Real Time Traffic Light Detection by Autonomous Vehicles using Artificial Neural Network Techniques

Mahesh .G, Satish Kumar .T

***Abstract:** Autonomous vehicles are the reality of the future, they will open up the way for future advanced systems where computers are expected to take over the decision making of driving. These automobiles are capable of sensing their environment and moving with little or no human input. The main goal of this research is to detect traffic light in real-time for autonomous vehicles. Apart from taking decisions to navigate in the right manner the autonomous vehicles important task is to detect traffic lights, so that it can obey the traffic rules with sufficient precision. The work carried out in this research makes use of two Artificial Intelligence technique, these techniques are compared in accomplishing the task of traffic light detection in real time. The two models that are designed and implemented are Convolution neural network (CNN) and Deep Convolution Inverse Graphics Network (DCIGN). The results clearly show that DCIGN out performance CNN by 8%.*

***Index Terms:** Convolution neural network, Deep Convolution Inverse Graphics Network, stochastic gradient descent algorithm, Autonomous vehicles, traffic light.*

I. INTRODUCTION

Autonomous vehicles are predicted to be one of the most useful inventions in the automobile and the technological industry, if the very purpose of this invention is rightly achieved. In order to ensure the success of autonomous vehicles, one of the most important and crucial tasks that an autonomous vehicle has to perform accurately is traffic light detection in real-time. Traffic light detection and classification is crucial for automated driving vehicles, more so in urban environments. Hence, there is a necessity to develop and implement a model or a system that performs the task of traffic light detection and classification, in real time, in an accurate and precise manner, in order to ensure that none of the traffic rules are violated and pedestrians and other drivers are not injured or harmed. This calls for the development of a model that can detect the presence of a traffic light, classify the traffic light accurately and guide the autonomous vehicle to take the correct and necessary action with respect to the detected traffic light. This can be achieved using image classification techniques along with machine learning.

II. RELATED WORK

Much of research is happening in the field of autonomous vehicle manufacture like detection of traffic lights and road signs by these vehicles. These vary on the techniques used, the environments consideration, and the vehicles considered for implementation. Some research groups have done extensive research in this field and few important finding are presented in this section.

Ozcelik et al have proposed A Vision Based Traffic Light Detection and Recognition Approach for Intelligent Vehicles [7]. Images are taken using a camera and processing is performed step wise to detect the traffic. The color of the traffic light is identified easily through the Support Vector Machines (SVM) classification model, that is a machine learning algorithm prepared beforehand, after the location of the traffic lights is determined in the image.**Muller et al** have proposed Detecting Traffic Lights by Single Shot Detection techniques [6].

Behrendt et al have proposed A Deep Learning Approach to Traffic Lights by Detection, Tracking, and Classification [2]. This proposed methodology presents a complete system consisting of a traffic light detector, tracker, and classifier based on deep learning, stereo vision, and vehicle odometry which perceives traffic lights in real-time.

Li et al have proposed a Traffic Light Recognition Technique for Complex Scene with Fusion Detections [5]. **Saini et al** have proposed A Vision-Based Traffic Light Detection and State Recognition technique for Autonomous Vehicles [9]. It presents a vision-based technique for the traffic light structure detection using CNN that is based on state recognition method, which is considered to be robust under different illumination and weather conditions. **Shi et al** have proposed Real-Time Traffic Light Detection with Adaptive Background Suppression Filter [10].

Hamdi et al have proposed to system for Road signs Classification by ANN for Real-Time Implementation [3]. This system proposes a real-time algorithm for shape classification of traffic signs and their recognition to provide a driver alert system. **Abedin et el** have proposed a system for Traffic Sign Recognition Using Hybrid Features Descriptor and Artificial Neural Network Classifier [1].

Dharani et al have proposed a method for Traffic Light and Sign Detection for Autonomous Land Vehicle Using Raspberry Pi [8].

Manuscript published on 30 August 2019.
*Correspondence Author(s)
Dr. Mahesh .G, Computer Science and Engineering, BMS Institute of Technology and Management, Bangalore, India..
Dr. Satish Kumar .T, Computer Science and Engineering, BMS Institute of Technology and Management, Bangalore, India..

Huu et al Real-Time Traffic Light Detection Using Color Density [11]. In this paper the authors have proposed a method to detect traffic lights that uses color density identification. **Yudin et al** have proposed the Usage of Fully Convolutional Network with Clustering for Traffic Light Detection [12]. This approach is compared with one of the most effective object detectors, that is called Single Shot Multibox detector.

Jung et el have proposed a Real-time Traffic Sign Recognition System with Deep Convolution Neural Network [4]. In this system, 6 types of traffic sign images are trained by 5 convolution neural network architecture. **Tjandra et al.** explore the use of Stochastic Gradient Variational Bayes (SGVB) method for training the deep neural network and suggested to investigate the use of SGVB in different ANN architectures which includes Convolutional Neural Network (CNN)[14].

The various research works mentioned here present many methodologies for the detection of traffic lights. But these methodologies are also hampered due to the presence of various drawbacks. The systems presented in [1], [4], [7], [8], [10], [11] and [12] fail in considering a vast variety of both Training and Testing datasets, that can be considered to scale the respective systems. [3] and [6] fail on being tested in harsh weather conditions. [2] requires high quality hardware components, to achieve the expected levels of accuracy.

III. PROPOSED SYSTEM

The proposed system aims at performing real time traffic light detection in autonomous vehicles. This can be achieved by implementing machine learning and image classification techniques. Artificial Neural Networks, Convolution Neural Networks and in specific, are one of the most accurate methods in order to achieve the desired result. Two algorithms CNN and DCIGN are implemented and also compared.

The basic ideology behind the proposed system/model is to develop a traffic detection model that is trained using a publicly available dataset. The model is trained using the various images present in the dataset, for which CNN and DCIGN are used. The trained model is then tested using the test dataset in order to test the working and prediction accuracy of the model. The dataset used in order to build and train the model is the Bosch Traffic Lights Dataset. This dataset is currently the largest publicly available traffic lights dataset. It consists of more than thirteen thousand images that have been captured in various traffic scenarios, thus helpful in order to build a wholesome and accurate traffic detection model. It is observed that SGDA and its other variants are time tested algorithms that have proven as some of the best learning algorithms for large datasets. Here in the experiments done in this research two aspects has been taken care of (i) Perform small-scale experiments are performed with subsets of the training data, and (ii) stony attention to the correctness of the gradient computation.

The CNN algorithm, has one or many layers of convolution units. A convolution unit receives its input from one or more units from the previous layer which is connected in proximity. Thus, the input units that form a small neighborhood share their weights. They are great for capturing local and also reduce the complexity of the model. The phases of a CNN algorithm are (also shown the respective modules implemented in the research project) :-

Model Construction

Step 1: Begin with the initial model.
Ex : model = Sequential()

Step 2: Then add the required number of layers with their types (input, hidden, output).
Ex : model.add(type_of_layer())

Step 3: The model is compiled, after adding sufficient number of layers.

Step 4: During model compilation it is important to write a loss function and an optimizer algorithm.
Ex : model.compile(loss= 'name of loss function', optimizer= 'name of optimizer algorithm')
The function(loss) shows the accuracy of each prediction made by the model.

Model Training

Step 5: The model is then fed the training data with the expected output for training the model.
Ex : model.fit(training_data, expected_output)

Model Testing

Step 6: In this phase of the algorithm a second set of data, that is called the validation set is loaded. Ex :
model.save("name_of_file.h5").

Step 7: If the accuracy of the built model does not meet a predefined threshold, then the model architecture must be changed (Step 2) or more training must be done (Step 5).

Model Evaluation

Step 8: Finally, the saved model can be used in the real world. The name of this phase is model evaluation. This means that the model can be used to evaluate new data.
Ex: model.predict(testing_data)

The second method of traffic light detection that is used in DCIGN model and it is composed of multiple layers of convolution operators and is trained using the SGDA. The DCIGN algorithm has the following phases": Input layer, one Kernel Layer, two convolutional Layer, two Probabilistic hidden layers, two convolutional Layer, one Kernel layer and Output layer. Even before applying DCIGN Stochastic Gradient Descent algorithm is used for training.

Using Stochastic Gradient Descent Algorithm (SGDA)

Since the Gradient Descent algorithm is infeasible, when the training data size is huge, stochastic gradient descent algorithm (SGDA) is used. The objective function for SGDA is the sum of a finite number of functions:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad -(1)$$

In expression (1) $f_i(x)$ represent the loss function that is based on the training data instance and indexed by i. In gradient descent the per-iteration computational cost is scaled linearly with the training data set size. Therefore one can expect a very high per-iteration computational cost for huge n. When generalized, the equation can be rewritten as equation (2), for each iteration a mini-batch β that represents the indices for training data instances can be sampled at uniform intervals.

$$\nabla f_\beta(x) = \frac{1}{|\beta|}\sum_{i\epsilon\beta}^{n} \nabla f_i(x) \quad -(2)$$

x is updated as

$$x = x - \eta \nabla f_{\beta}(x) \quad -(3)$$

In equations (2) and (3) β represents the cardinality of the mini-batch. In equation (3), the positive scalar η is the step size. Also, the mini-batch stochastic gradient $\nabla f_{\beta}(x)$ is an estimator that is unbiased for the gradient $\nabla f(x)$:

$$E_{\beta} \nabla f_{\beta}(x) = \nabla f(x) \quad -(4)$$

The computational cost per-iteration is O(|β|).

IV. IMPLEMENTATION

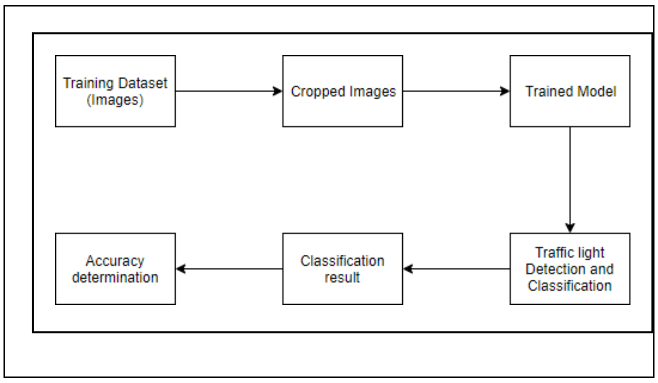

Figure-1: General Flow of the Developed System

Figure-1 represents the overall process of the project. It depicts the general flow of the developed model. The following steps constitute the basic flow of the developed system:

i. Images depicting various scenarios in a real time traffic environment are captured using multiple cameras. These images constitute the dataset which is used to train, validate and test the model.

ii. The images in the dataset cover a decent variety of road scenes. These images consist of many objects apart from traffic signals such as trees, vehicles, road signs, buildings, decorative lights, etc. The images are cropped to meet the bounding box dimensions mentioned in the configuration file and labeled accordingly. Now the cropped images would contain only the traffic signal.

iii. The machine is trained using the cropped images to apply various filters and algorithms to determine the color of the traffic signal accordingly. A trained model is generated after the training process.

iv. The trained model is validated using a validation data set, which is a subset of the original dataset that is gathered. Validation is done in order to improve the accuracy of the model by adjusting its parameters.

v. The test dataset (images) is then given as input to the trained model in order to verify the working of the trained model.

vi. The model then performs the classification of the traffic signals and it outputs the color and direction displayed by the traffic signal.

vii. The extent to which the trained model performs the classification correctly constitutes to the accuracy of the classification. This accuracy of the model is output along with the classification result for a particular image.

In this research work, a regression-based algorithm knows as the YOLO (You Only Look Once) Algorithm has been used to perform object detection.

This algorithm is used for real-time object detection. In this algorithm, classes and bounded boxes are predicted for the whole image in one run of the algorithm. The YAML file consists of the cropping dimensions, the object labels and the RGB values, which are used to perform the cropping function.

The cropping function is then performed in order to obtain only an image of the traffic signal that has to be detected and classified by the automated vehicle. The cropped images are then used to train the proposed model, which is followed by the prediction function. The model makes use of the test dataset in order to predict the classification of traffic lights. The prediction results and the actual classification together can be used to calculate the classification accuracy of the proposed system.

YOLO ALGORITHM (You Only Look Once)

YOLO is an extremely fast real time multi object detection algorithm. YOLO stands for “You Only Look Once”. It is a regression-based object detection algorithm that is used to detect objects in real time. The algorithm applies a neural network to an entire image. The network divides the image into an S x S grid and comes up with bounding boxes, which are boxes drawn around images and predicted probabilities for each of these regions.

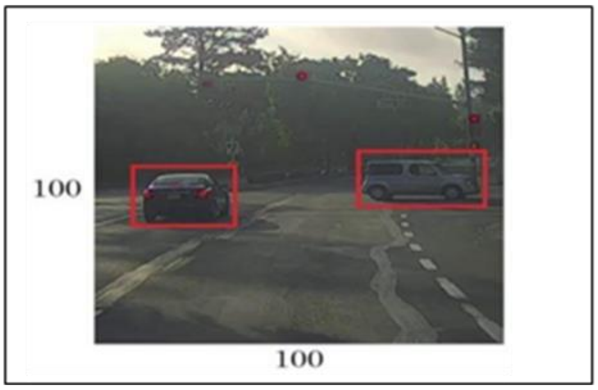

Figure 2: Example of an Input Given to the YOLO Algorithm

The method used to come up with these probabilities is logistic regression. The bounding boxes are weighted by the associated probabilities. For class prediction, independent logistic classifiers are used.

STEP 1: YOLO first takes an input image, as shown in figure-2:

STEP 2: The framework then divides the input image into grids (say a 3 x 3 grid) as shown in figure-3:

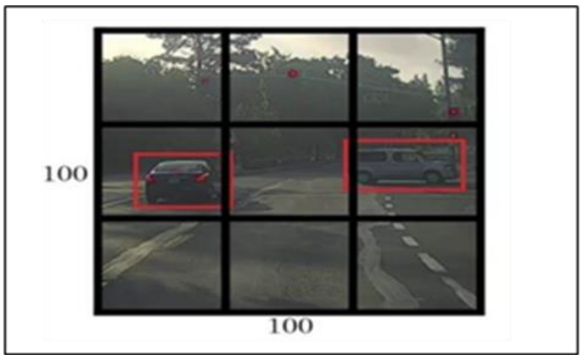

Figure 3: Grid Formation by the YOLO Framework

STEP 3: In this step the Image classification is done and then followed by localization that are applied on each grid. YOLO algorithm then helps in predicting the bounding boxes and their corresponding class probabilities for objects

- The YOLO algorithm is a regression-based object detection algorithm.
- This algorithm is used to detect objects in real time.
- Classes and bounded boxes are predicted for the entire image in one run of the algorithm.
- YOLO algorithm takes an entire image and splits it into an S x S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.

After going through the entire process of researching, implementing and testing the developed model, it can be concluded that Artificial Neural Network Technique proves to be a very efficient method for the detection and classification of traffic lights in real time.

One of the most important findings was during the training process. It was found that model training failed if the training process was interrupted in between. Due to the nature of the training process and the characteristics of the algorithm being used, it is necessary that the training process remains uninterrupted in order to make sure that model training completes successfully.

Another important finding was encountered while building the model. As mentioned in Table-1, the model was tested with different epoch values in order to monitor its performance at various stages. One epoch is when an entire dataset is passed forward and backward through the neural network only once. A training process may consist of more than one epoch. As the number of epochs increases, a greater number of times the weights are changed in the neural network. This gradually changes from underfitting, to optimal to overfitting of the model.

Table-1: Testing with different epoch values

Epoch Value	Validation Accuracy	Validation Loss
10	0.9325	.1998
20	0.9375	.1878
30	0.9474	.1792
40	0	0.1774
50	0	0.1745

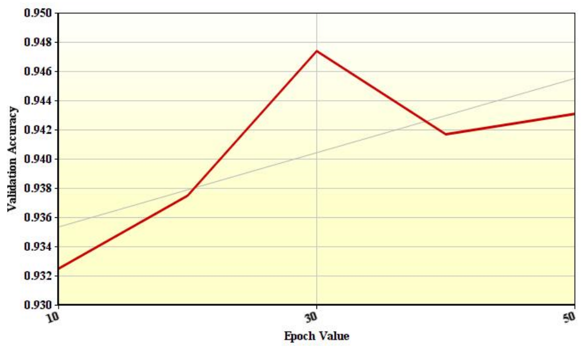

Figure 4: Graph depicting Validation Accuracy vs Epoch Value

Figure-4 is a graph that is plotted against the validation accuracy of the model and the epoch values used for training. It can be seen that, after an epoch value of 30, there is a change in the direction of the curve. This denotes overfitting of the model. For the traffic light detection and classification model developed in this project, it was found that for a constant batch size of 250, as the epoch value was increased from 10 to 30, the model went from underfitting to an optimal stage. On further increasing the epoch value beyond 30, it was noticed that overfitting of the model was taking place. In Figure 6, the peak of the curve at an epoch value of 30 denotes that the model performs optimally for an epoch value of 30 and batch size of 250. Any epoch value less than 30 would lead to underfitting of the model, and similarly any epoch value greater than 30 would lead to overfitting of the model. Another important result was found during the prediction process. The developed model was able to detect multiple traffic lights in a single image, and perform the prediction process on the traffic light that is nearest to the vehicle under consideration. This is important as the real-world scenario works in the exact same way. Also, in case there is no traffic light in a particular situation, the model is able to guide the vehicle in the right manner.

Table-2: Comparison between CNN and DCIGN for detecting the signals.

METHOD	ACCURACY
CNN	90.4 ± 0.61%
DCIGN	98.2±1.32%

Table-2 compares the accuracy of CNN and the DCIGN algorithm, it can be observed that CNN achieves 90.4% and DCIGN achieving 98.2% accuracy, which is significantly higher than that for the CNN method.

V. CONCLUSION AND FUTURE ENHANCEMENT

One of the most important findings was during the training process. It was found that model training failed if the training process was interrupted in between. Due to the nature of the training process and the characteristics of the algorithm being used, it is necessary that the training process remains uninterrupted in order to make sure that model training completes successfully. Clearly, the CNN when compared to DCIGN algorithm, DCIGN out performance CNN by 8%, it is evident from the table-2. In future the hidden layer in the neural network may be increased to reduce the error rate and improve the accuracy of the model.

ACKNOWLEDGMENT

We would like to whole heartedly thank our graduate students P. Vignesh, Radhika B Raman and Ullas V B for providing the literature survey and implementing a part of this work as their project.

REFERENCES

1. Abedin, Md Zainal, Prashengit Dhar, and Kaushik Deb. "Traffic sign recognition using hybrid features descriptor and artificial neural network classifier." 2016 19th International Conference on Computer and Information Technology (ICCIT). IEEE, 2016.
2. Behrendt, Karsten, Libor Novak, and Rami Botros. "A deep learning approach to traffic lights: Detection, tracking, and classification." 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.
3. Hamdi, Sabrine, et al. "Road signs classification by ANN for real-time implementation." 2017 International Conference on Control, Automation and Diagnosis (ICCAD). IEEE, 2017.
4. Jung, Seokwoo, et al. "Real-time Traffic Sign Recognition system with deep convolutional neural network." 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, 2016.
5. Li, Xi, et al. "Traffic light recognition for complex scene with fusion detections." IEEE Transactions on Intelligent Transportation Systems 19.1 (2018): 199-208.
6. Müller, Julian, and Klaus Dietmayer. "Detecting traffic lights by single shot detection." 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018.
7. Ozcelik, Ziya, et al. "A vision-based traffic light detection and recognition approach for intelligent vehicles." 2017 International Conference on Computer Science and Engineering (UBMK). IEEE, 2017.
8. Priyanka, D., et al. "Traffic light and sign detection for autonomous land vehicle using Raspberry Pi." 2017 International Conference on Inventive Computing and Informatics (ICICI). IEEE, 2017.
9. Saini, Sanjay, et al. "An efficient vision-based traffic light detection and state recognition for autonomous vehicles." 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017.
10. Shi, Zhenwei, Zhengxia Zou, and Changshui Zhang. "Real-time traffic light detection with adaptive background suppression filter." IEEE Transactions on Intelligent Transportation Systems 17.3 (2016): 690-700.
11. Tran, Tai Huu-Phuong, et al. "Real-time traffic light detection using color density." 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia). IEEE, 2016.
12. Yudin, Dmitry, and Dmitry Slavioglo. "Usage of fully convolutional network with clustering for traffic light detection." 2018 7th Mediterranean Conference on Embedded Computing (MECO). IEEE, 2018.
13. Kulkarni, Tejas D., William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. "Deep convolutional inverse graphics network." In Advances in neural information processing systems, pp. 2539-2547. 2015.
14. Tjandra, Andros, et al. "Stochastic gradient variational bayes for deep learning-based ASR." 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015.

AUTHORS PROFILE

Dr. Mahesh G, holds B.E., M.Tech and Ph.D in Computer Science & Engineering from VTU. He was a 2nd Rank holder in M.Tech. He is currently working with Department of CSE, BMSIT. Prior to this he was associated with Acharya Institute of Technology, Bangalore. He has 15 years of professional experience, which spans from academics, research and consultancy.
He has published around 20 papers in reputed International Journals / Conferences. His current research interests include stochastic and Petrinets modeling of wireless networks. He is a member of Society of Digital Information & Wireless Communication, International Association of Engineers and Indian Society for Technical Education. He was also the BOS member of Dr.AIT, Bangalore. Cognizant Technology Solutions has honored him with the Best Faculty Award in 2017.

Dr. Satish Kumar .T, holds B.E from Bangalore University, M.Tech and Ph.D in Computer Science & Engineering from ANNA University.
He is currently working with Department of CSE, BMSIT. Prior to this he was associated with RNS Institute of Technology, Bangalore. He has 19 years of professional experience, which spans from Industry, academics, research and consultancy. He has published around 28 papers in reputed International Journals / Conferences.
His research primarily focuses on Code Optimization on High performance Computing (HPC) systems using heuristics methods. Specifically, using Multi-core clusters with high degree of computations. Over the years Code optimization, while highly relevant to HPC is slowly picking up grounds in mobile computing and IOT. His current research extends to development of Compiler Optimization Algorithms, design of RTOS and Embedded system design, and its synchronization.
He is a member of Indian Society for Technical Education and IEEE. He was also the BOE member of VTU, Belagavi and journal reviewer for several journals.