern. 47, 920–933, 2017
- [13] Zeng, Y., Xu, X., Shen, D., Fang, Y., Xiao, Z.: Traffic sign recognition using kernel extreme learning machines with deep perceptual features. IEEE Trans. Intell. Transp. Syst. 18, 1647–1653 (2017)
- [14] Hoang VD., Le MH., Tran T.T., Pham VH. (2018) Improving Traffic Signs Recognition Based Region Proposal and Deep Neural Networks. In: Nguyen N., Hoang D., Hong TP., Pham H., Trawiński B. (eds) Intelligent Information and Database Systems. ACIIDS 2018. Lecture Notes in Computer Science, vol 10752. Springer, Cham.
- [15] Alghmgham, Danyah A., Ghazanfar Latif, J. Alghazo and Loay Alzubaidi. "Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN." Procedia Computer Science 163 (2019): 266-274.
- [16] Shustanov, A. and P. Yakimov. "CNN Design for Real-Time Traffic Sign Recognition." Procedia Engineering 201 (2017): 718-725.
- [17] A. Avramović, D. Tabernik and D. Skočaj, "Real-time Large Scale Traffic Sign Detection," 2018 14th Symposium on Neural Networks and Applications (NEUREL), 2018, pp. 1-4
- [18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," CoRR, vol. abs/1506.02640, 2015