Intelligent Traffic Signal Control in Smart Cities using Deep Learning Model

Mansi Subhedar[a*], Uttaran Roychowdhury[a]

[a]*Department of Electronics and Telecommunication Engineering, Pillai HOC College of Engineering and Technology, Rasayani Tal. Khalapur, Dist Raigad, Maharashtra, India (mansissubhedar@gmail.com)*

Abstract

In many developing nations, massive infrastructure projects like flyovers or underpasses are a big investment and most take durations of up to multiple number years to complete. This work provides an AI-based traffic control system for intersections at such locations. The proposed system is viable for modern smart cities where traffic intersections exist with the need for adaptive switching as a future-proof solution when the cities get in more vehicle load. This work can be built on an already existing surveillance system of close circuit cameras used to monitor traffic and keep an eye on traffic rules violations. In recent years, deep learning solutions have shown new avenues of automation and it is still in the growing phase of its life cycle. These neural network systems are flexible and can be molded to do a variety of tasks. The proposed work contributes to the design of three neural networks that work in synergy to achieve the task of fluid automated traffic signal switching. Specifically, a Convolutional Neural Network (CNN) in Inception V2, a Long Short Term Memory Neural Network (LSTM) for forecasting, and a Deep Neural Network (DNN) for the classification of switching decisions are developed. In this work, adaptive traffic signal switching is treated as a classification problem and is handled in a sequential manner utilizing the specific strengths of different neural networks that are easy to maintain in the future. The cost over time for implementing such systems will get lower as technology progresses and gets easily accessible. This perspective will minimize traffic congestion and aid in reducing traffic rule violations. Congestion reduction in turn also translates into less wait/travel time and less pollution. As fuel costs increases, we need to get more efficient in every aspect of transportation and can be achieved with the proposed work.

Keywords: Smart Cities, Deep learning, DNN, Inception, LSTM, Traffic light Control

1. Introduction

Traffic management is a very important issue all over the world and many developed and developing countries are looking for a solution to the ever-rising number of road vehicles. Many countries concentrated on going towards public transport and sacrificing the luxury and convenience of the private vehicle sector. The other solution was to upgrade the infrastructure by building flyovers and expanding the width of the roads. All these solutions have come a long way to improve the experience of private travel on road, but the traffic signals have been the same for a long time and only brought to the digital era by the integration of microprocessors. But signals in concept were not smart or adaptive in nature. A need for adaptive traffic signal control arose from the fact that there are variations in the performance of the previously implemented systems. These variations were mostly in a day or soon after the systems were up and operating. The deterioration happened as soon as the conditions deviated significantly from the base conditions [1]. The proposed system aims to bring this to the information era where data can be used to optimize everything and anything. This work aims to modernize the system by making it adaptive to the changing traffic conditions depending on the time of the day and as well as the live input. The process goes through 3 steps of neural networks, the first section of the model is the CNN network; an Inception V2 network [2] used for the detection and counting of vehicles in the visible region of the road. The second section is the DNN model used for the classification of the type of switching that must be done at the intersection. The third section is the forecasting done by the acquired data from the very first step (CNN network) and predicts the daily traffic by the minute. All these different deep learning systems work in synergy and help us achieve the desired goal. This proposed system does not provide traffic signal synchronization rather it is a method for the switch in a single direction of the road. This whole system can be implemented for the N-number of roads connecting to the intersection with an appropriately designed synchronization tool.

The paper is organized as follows: Section 2 deals with related works. Section 3 presents the proposed work. Simulation results are presented in Section 4. Section 5 concludes the work.

* Dr Mansi Subhedar
E-mail address: mansissubhedar@gmail.com

2. Related Works

Traffic forecasting and prediction have a long research history, and many have suggested novel implementations using machine learning and deep learning techniques. Many systems have been suggested for smart traffic management in many varying platforms such as traffic lights, smart vehicles, etc. NCHRP says that there have been many adaptive traffic solutions developed by transportation research institutes and agencies all over the world but less than half of them have been deployed and tested in real-world scenarios [3]. Wang et al. discuss multiple levels of transport intelligence methods and ranks them into 5 levels starting from one that is Model-based, MAXBAND, etc. methods, and the final level is the Model-free learning adaptive control systems. One of those systems that improved upon the timer-based systems is the Split Cycle Offset Optimization Technique also known as SCOOT [4]. It is a level 2 intelligent decision-making system. SCOOT was developed by Transport Research Laboratory for the Department of Transport in 1980; it uses data from the motion sensor to adjust traffic signal delays. SCOOT is most widely used in the United Kingdom today. Another, level 2 intelligence system is Sydney Coordinated Adaptive Traffic System (SCATS) [5]. It is a system that manages the traffic signal in a dynamic manner using road traffic sensors and pedestrian switches to optimize cycle times, phase splits and offsets. It also incorporates a priority traffic facility within it and considers buses and trams as well. The system is owned by Australia and is in operation in Sydney. The SCAT system is installed in more than 42,000 intersections across 40 countries. Another work uses USB based camera systems as image input, and certain image recognition and object counting techniques are employed to determine the amount of traffic and calculate traffic density [6]. They have also used an RFID reader and tag system to transmit the information required to begin capturing the images. This system mainly depends on the machine learning technique of Support Vector Machine (SVM) for object counting which works better than the existing timer-based systems. Arif et al suggested a motor-mounted camera system for a similar purpose as in [6] but they rely on image processing techniques to determine the region of interest and assess the amount of time to assign to the traffic light timers. The proposed system involves three different techniques together and hence some of the technique's related study on an individual basis is explored. Another prominent approach that many research studies on this topic was the implementation of deep reinforcement learning [8].

The field of deep learning and neural networks shows a lot of promise when it comes to image recognition-based systems. Several image recognition approaches can be found; one of them being the localization with a selective search algorithm that feeds into a DCNN for recognition of the proposed region and uses a linear SVM for classification [9]. The performance of their system is verified using multiple criteria based on road conditions such as free-flowing traffic, congested traffic, and stop-and-go traffic in different weather conditions along with camera resolution changes. It was found that the system performance is better at the daytime with free-flowing traffic conditions. This was especially important for consideration with time of day and the resolution of the camera inputs. Over the time, the camera resolution will improve but the impact of weather conditions on the input video stream is something to be taken into consideration.

The other part of the proposed system is the LSTM forecasting method; there have been various attempts at predicting traffic patterns. A detailed study of the road traffic data using different parameters and techniques is conducted in [9] which determines that the two fixed points in the data are jam and free-flow, the abrupt changes prove the non-linear nature of the traffic. The data used for the study is from North-Rhine Westphalian motorways. ARMA and ARIMA are two methods widely used for forecasting [10]. But these are linear autoregressive models and cannot predict the stochastic and non-linear nature of the traffic flow [11]. The LSTM & GRU (Gated recurrent unit) fair so much better than those due to the ability to memorize long-term dependencies. If the LSTM models and GRU models are compared, the GRU units are much simpler to implement and more efficient for computation. This is due to the fact that the method used for mitigating vanishing gradients is done by only two gates being a reset gate and an update gate for each cell, meanwhile LSTM has three gates being input gate, output gate and the forget gate. A proper comparison study has been conducted in [12]. Traffic time-series studies are being conducted for a long time. A technique presented for prediction using ARIMA is discussed in [13]. A traffic control optimization technique is proposed by Li et al. using multi-agent deep reinforcement learning by improving the coordination between traffic signals [14]. Egea et al proposed an urban traffic control mechanism using reinforcement learning with three different architectures and compared it with existing commercial systems [15]. Li et ad proposed various models for adaptive traffic control that rely on data collected only from connected vehicles [16]. Several similar studies are discussed in [17]-[18].

3. Materials & Methodology

The details of the setup used to train and evaluate the proposed model are provided here. The deep neural network and the Long Short-Term Memory Neural Network (LSTM) were built on Keras API with TensorFlow backend. The whole experiment is done on a standard laptop PC running windows 10 OS with Nvidia GeForce GTX 1650 card with 4 GB video memory and an Intel core i5-9300H CPU, CUDA v11 and TensorFlow 2.3. The programming environment used is anaconda with python v3.6. The Object detection and counting API is running on TensorFlow 1.X settings with TensorFlow 2.3. Specific changes disabled.

a. Dataset

The time series prediction model data on traffic density and particularly in this kind of implementation was sparse, even the available data was not suitable for the needs of this work. Hence, custom synthetic data was developed for training and testing purpose and would serve until the real-world data is created by the system to run inference over it again by using real-world data.

The custom dataset is used for both the decision making deep neural network and sequence model in the traffic forecasting neural network. The data ingestion pipeline is handled by pandas and made into dataframe from .csv file. The three compare columns in the dataset are made by random number generator that gives numbers at random a at range of -1, 0 and 1. The values here are associated to a certain condition as follows; first '-1' is when the compared roads count is less than the main road on which the operation is taking place on, '0' is for condition when both are equal and finally '1' is for when the compared road has more vehicles than the current one. Another column is a comparison between the car counts at the video against the predicted car count at that instant from the forecasting LSTM model. The last column is the decision. It is provided by considering the four comparison columns. For real world implementation, the csv files are not ideal so use of SQL databases is recommended as the data would be too massive to contain in standard csv file formats that has limited scalability. The implementation of SQL based database would also make the reading of the input data much quicker as it will not be fragmented into separate files as the amount of data increases. For this project the dataset contained around 80,000 data points.

For experimental setup, here, csv file is considered.

Table 1: First 5 entries in the input data

Time-stamp	Car_count	compare_left	compare_opp	compare_right	Train_Prediction	count_and_pred	min	hour	day_of_week	day_of_month	month	Decision
23-09-2020 00:00	2	0	0	-1	0	0	0	0	2	23	9	0
23-09-2020 00:01	2	0	0	-1	2.21491695	1	1	0	2	23	9	0
23-09-2020 00:02	2	0	-1	-1	2.21491695	1	2	0	2	23	9	1
23-09-2020 00:03	3	-1	-1	-1	2.21491695	0	3	0	2	23	9	5
23-09-2020 00:04	3	0	-1	-1	3.87389994	1	4	0	2	23	9	1

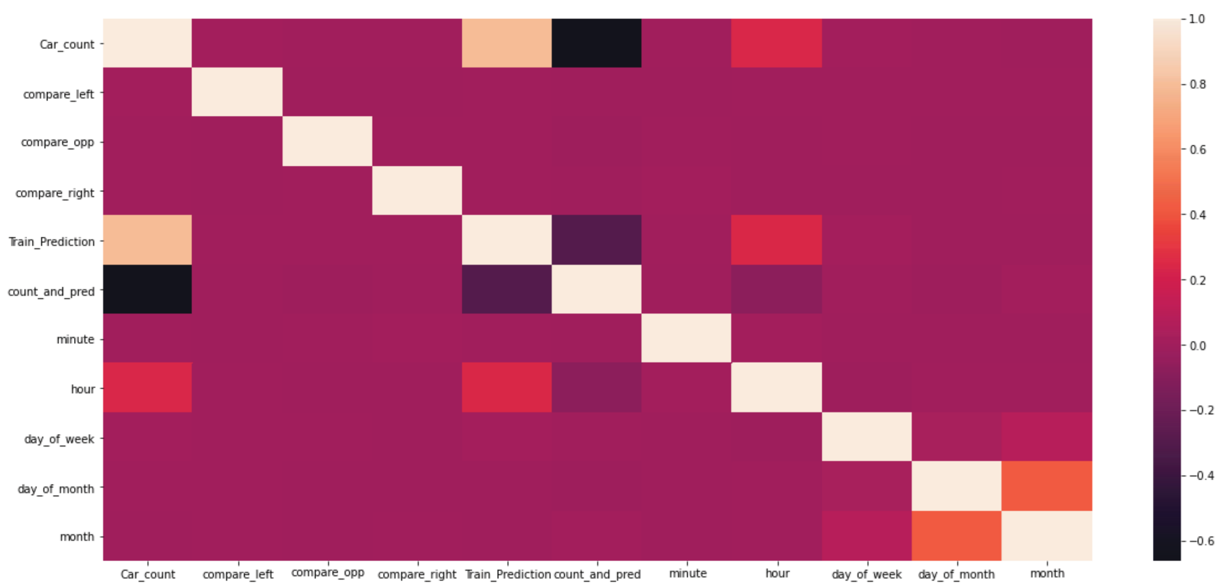

Fig. 1. Correlation Heat Map

1.1 Preprocessing and augmentation

The entries in the dataset are preprocessed, cleaned for null values, and checked for errors by using pandas and NumPy library in python. The video input for this experiment is considered as a video downloaded from YouTube as a sample called "4K camera example for Traffic Monitoring (Road)". It is used in an .mp4 format. But live input footage can also be employed and obtained from an IP based close circuit camera by using OpenCV library if required during deployment [19].

1.2 Models and Overall Working

This is a hybrid three-model system where the input is a video stream that provides the number of vehicles in the frame. Those numbers are saved in a .csv file along with the current time stamp. Some of the sample values are shown in Table 1. The dataset for all values is maintained in a similar way. A correlation heat map is depicted in Figure 1. This CSV file is used as input for the sequence model or Long Short-Term Memory Neural Network. (LSTM) model is used for forecasting the monthly traffic for that specific intersection. The predicted values are again written into the same .csv file. LSTM model output is a regression-based output so the accuracy is not the metric here. Rather, the mean squared error that measures how close the prediction is to the actual value is employed. Finally, the Deep Neural Network is used to classify the desired decision. The decision is the right-most column in the dataset and contains the labels for the classification. In this work, the decision is considered a classification problem because the decisions are finite in nature and can be selected from these predefined classes. There are six classes labeled as right, right & forward, only forward, left and forward, left and no change. The output given by the deep neural network is a number that corresponds to one of those six decisions. The overall working model is shown in Figure 2.

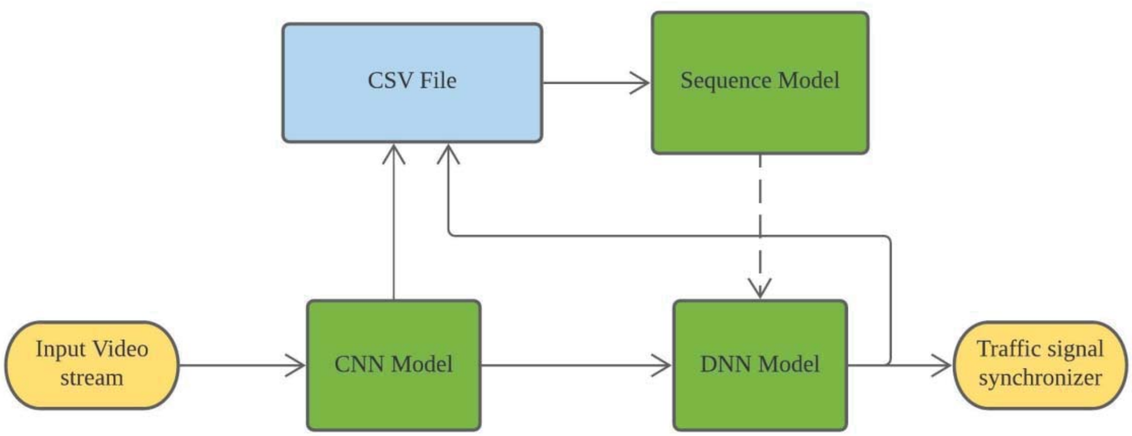

Fig. 2. The proposed overall working model

1.3Object counting

Object counting is done by the API with pre-trained convolutional neural network called Inception-V2 trained on MS coco [20] but ImageNet [21] can also be used for a custom dataset. Inception V2 is an object recognition model and is used to produce bound boxes around the object of interest. This pre-trained neural network can classify 91 classes ranging from cars and buses to cats and dogs, etc. It is used to recognize the number of vehicles in the current frame. Vehicles include cars, buses, trucks, and motorcycles; the sum of a number of all these objects in the frame is used to determine the number of vehicles. The API does the counting by giving the number of bound boxes on the frame as shown in Figure 3. The object recognition rate and localization are not very high the inception-V2 model here.

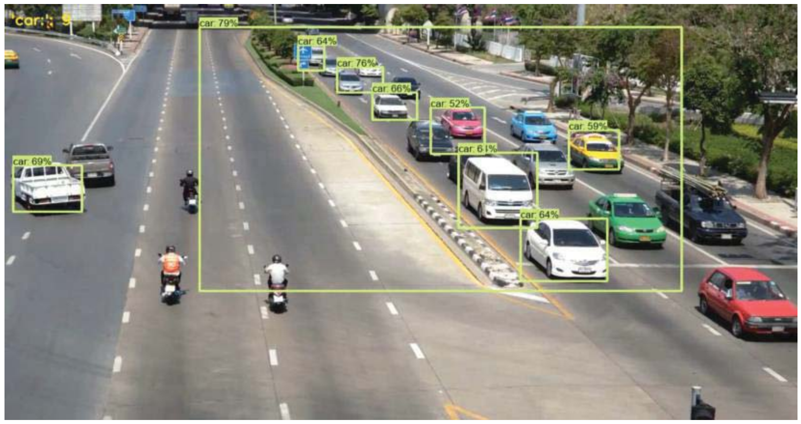

Fig. 3. **Object counting for one frame**

The count accuracy of the API depends on the model's accuracy. For experimentation purpose, the accuracy is traded for faster runtimes and lower memory consumption. But more deeper and advanced models like RetinaNet-50 or CenterNet [22] would be more suitable for deployment purposes.

1.4 Long Short-Term Memory Neural Network (LSTM)

The Long Short-Term Memory neural network or LSTM is
a sequence model and determines the repeating nature of the data. Traffic like many things such as weather and earthquakes are a predictable pattern-based phenomenon. This pattern can be predicted and forecasted beforehand to be prepared for
what's to come in near future. Traffic at certain places follows this repeated pattern monthly or annually depending on the place.

```
_________________________________________________
Layer (type)            Output Shape       Param #
=================================================
lstm (LSTM)             (None, 1, 32)      4352
_________________________________________________
lstm_1 (LSTM)           (None, 1, 64)      24832
_________________________________________________
lstm_2 (LSTM)           (None, 32)         12416
_________________________________________________
dense (Dense)           (None, 32)         1056
_________________________________________________
dropout (Dropout)       (None, 32)         0
_________________________________________________
dense_1 (Dense)         (None, 1)          33
=================================================
Total params: 42,689
Trainable params: 42,689
Non-trainable params: 0
_________________________________________________
```

Fig. 4. Proposed LSTM architecture

The proposed model as shown in Figure 4 and 5 consists of three layers of LSTM and a dropout layer with 50% tolerance and the last layer is a Dense layer with one neuron as output. LSTM is not a classifier and hence accuracy metric is not followed for measuring the performance. Mean squared error as the measure for performance is employed here as well as for calculating the loss function. The proposed model uses Adam optimizer for training. The generated forecasts are from 23rd September 2020 to 23rd November 2020 with every minute taken as an entry and one time step at a time. The converted raw input data representation is shown in Figure 4. Another test was conducted with a GRU model with 64-neuron first layer and a 32-neuron second layer and 2 dense layers with eight and a single neuron respectively with a recurrent dropout at the second layer. This resulted in similar results but was unstable during subsequent reruns of the model. Hence, LSTM model was chosen. But GRU model execution times were faster compared to the LSTM model. Figure 6 shows input data entries as sample. Several trials were taken to set the values and accordingly fine tuning was done while choosing the LSTM values. The predictive model was used to generate the expected traffic load which is compared with the current traffic load which aided in better decision making during switching the lights.

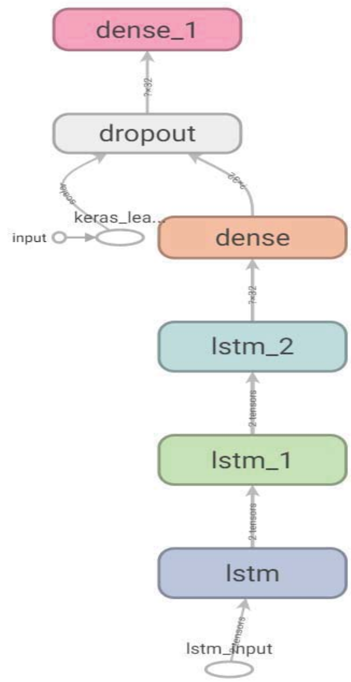

Fig. 5. Proposed LSTM model architecture

1.4 Deep Neural Network (DNN)

The deep neural network is the final part of in the predictive models. This model takes input from the object recognition counter and the long short-term memory neural network models to classify the type of decision that suits the current conditions. The proposed model architecture is in Figure 7 and Figure 8.

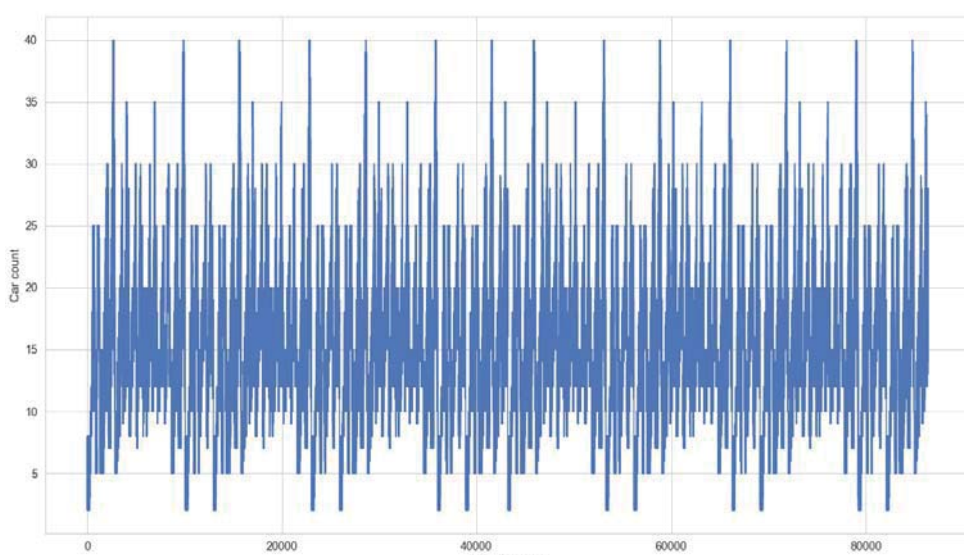

Fig. 6. **Input data across all the 80,000+ entries**

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 256)               3072
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_4 (Dense)              (None, 128)               16512
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_6 (Dense)              (None, 64)                4160
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_7 (Dense)              (None, 6)                 390
=================================================================
Total params: 131,078
Trainable params: 131,078
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. Proposed Deep Neural Network

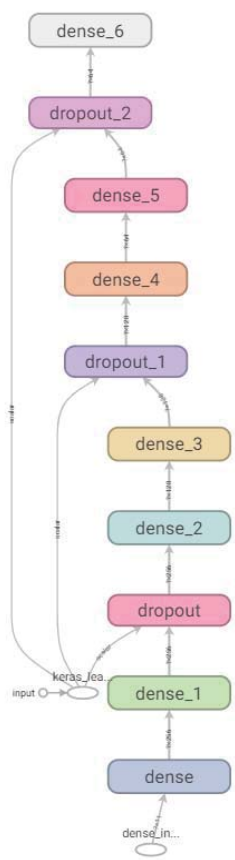

Fig. 8. Proposed DNN Model Architecture

The input is similar as shown in Table 1. This proposed model consists of 7 Dense Layers which are in pairs of 2 and one Dropout layer in-between each pair with 50% tolerance and the last dense layer has 6 neurons as this is a multiclass classification and each neuron out of the 6 is associated to the decision labels. This is done so that the system could be simplified to a decision, and each side can give independent outputs to the synchronizer. The DNN training is done in batch size of 32 in 30 epochs with a learning rate scheduler. The learning rate scheduler is used to reduce the learning rate by a certain amount after every 10 epochs, this helps the loss function find the proper desired local minima. The DNN is using labels created with respect to the data and what would be most optimal switching operation to be performed in such a condition to minimize the wait times of vehicles.

3.Results and Discussion

To evaluate the results of this experiment, the accuracy and loss function values of each of the models can be employed. The object recognition model is pretrained and as per Google AI blog has a Top-1 accuracy of 80.4% and a Top-5 accuracy of 95.3%. Considering the model is more than four years old the Top-1 accuracy is quite low for real world application where human safety should be utmost priority. But as discussed earlier, newer and deeper models can solve this problem. Also, a proper camera frame setting with only one side of road view is needed for good results as the vehicles that have already left the lights after it became green need not be counted. The Long Short-Term Memory Neural Network (LSTM) neural network is a regression type deep learning model; hence mean square error, mean absolute error and root mean square error metrics are used to determine the performance of the model.

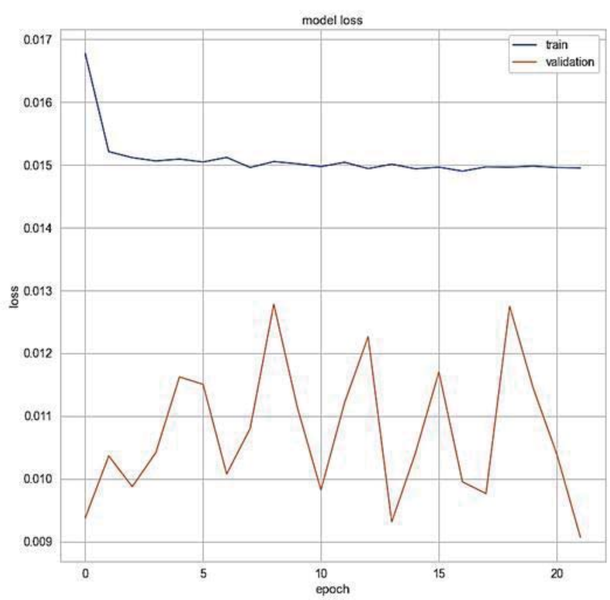

Fig. 9. Loss function for the LSTM

The mean square error is also the loss function in this case so one of the graphs is equal to the other and it is intended as such. The mean squared error stays around 0.0138 and 0.0137 for training and for validation it finishes at around 0.011 and 0.010 the Figure 9 and 10 show results for final model used for prediction. The mean absolute error is around 0.0915 and 0.0920 for training and 0.076 for validation. The LSTM model has a Root mean square error score of 4.37 in training and 4.32 in test set. The outcomes of the LSTM's forecasting were in acceptable range and not much farther from the original values, but the model had difficulty in predicting the higher end of the values or the maximum values in the traffic counted.

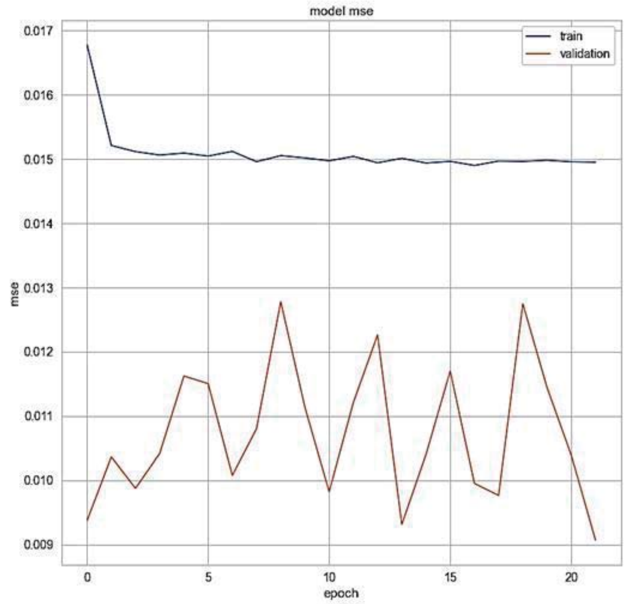

Fig. 10. Mean squared error for the LSTM

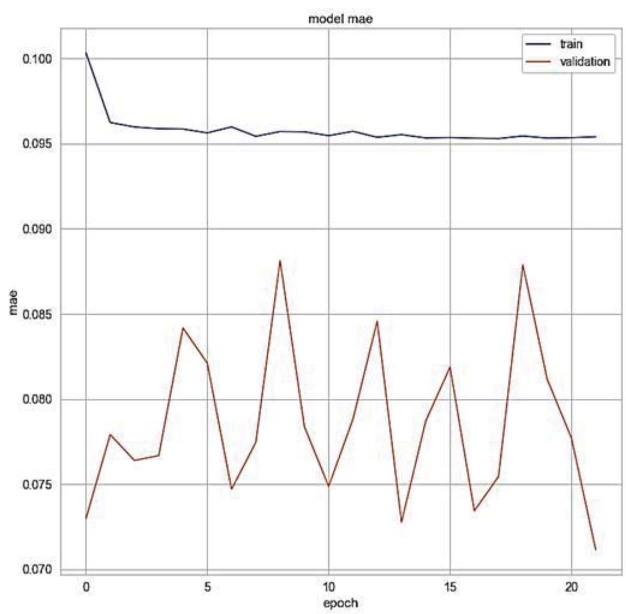

Fig. 11. Mean absolute errors for the LSTM

But it did show a rise in the predicted values which could be considered as the higher end during operations. The graph of predicted values against the original input values is displayed in Figure 12. The deep neural network's performance can be easily evaluated by measuring the loss and accuracy of the trained model. The Figure 12 shows the loss function graph and Figure 13 shows the accuracy metric where the highest training accuracy achieved is of 99.95% and highest validation accuracy of 100%. These metrics show that this system with multiple models can perform the switching task very reliably. The performance result could not be compared due to the fact that the dataset is a synthetic one and is assuming some specific kind of scenarios. The synthetic dataset was used because the data that we came across wasn't suitable or lacking specific traffic count criteria in multiple directions which was crucial for the comparison and decision-making process.

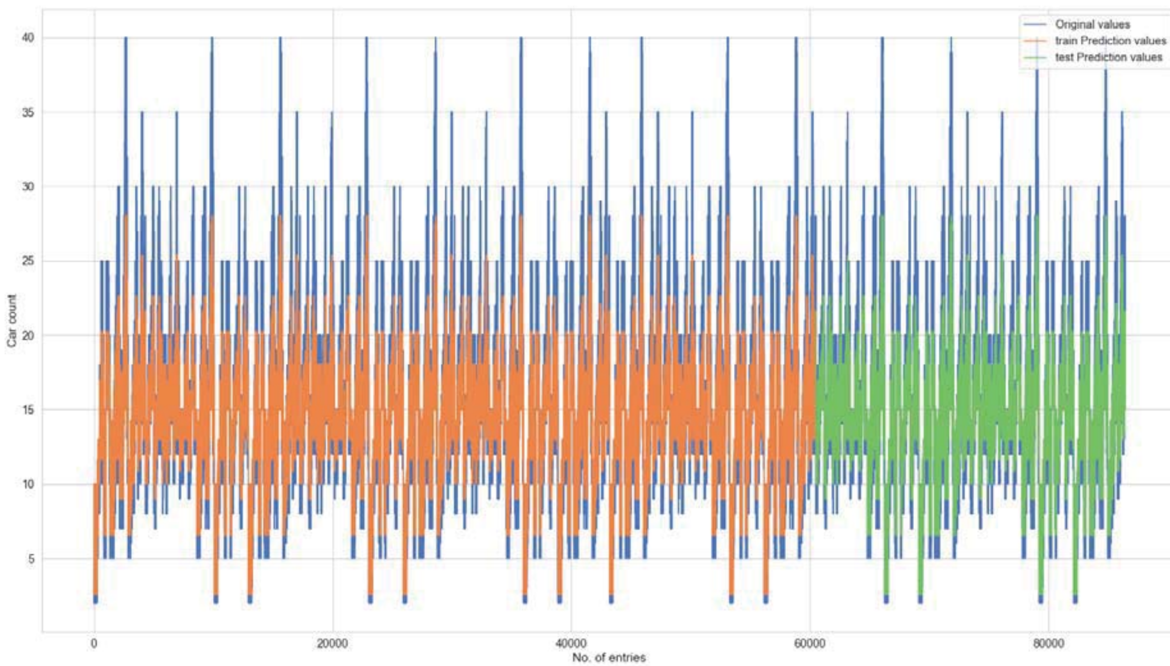

Fig. 12. Predicted against original input values in the LSTM network

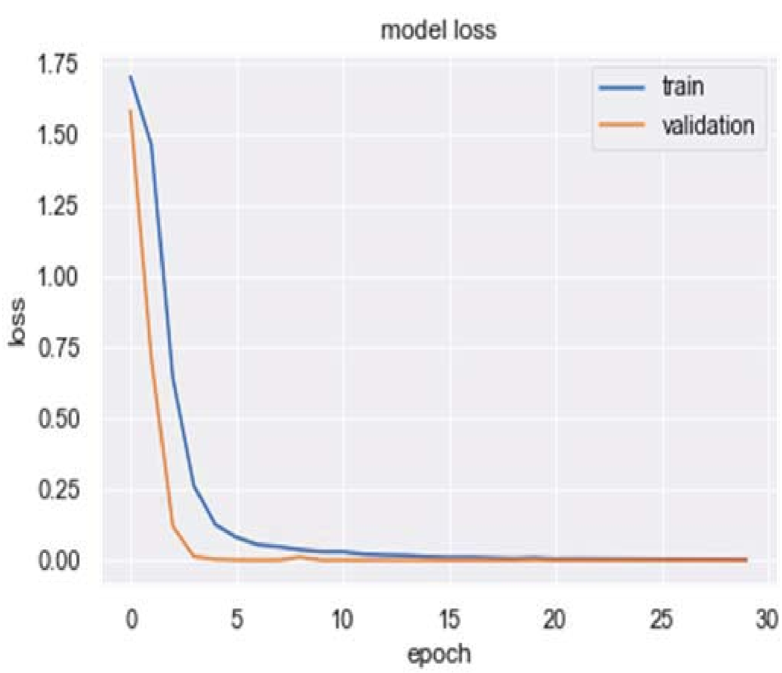

Fig. 13. Loss graph for DNN

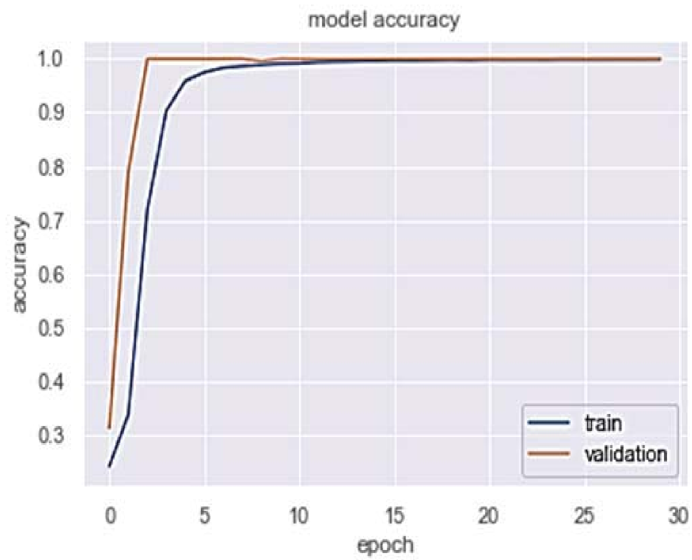

Fig. 14. Accuracy for DNN

4. Conclusion

Proposed System provides a new smarter traffic signal controller for the traffic lights was the objective of proposed work. The traffic signal switch incorporates a triple model setup which can be deployed at intersections where the traffic management systems are struggling to keep up with the rapidly growing population around and at a much lower investment cost than that of full-scale overhauls like constructing flyovers or underpasses. This method can become the primary foundation to expand upon and connect the whole road network to different artificial intelligence-based systems and create a future with self-driving vehicles or semi-self-driving vehicles that can predict and pre-plan routes as required. The Adaptive nature of the system helps it a lot and reduces the maintenance issues which might arise on a hardcoded timer-based system. This design can be further improved in regards of traffic forecasting and counting as more resources are available to run faster and bigger computations. Also, the scalability of deep learning models is very helpful when it comes to operations such as this where there are massive amounts of input data to learn from and store, this will allow easier upgradations to bigger datasets. Also helps in keeping the model up to date for future changes and fluctuations that might have been missed in the current sample set. There can be proper implementations and integration of this system within the Smart city sector and as well as in modern trends like Internet-of-things (IoT) by connecting other peripheral components to the vehicles and the traffic signal system which can communicate with each other with a better more fluid data sharing system. The aim and vision for proposed work to design a system for the current road systems which will mesh modern deep learning techniques for enhancing road traffic systems for a better more modern future. The models performed well and showed good enough results to be considered for deployment. This work is proposed for smart city roads while keeping in mind the existing infrastructure that will need an upgrade. This work is intended to be applied where a high to moderate amount of traffic could be expected.

5. Future work

The current system can have better performance in regards to detection accuracy and precision if better models such as CenterNet [21] can be implemented as mentioned before. The LSTM can also be improved upon by implementing Minmax scaled attention into it, which will improve the predictions at the higher end of the predicted values and make the model overall more reliable. Along with the previously mentioned improvements, more optimized patterns in switching and synchronizing the traffic lights can be explored. A few additional minor hyperparameter changes can also be explored in order to achieve improved efficiency. An improved version of data pipeline can be done by migrating to the SQL database or Hadoop ecosystem. It has to be done because in real-world implementation a lot of data will be generated and hence need for a robust data storage solution is felt.

References

1. Stevanovic A, Kergaye C, Stevanovic J., "Long Term Benefits of Adaptive Traffic Control under Varying Traffic Flows during Weekday Peak Hours", *Transportation Research Record*; 2311 (1): 99-107, 2012
2. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision", arXiv:1512.00567, 2015
3. Yizhe Wang, Xiaoguang Yang, Hailun Liang, Yangdong Liu, "A Review of the Self-Adaptive Traffic Signal Control System Based on Future Traffic Environment", *Journal of Advanced Transportation*, vol. 2018, Article ID 1096123, 12 pages, 2018 https://doi.org/10.1155/2018/1096123
4. P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, "e SCOOT online traffic signal optimization technique*," Traffic Engineering and Control*, vol. 23, no. 4, pp. 190–192, 1982.
5. A. G. Sims and K. W. Dobinson, "e Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 2, pp. 130–137, 1980
6. Sanchal Ramteke, and B. B. Gite, "AI BASED TRAFFIC SIGNAL CONTROL SYSTEM, *International Research Journal of Engineering and Technology (IRJET),* Volume: 06 Issue: 11, Nov 2019
7. Arif A. Bookseller, Rupali R Jagtap, "Image processing based Adaptive Traffic Control System", *Second International Conference on Emerging Trends in Engineering (SICETE), IOSR Journal of Electronics and Communication Engineering*, pp. 33-37, 2018
8. Duowei Li, Jianping Wu, Ming Xu, Ziheng Wang, Kezhen Hu, "Adaptive Traffic Signal Control Model on Intersections Based on Deep Reinforcement Learning", *Journal of Advanced Transportation, vol. 2020, Article ID 6505893*, 14 pages, 2020. https://doi.org/10.1155/2020/6505893
9. Yaw Okyere Adu Gyamfi, Sampson Kwasi Asare, Anuj Sharma, and Tienaah Titus, "Automated Vehicle Recognition with Deep Convolutional Neural Networks, Transportation Research Record" *Journal of the Transportation Research Board*, No. 2645, pp. 113–122, 2017
10. H. Siegel and D. Belomestnyi, "Stochasticity of Road Traffic Dynamics: Comprehensive Linear and Nonlinear Time Series Analysis on High-Resolution Freeway Traffic Records", Data Analysis, Statistics and Probability, 2006
11. Williams B M, Hoel L A, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664- 672, 2003.
12. R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China*, pp. 324-328, 2016, doi: 10.1109/YAC.2016.7804912
13. Ahmed, M. S., and A. R. Cooke, "Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques", Transportation Research Record 772, TRB, National Research Council, Washington, D.C., pp. 1-9, 1979
14. Lia, Hao Yub,Guohui Zhangc, Shangjia Dongd, Chengzhong Xu, "Network-wide Traffic Signal Control Optimization Using a Multi-agent Deep Reinforcement Learning", Transportation Research Part C: Emerging Technologies, Volume 125, April 2021
15. Alvaro Cabrejas Egea, Raymond Zhang, Neil Walton: "Reinforcement Learning for Traffic Signal Control: Comparison with Commercial Systems." CoRR abs/2104.10455, 2021
16. Wangzhi Li, Yaxing Cai, Ujwal Dinesha, Yongjie Fu, Xuan Di, "CVLight: Deep Reinforcement Learning for Adaptive Traffic Signal Control with Connected Vehicles" arXiv:2104.10340
17. James Ault, Guni Sharon, "Reinforcement Learning Benchmarks for Traffic Signal Control", *35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, June 2021
18. Bouktif, S.; Cheniki, A.; Ouni, A. "Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning." Sensors 2021.
    https://doi.org/10.3390/ s21072
19. Bradski, G, "The OpenCV Library", *Journal of Software Tools for the Professional Programmer*, 2000.
20. Lin TY. Et al. Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) "Computer Vision ECCV 2014." Lecture Notes in Computer Science, vol 8693. Springer, Cham. ,2014 https://doi.org/10.1007/978-3-319-10602-1_48
21. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, "Imagenet large scale visual recognition challenge.", 2014.
22. Kaiwen Duan and Song Bai and Lingxi Xie and Hong ang Qi and Qingming Huang and Qi Tian, "CenterNet: Keypoint Triplets for Object Detection" preprint 1904.08189, 2019.