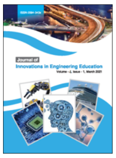

Journal of Innovations in Engineering Education
JIEE 2022, Vol. 5, Issue 1, pp 47-54.

https://doi.org/10.3126/jiee.v5i1.43682

THAPATHALI CAMPUS
Institute of Engineering
Tribhuvan University

Design and analysis of computer vision techniques for object detection and recognition in ADAS

Grishma Ojha[a,*], Diwash Poudel[a], Jibit Khanal[a], Niraj Pokhrel[a] and Shanta Maharjan[a]

[a] *Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Thapathali, Kathmandu, Nepal*

ARTICLE INFO

Article history:

Received 6 March 2022
Received in revised form 11 April 2022
Accepted 2 June 2022

Keywords:

ADAS
Artificial Neural Network
Convolutional Neural Network
Feature Engineering
Lane Finding
Machine Learning
Object detection
RC car

Abstract

The rising number of road accidents has been an increasing concern in recent years in the world. To ensure overall road safety, there is a need for a reliable system that could assist drivers to avoid road accidents. This study aims to find, analyze and develop reliable techniques and algorithms that can ensure the safety of the drivers by minimizing the factor of human error in most road accidents. By focusing on the above concerns, we proposed a system in which an electronic device is used in the vehicles to assist drivers by providing accurate and reliable data about the road environment with the help of different sensors, ensuring overall road safety. The system uses a camera module for visual feed to the system and has an ultrasonic sensor to sense different obstacles on the rear side of the model car. Two convolutional neural networks (CNNs) have been compared and the best one has been selected for object detection and recognition in this system. The system has the feature of alerting the driver if the vehicle departs from its circumscribed lane. The system is effective in applications like blind-spot monitoring, forward collision warning, and so on.

1. Introduction

With the increase in the usage of vehicles, the number of road accidents has also increased exponentially. Among the different causes, the carelessness of the driver is responsible for the most number of road accidents. The use of robotics and automation along with information technology has proven to be effective in designing systems that could reduce road accidents. The human errors and limitations during driving are compensated by the usage of the Advanced Driver Assistance System (ADAS). ADAS refers to various high-tech in-vehicle systems that are designed to increase road traffic safety by helping drivers gain better awareness of the road and its potential hazards as well as other drivers around them[1]

In order to let the individuals driving motor vehicles become capable of avoiding potential risks and dangers, it is desirable to have a highly accurate ADAS system to assess any situation in real-time while driving and alert the driver, and at times take certain decisions. A group of applications such as the detection of lanes, obstacles, traffic signs, and many more in real-time driving environments can be detected just by perceiving their positions, although there exist certain limitations where not all these applications are favored by drivers [2].

The ADAS is constantly evolving in the world with the integration of more advanced features to ensure overall traffic safety. The major aim of the ADAS is to help the driver with its warning mechanisms to the probability of accidents detected by the system. The ADAS can also automate several functionalities of the vehicle reducing the pressure of vehicle control from the driver. Demand for ADAS in vehicles is expected to increase over the next few decades for regulating and monitoring different control mechanisms of the vehicles to protect drivers and

*Corresponding author:
: gojha18@gmail.com (G. Ojha)

reduce accidents. The current industrial implementation of ADAS is restricted to high-end vehicles [3], while this paper aims at discussing the design of an ADAS system that could be implemented in vehicles at a cheaper cost as well.

The use of Artificial Intelligence (AI) and Machine Learning (ML) in ADAS is taking us closer to designing and implementing reliable and efficient autonomous vehicles. The use of multiple cameras and sensors helps the ADAS to detect the objects and the machine learning algorithms used in the system help in perceiving such objects effectively.

ML is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead [4]. A subset of ML is Deep Learning (DL) which is an approach to AI, based on the creation of several layers, with a final graph characterized by a depth [5]. The Artificial Neural Network (ANN) is the basis of deep learning that adopts the principle of functioning of the human brain and its biological neural networks. There are multiple nodes in ANN that perform similar functionalities of the biological neuron. Each node in the ANN is interconnected with the other to transmit information. Neurons are connected in various patterns, to allow the output of some neurons to become the input of others. The network forms a directed, weighted graph [6].

2. Related Works

The ADAS is being extensively used in different kinds of vehicles in different scenarios to ensure overall road safety and reduce the occurrence of accidents due to human errors. This system was adopted for the first time in the 1950s in the form of an anti-lock braking system [7]. The initial ADAS made use of sensors and advanced electronic circuitries for functionalities like lane departure warning, anti-lock braking, blind-spot detection, and so on. The use of Light Detection and Ranging (LiDAR), laser, camera-based sensors, etc. provided information regarding the obstacles in front and around the vehicle.

In modern vehicles, ADAS is equipped with visual recognition system (VRS) that performs image classification, object detection, segmentation, and localization for basic ocular performance [8]. Object detection is emerging as a subdomain of computer vision (CV) that benefits from DL, especially CNNs [9]. The use of concepts like DL and CV have made the design and implementation of algorithm-based and data-driven ADAS possible in commercial vehicles. The autonomous vehicles designed by Carnegie Mellon University and the Defense Advanced Research Project Agency (DARPA) have played a major role in the development of ADAS and autonomous vehicles [10]. The Auto-pilot technology developed by Tesla Motors for its implementation in their vehicles could predict the collisions with high accuracy of 76% and prevented the probable collisions with a prevention rate of over 90%.The railway systems like SkyTrain of Canada and Yurikamome of Japan make use of advanced technologies to make the self-driving of trains possible. With the improvement of the accuracy of CNN in the detection and recognition of the object at a very fast rate, the use of CNN or DL for real-time image processing the vehicles has been possible. In computer vision, CNN-based semantic segmentation is one of the most complex tasks, which provides a core aspect for traffic scene understanding. Compared to the image classification task, this process consists of densely classifying all the pixels in an image (i.e. pixel-wise prediction) by predicting a spatial mask of a segmented ROI, which implies a significant consumption of hardware resources. In addition to the road detection task, many studies, based on semantic segmentation, have been proposed to address some realistic challenges, such as autonomous driving, robotic navigation, etc. Generally, fully convolutional networks (FCNs) [11] are the most common deep CNNs architectures used to generate dense predictions by feeding arbitrarily-sized input data into them. Traditionally, traffic sign identification has been based on color and shape patterns, with two associated stages: detection and classification [12, 13]. After many pre-processing processes, such as data transformation and normalization, which consists of extracting regions of interest (ROI) based on color segmentation and "sliding window" fashion, traffic signs are detected in the image. Following the pattern recognition stage, the classification stage involves classifying each sign feature into categories such as 'speed limits', 'pedestrian crossing', and so on. The template matching technique was used to improve the feature classification process in [12]. The probable traffic indicators are then classified using a shallow neural network (i.e., a multi-layer perceptron (MLP)). The authors of [13] presented a hardware architecture that uses a template matching method to classify traffic indicators.

The design of ADAS discussed in this paper uses DL for image processing and recognition to make it an accurate, reliable, and cheaper alternative to the costly commercial ADAS.

3. Methodology

This section explains the methodologies used in the system's design and implementation

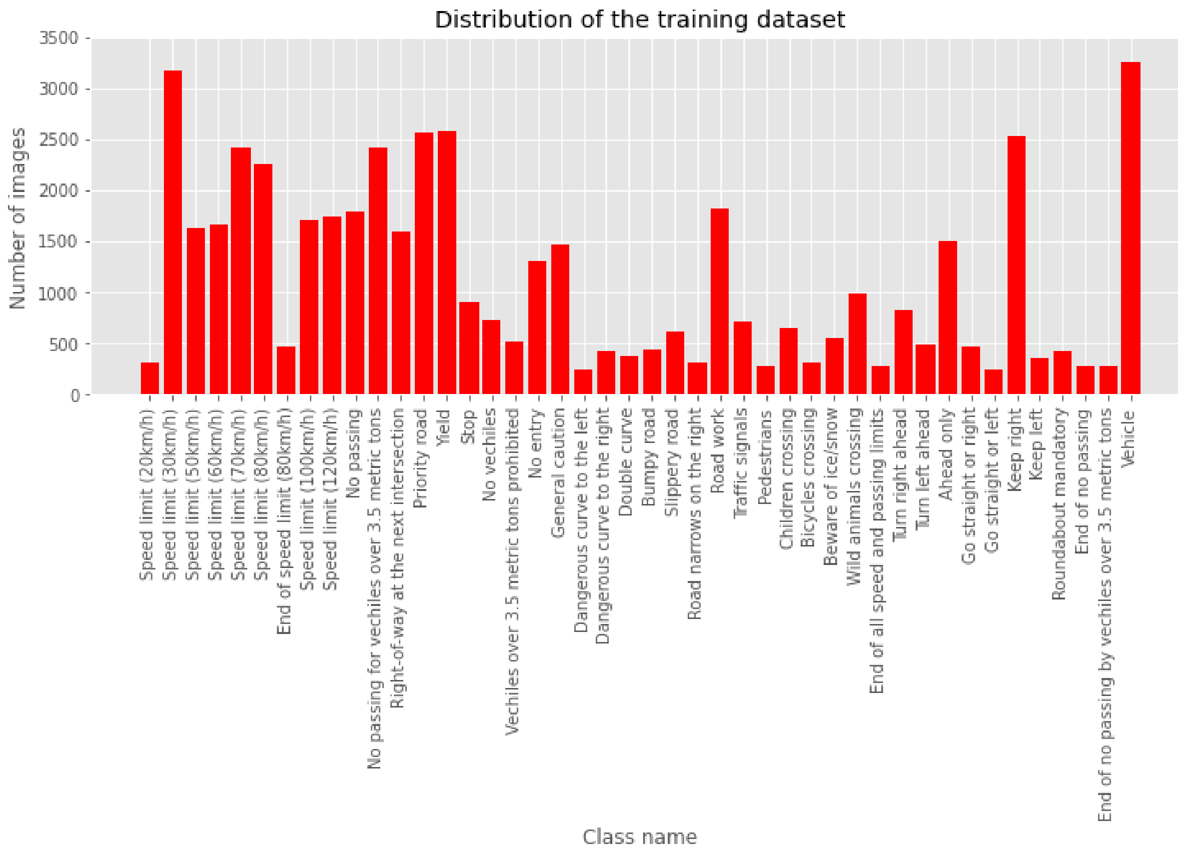

Figure 1: Distribution of dataset

3.1. Dataset for deep learning algorithm

The dataset we have used is a combination of two datasets; they are The German Traffic Sign Benchmark and The Vehicle Image Database provided by the Polytechnic University of Madrid. The German Traffic Sign Benchmark is used to detect road signs as well as identify the meaning of road signs. It contains more than 50,000 roads sign images and is divided into 43 classes. A combination of these two datasets is used to detect traffic signs as well as vehicles on the road.

This is the distribution of a combined dataset. There are a total of 44 classes. Each class contains a different number of images.

3.2. System block diagram

All I/0 functions are controlled by the Raspberry Pi, which serves as the main processor. All of the electronic parts can be stored inside the model RC car, which can be controlled via an Android application. The ultrasonic sensor provides information about the obstacle's proximity, whilst the Pi camera provides information to the Raspberry Pi in video format.

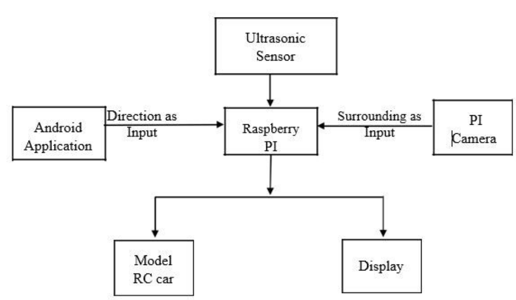

Figure 2: system block diagram

4. Instrumentation and Software Used

Model RC car uses Raspberry Pi 4 as the CPU in the system. An ultrasonic sensor is used to detect obstacles along with the distance information and the Raspberry Pi camera module is used for visual feed. To control the dc motor, the L298N motor driver is used. Geany IDE is used to program Raspberry Pi 4. Jupyter notebook and Pycharm are used to run Deep learning and OpenCV codes. To quickly train the deep learning modelGoogle Colab is used since it provides GPU to train neural network.

5. Implementation Details

The details involved in putting the project plan into action are described in the section below.

5.1. Working of the lane finding system

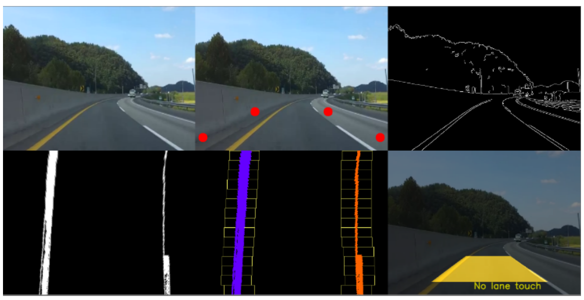

Figure 3: Split-screen working of lane finding system

Red dotted warp points are placed in the original image for perspective transformation. Canny edge detection is used for finding the border of lane lines. The Canny edge detection technique produces smoother edges [14], which is the reason it is chosen among different edge detection techniques. The canny filter works in different stages. The filter that it uses is based on Gaussian's derivative, which is used to calculate the gradients' intensity. The Gaussian decreases the noise found in the image. Thinning down of potential edges is done to 1-pixel curves by removing gradient magnitude's non-maximum pixels [15]. Finally, edge pixels are kept or removed using hysteresis thresholding on the gradient magnitude. Then, perspective transformation along with Polygon fitting to track the lane line as shown in Figure 3

5.2. Object detection system

For detecting and classifying objects such as traffic lights and vehicles we have tested two CNN models that we named model 1 and model 2 with the different hyperparameters. CNN is a type of deep neural network which is used to analyze visual imagery. Because of its self-learning capabilities, it uses little preprocessing to optimize its filters [16].

All the convolutional layers of CNN model 1 are shown in the Figure 4. We used images of size $32 \times 32 \times 3$ where 32 is the height, 32 is the width, and 3 resembles the three channels, i.e., RGB. Filters of size 5×5 are used, and the output of using these filters to input gives an output of size $28 \times 28 \times 30$, where 28 is the height, 28

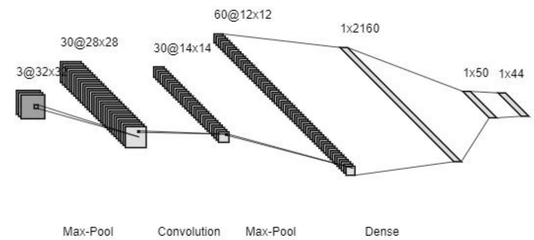

Figure 4: CNN Architecture of model 1

is the width, and 30 is the number of filters. The max-pooling of the size of 2×2 is applied to conv2d_1 of size $28 \times 28 \times 30$ and the output comes as conv2d_1 of size $14 \times 14 \times 30$. Similarly, 60 filters (kernels) of size 2×2 are used on conv2d_1, and the output of using these filters to input gives the output of size $12 \times 12 \times 60$. Then, the max-pooling of the size of 2×2 is applied to conv2d_1 of size $12 \times 12 \times 60$, and the output comes as conv2d_1 of size $6 \times 6 \times 60$. Then, a dropout value of 0.5 is used because it improves the performance of neural networks by reducing overfitting [17]. The data that is the output of dropout is flattened to 2160 and fed into an ANN with a dense layer of 50 nodes. Activation function Relu and dropout 0.5 are used between the Dense layer. Since the output has 44 classes, we used a dense layer of 44 nodes and Softmax as the final activation function. Relu is used between dense layers because it removes the problem in backpropagation called vanishing gradients [18].

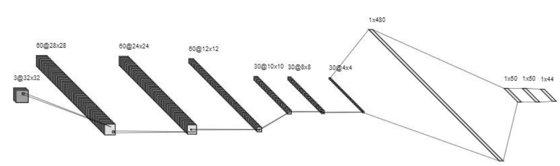

Figure 5: CNN Architecture of model 2

In Figure 5, all the convolutional layers of CNN model 2 are depicted. We used images of size $32 \times 32 \times 3$ similar to what we did in model 1. 60 filters (kernels) of size 5×5 are used and the output of using these filters to input gives the output of size $28 \times 28 \times 60$ where, 28 is the height, 28 is the width and 60 is the number of filters. Again a convolutional layer of size $24 \times 24 \times 60$ is used. The max-pooling of the size of 2×2 is applied to conv2d_1 of size $24 \times 24 \times 60$ and the output comes as conv2d_1 of size $12 \times 12 \times 60$. Similarly, 30 filters (kernels) of size 2×2 are used on conv2d_2, and the output of using these filters to input gives the output of size $10 \times 10 \times 30$. Again a convolutional layer of size $8 \times 8 \times 30$ is used. The max-pooling of the size of 2×2 is applied to conv2d_1 of size $8 \times 8 \times 30$ and the output is of size $4 \times 4 \times 30$. Then a dropout value of 0.5 is used. The data that is the output of dropout is flattened to 480 and fed into an ANN with a dense layer of 50 nodes. After that, we used the activation function Relu and dropout 0.5 between the Dense layer. Finally, a Dense layer of 44 nodes is used along with Softmax as the final activation function.

6. Result

6.1. Output of lane detection

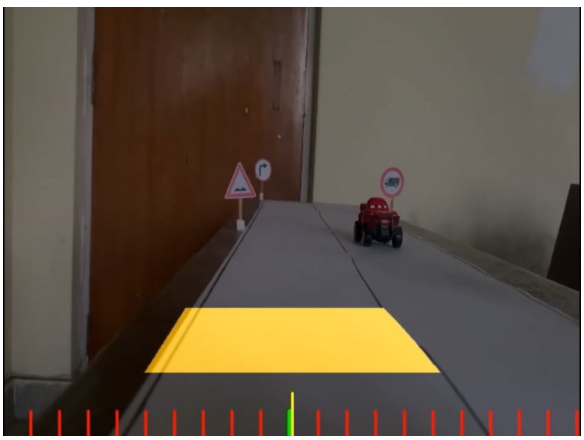

Figure 6: Implementation of the lane detection system

Figure 6 is the output obtained from our lane detection system. The Lane Detection System was able to detect lanes correctly.

Figure 7: No lane detection

The lane lines are not detected in some cases during the observation and we concluded that the reason behind this is the absence high-end ultra-wide camera in our system.

6.2. Output of object detection

Figure 8 depicts the training and validation losses of the two models in each epoch. Model 1 has training and validation loss 1.1207 and 0.3914 respectively which is higher than the loss values in Model 2 that has training and validation loss of 0.0221 and 0.0055 respectively.

Figure 9 shows the training and validation losses of the two models in each epoch and figure 10 compares the validation accuracy. The training and validation accuracy of model 1 is 0.641 and 0.8805 respectively, which is far lower than that of model 2 which has training and validation accuracy of 0.9912 and 0.9985 respectively.

As observed in Figure 10, Model 2 is superior compared to Model 1 because it has higher accuracy of 99.85% compared to 88.05% of Model 1. Other research like [19] and [20] used a very similar German Traffic Sign Benchmark dataset and got an accuracy of 99.15% and 99.61%, respectively, which is very close to ours.

Table 1: Performance metrics comparison of TWO CNN MODELS

CNN Models	Precision	Recall	F1
Model 1	0.9762	0.8101	0.7494
Model 2	0.9792	0.9281	0.8904

According to Table 1, we can observe that Precision, Recall, and F1 scores are also high on Model 2 than on Model 1. From all the performance metrics, we can conclude that the object detection system made from Model 2 does a much better job than Model 1.

6.3. Output of lane and objection detection system combined

Figure 11 presents the final output of our lane as well as the object detection system in action. Here yellow rectangular box shows the lane line which is detected by our lane detection system. The green bounding box is drawn around the traffic sign, which is detected by our object detection system. After identifying the traffic sign, our system also showed in which class traffic sign belongs, and showed its class as '80km/hr.' to the user.

6.4. Output of mobile application

Figure 12 above shows the user interface of our mobile application. It is used to control the model car with the help of buttons for respective directions.

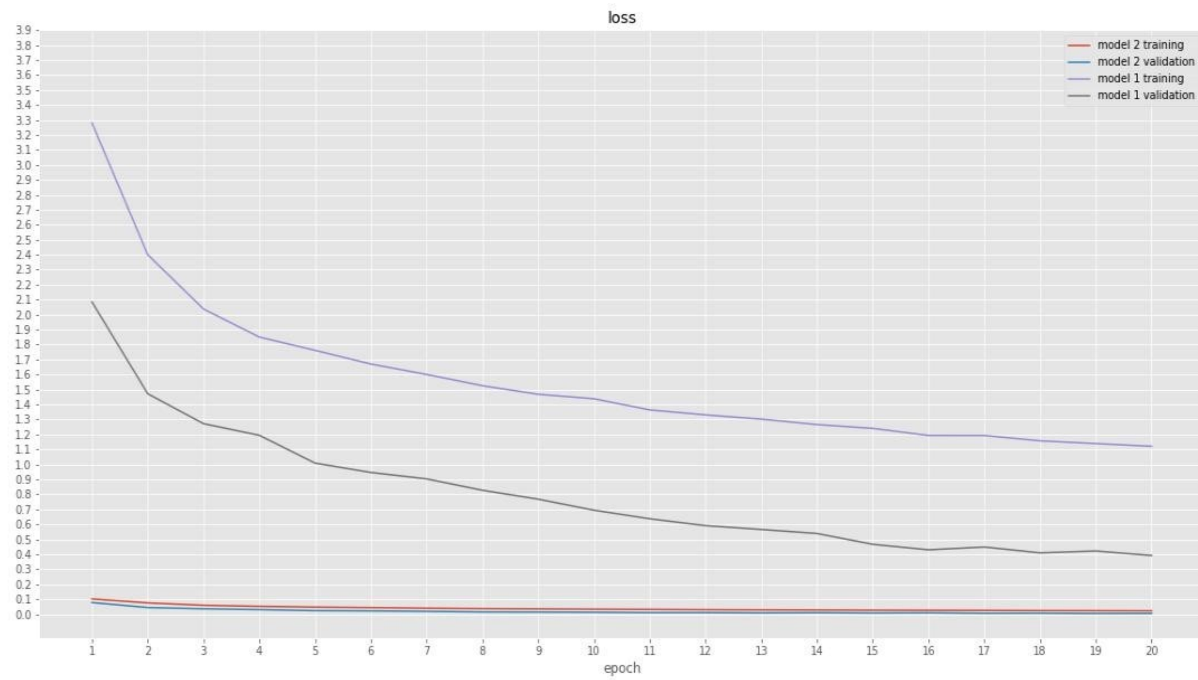

Figure 8: Loss comparison between two models in each epoch

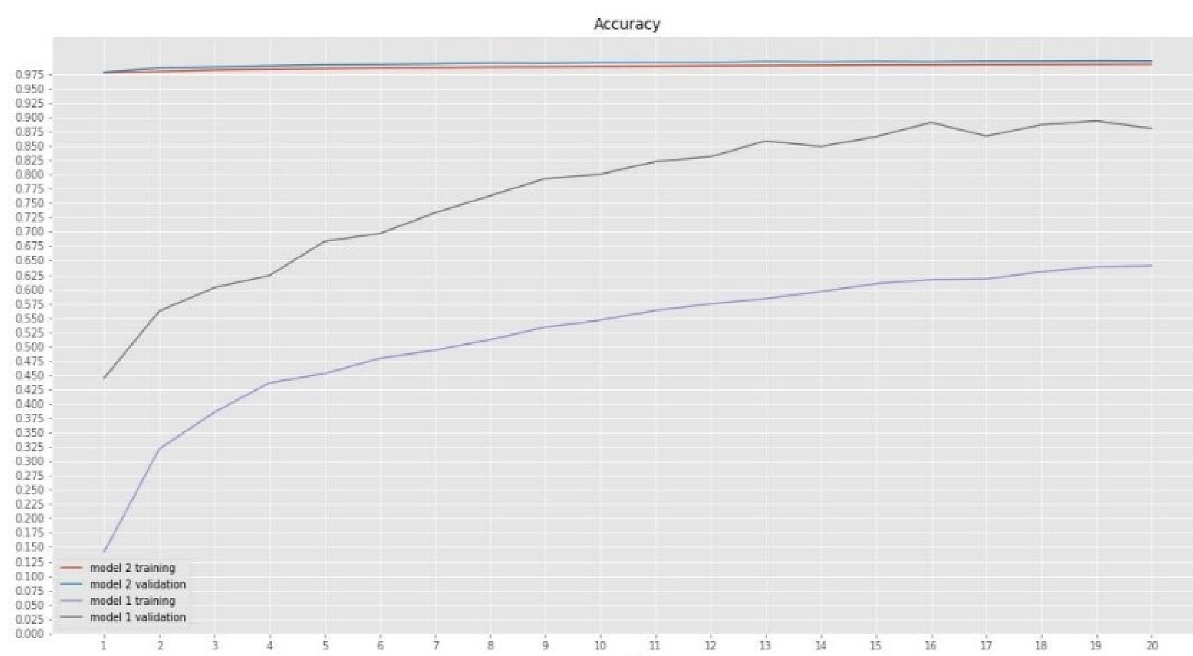

Figure 9: Accuracy comparison between two models in each epoch

7. Future Enhancement

The system designed in this project is not highly accurate and applicable in the commercial vehicles due to the limitations of processing capability of the processor used in the system. Our system is valuable for study purposes to understand the basic functionalities of ADAS. Our system needs to be improved to make it effective and efficient in real-life scenarios using advanced components. The components and processors used in the system are not advanced enough to provide

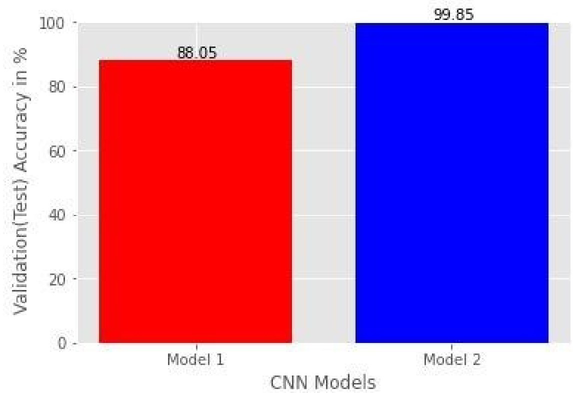

Figure 10: Accuracy comparison between two models

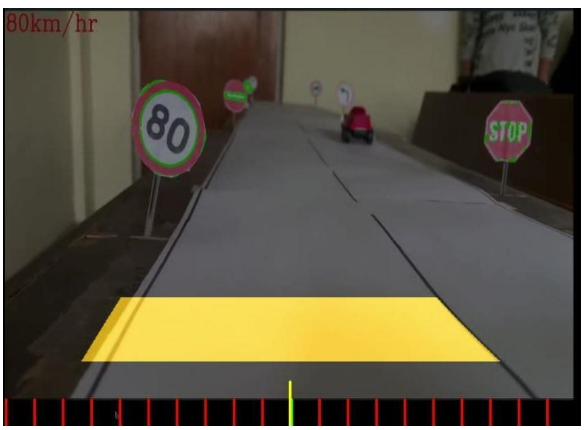

Figure 11: .The final output of the lane and object detection system

faster or real-time processing for object detection and recognition needed in the commercial ADAS.

8. Conclusion

The area of ADAS and autonomous vehicles is continuously expanding, and technological improvements are being made at a great pace. The margin for error of these systems is very less in the commercial application of ADAS and there is always some room for improvement in the designed system. The results obtained from the design of this system are satisfactory. The CNN Model 2 is found to be superior compared to the Model 1 because of its higher performance metrics scores. In perfect scenarios, the lane detection system also performed well. However, if there was an imperfect field of view, the detection system was found to be unreliable. At long last, all the functionalities of the system like object detection system, lane-finding system, and Android application were working significantly well, and the outcome of the overall system was satisfactory.

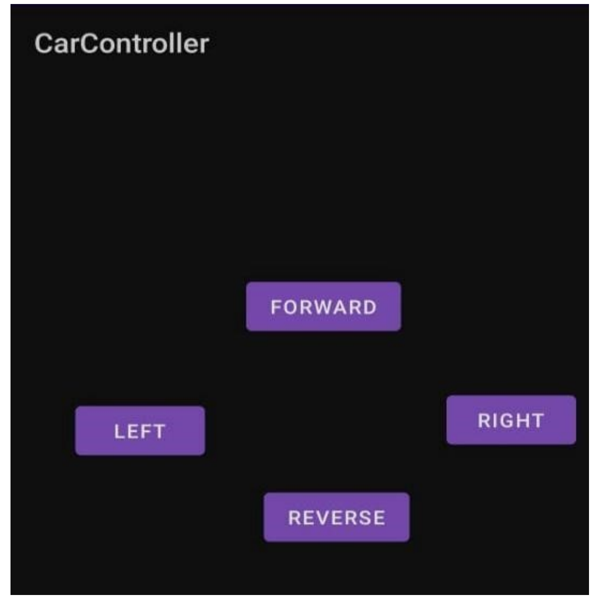

Figure 12: The final output of the lane and object detection system

Acknowledgment

We would like to thank the Department of Electronics and Computer Engineering, Thapathali campus for providing the necessary resources and guidelines to successfully execute this project. We express our deepest gratitude to Assoc. Prof. Shanta Maharjan for his commitment, support, encouragement, and guidelines in this project. We thank all the critics for their valuable feedback on this paper.

Conflict of interest

No conflict of interest

References

[1] Hoferlin B, Zimmermann K. Towards reliable traffic sign recognition[C/OL]// 2009 IEEE Intelligent Vehicles Symposium. IEEE, 2009: 324-329. DOI: https://doi.org/10.1109/IVS.2009.5164298.

[2] Lijarcio I, Useche S A, Llamazares J, et al. Availability, demand, perceived constraints and disuse of adas technologies in spain: Findings from a national study[J/OL]. IEEE Access, 2019, 7: 129862-129873. DOI: https://doi.org/10.1109/ACCESS.2019.2939302.

[3] D. Larsen R K. Mercedes-benz "intelligent drive": Adas technologies and rollout strategy in us market[J]. 2013.

[4] Nilsson N. Introduction to machine learning[M/OL]. Cambridge, 2019. DOI: https://ai.stanford.edu/~nilsson/MLBOOK.pdf/.

[5] Goodfellow I, Bengio Y, Courville A. Deep learning[M/OL]. MIT press, 2016. https://www.deeplearningbook.org/.

[6] Zell A. Simulation neuronaler netze: volume 1[M]. Addison-Wesley Bonn, 1994.

[7] Galvani M. History and future of driver assistance[J/OL]. IEEE Instrumentation & Measurement Magazine, 2019, 22(1): 11-16. DOI: https://doi.org/10.1109/MIM.2019.8633345.

[8] Brunetti A, Buongiorno D, Trotta G F, et al. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey[J/OL]. Neurocomputing, 2018, 300: 17-33. DOI: https://doi.org/10.1016/j.neucom.2018.01.092.

[9] Vodrahalli K, Bhowmik A K. 3d computer vision based on machine learning with deep neural networks: A review[J/OL]. Journal of the Society for Information Display, 2017, 25(11): 676-694. DOI: https://doi.org/10.1002/jsid.617.

[10] Haidar L. medium[J/OL]. 11 Sep. 2021. https://medium.com/codex/sobel-vs-canny-edge-detection-techniques-step-by-step-implementation-11ae6103a56a/.

[11] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[12] Hechri A, Hmida R, Mtibaa A. Robust road lanes and traffic signs recognition for driver assistance system[J]. International journal of computational science and engineering, 2015, 10 (1-2): 202-209.

[13] Hmida R, Ben Abdelali A, Mtibaa A. Hardware implementation and validation of a traffic road sign detection and identification system[J/OL]. Journal of Real-Time Image Processing, 2018, 15 (1): 13-30. DOI: https://doi.org/10.1007/s11554-016-0579-x.

[14] Ryan M. The future of transportation: ethical, legal, social and economic impacts of self-driving vehicles in the year 2025[J/OL]. Science and engineering ethics, 2020, 26(3): 1185-1208. DOI: https://doi.org/10.1007/s11948-019-00130-2.

[15] Maitra S. medium[J/OL]. 11 Sep. 2021. https://medium.com/codex/sobel-vs-canny-edge-detection-techniques-step-by-step-implementation-11ae6103a56a/.

[16] Valueva M V, Nagornov N, Lyakhov P A, et al. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation[J/OL]. Mathematics and computers in simulation, 2020, 177: 232-243. DOI: https://doi.org/10.1016/j.matcom.2020.04.031.

[17] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

[18] Jadhav P. medium[J/OL]. 13 Jun. 2021. https://medium.com/geekculture/relu-vs-sigmoid-5de5ff756d93/.

[19] Cireşan D, Meier U, Masci J, et al. A committee of neural networks for traffic sign classification[C/OL]// The 2011 international joint conference on neural networks. IEEE, 2011: 1918-1921. DOI: https://doi.org/10.1109/IJCNN.2011.6033458.

[20] Zhang J, Wang W, Lu C, et al. Lightweight deep network for traffic sign classification[J/OL]. Annals of Telecommunications, 2020, 75(7): 369-379. DOI: https://doi.org/10.1007/s12243-019-00731-9.