International Journal of Innovative Research in Engineering
Volume 3, Issue 3 (May-June 2022), PP: 564-572.
www.theijire.com
ISSN No: 2582-8746

Automobile Safety System

Dr. J.B.Jona[1], S.A.Gunasekaran[2], C. G.Nivetheni[3], S.Saagarika[4], R.T.Subavarsshini[5]
[1,2] Department of Computer Applications, Coimbatore Institute of Technology, India
[3,4,5]Department of Computing, Coimbatore Institute of Technology, India

How to cite this paper:
Dr. J.B.Jona[1], S.A.Gunasekaran[2], C. G.Nivetheni[3] , S.Saagarika[4], R.T.Subavarsshini[5], "Automobile Safety System", IJIRE-V3I03-564-572.

Abstract: *Car License Plate Recognition is one of the important modules for a camera surveillance-based security system. This can be done by extracting the license plate from an image using computer vision techniques and by using Optical Character Recognition to recognize the license number. This project will also be making a drowsiness detection system. Day and night, a countless number of people drive on the highway. Truck drivers, bus drivers, and taxi drivers traveling long-distance suffer from a lack of sleep. Driving while sleepy becomes very dangerous as it may lead to accidents. Most accidents occur due to the drowsiness of the driver. The purpose of this project is to build a security system that prevents these accidents using Python, OpenCV, and Keras which will alert the driver when he feels sleepy. The aim of this project is to use the traffic database and apply the Deep Learning algorithms to it. This project can be used to help urban cities, especially smart cities, have a better understanding of traffic patterns. This leads us to better traffic management, which is a great need today. It can also be used to ensure that smart car drivers get their route without any further delays or traffic waiting time, solving many problems. Traffic forecasting can be used for infrastructure and security development. Traffic forecasts can be made in many cities. The main purpose of this project is to build an automated system that detects traffic signal violations and makes it easier for the traffic police department to monitor traffic and take action against the owner of the vehicle who drives in a fast manner. Detecting and tracking a vehicle and its operations accurately is a priority for the system.*

Keywords: *License Plate, Drowsiness detection, Python, OpenCV, Deep learning, Shortest part, Signal Violation.*

I. INTRODUCTION

Owing to the rise in vehicles day by day on the road, to detect the violated vehicles a License plate recognition (LPR) system is very important. Traffic rules violations, vehicle stealing, parking in restricted areas, all these violations are increasing linearly. Thus, to avoid these practices, license plate recognition is intended. Driver Drowsiness detection is a human protection system that prevents road accidents caused by drivers who fall asleep while driving. This system will detect whether a person's eyes are opened or closed even for a few seconds. Once the system identifies that the person's eye is closed, it will alert the driver with an alarm sound. Mostly truck drivers, bus drivers, taxi drivers, and long-distance commuters suffer from insomnia as they drive on the highway day and night. It is very dangerous to drive while feeling drowsy.Many accidents occur as a result of drowsiness. Therefore, to prevent these accidents this model will build a system using Python, OpenCV, and Keras that will alert the driver when he falls asleep.

The effects of traffic congestion on lost time, excess fuel consumption, and low speed and idle vehicles create air pollution. The Urban Mobility Report of 2019 states that traffic congestion in the US has resulted in a loss of 54 hours per passenger, which is 3.3 billion gallons of fuel and $166 billion in total. Lack of proper information on congestion can also cause increased delays, accidents, and other congestion due to inexperienced travel options. The impact of congestion can be reduced through the provision of advanced information on navigation systems and highway signals. Concentration congestion can also provide insight into congestion analysis and identify traffic flow constraints, which can assist in strategic planning and implementation. The need to use mathematical or machine learning models to predict or analyze congestion is that traffic flow is very stochastic. There are many confusing factors such as road conditions, daytime variations, special events, weather, and natural variations in human decision-making.

The increasing number of cars in cities can cause a huge amount of traffic, and it means that traffic violations are becoming more serious today all over the world. This creates property damage and many accidents that can endanger human life. To solve the alarm problem and prevent such mysterious consequences, traffic violation systems are the system that applied the relevant traffic rules at all times, and held those who did not comply with them. The process of detecting traffic violations must be done in real time as the authorities are constantly monitoring the roads. Therefore, road operators will not only be free to use safe roads more accurately but also more efficiently; as the traffic acquisition system detects violations faster than humans. This program can detect robots in real time. An easy-to-use graphic interface is integrated with the system to make it easier for the user to operate, monitor traffic, and take action against the violated vehicles.

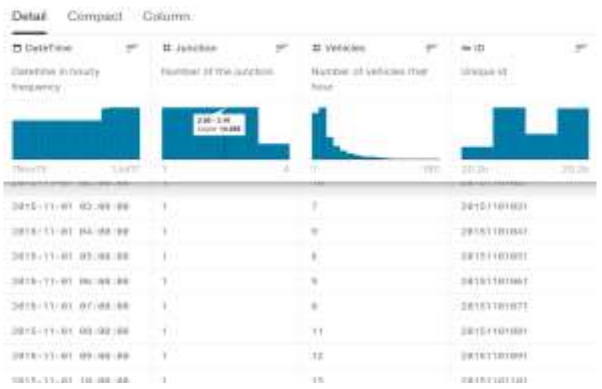
II.DATASET & FEATURES

License plate dataset

The License Plates data set is the database of different vehicles (i.e. cars, vans, etc.) and their number plates. Annotations include examples of "car" and "license plate". This database may be used to create a vehicle acquisition model with numerical values.

Traffic dataset

The dataset contains date time, vehicles, and traffic information. Traffic congestion is increasing in cities around the world. Impact factors include an increase in urban population, aging infrastructure, inefficient and unstructured traffic signal time, and a lack of real-time data. Each of these sensors has been collecting data at different times. Some of the frameworks provided limited or partial data that needed to be considered when constructing future estimates.



DateTime	Junction	Vehicles	ID
Datetime in hourly frequency	Number of the junction	Number of vehicles that hour	Unique id
2015-11-01 03:00:00	1	7	20151101031
2015-11-01 04:00:00	1	9	20151101041
2015-11-01 05:00:00	1	8	20151101051
2015-11-01 06:00:00	1	9	20151101061
2015-11-01 07:00:00	1	8	20151101071
2015-11-01 08:00:00	1	11	20151101081
2015-11-01 09:00:00	1	12	20151101091
2015-11-01 10:00:00	1	15	20151101101

Drowsiness dataset

The dataset for this model has been built manually by the authors. To create the dataset, a script has been written that captures eyes from a camera and stores them in the local disk. It is then separated into their respective labels ‘Open’ or ‘Closed’. The data were cleaned manually by removing the unwanted images which are not necessary for model building. The data consists of about 7000 images of people’s eyes under different lighting conditions.

Tools and Techniques

The dataset that has been collected are processed and analyzed using Python. Open Computer Vision (OpenCV) library has been utilized for image Pre-processing and TensorFlow and Keras are implemented for performing deep learning tasks, NumPy for matrix computations, and Pandas for manipulation. Flask is used for deployment. GRU model has been used for traffic prediction in the junction. YOLO model for license plate detection.

Technologies: Python, Jupyter Notebook.
Libraries: Pandas, NumPy, matplotlib, Seaborn, scikit-learn, Keras, and TensorFlow

Packages installed: Dlib, OpenCV-4.2.0.32, Pyre base- 3.0.27, smtplib, threading, requests.

Data Pre-processing

Since the data license plate and eyes are image data and it cannot be read or processed by the computer, This system applies a significant amount of image pre-processing. Using OpenCV the model converts all the images to gray color. With the use of OpenCV, the images are being resized into the standard format of 100 by 100. The images are labeled according to their type. The image is divided by 255 and reshaped into a four-dimensional array for feeding it into the neural network and the target was also reshaped into an array.

In pre-process method, the following steps are being followed:

1. Blur the Image.
2. Conversion to Grayscale.
3. Finding the vertical edges of the image.
4. Threshold of the vertical-edged image.
5. Close Morph the Threshold image.
6. Method extract_contours returns all external contours from the pre-processed image.
7. Using the method find_possible_plates preprocess, the image is being pre-processed. It is then extracted by using contours with the extract_contours method. It will then check the side ratios and also the area of all extracted contours. The image is being cleaned inside the contour with check_plate and clean_plate methods. When the cleaning of the contour image with the clean_plate method is completed, it will find all characters on the plate with the find_characters_on_plate method.
8. The find_characters_on_plate methodology will use the segment_chars' function to find the characters. It works by finding the characters by computing the convex hull of the contours of a thresholded value image and drawing it on the characters to reveal them.
9. In order to detect the face in the image, it must be first converted into grayscale as the OpenCV algorithm for object detection takes gray images in the input.
10. The isn't any necessity to know the color information to detect the objects. The usage of the heap cascade classifier is implemented for face detection.
    11. The setting of the classifier face is done by using the function, cv2.CascadeClassifier('path to the desired haar cascade XML file').

III.OBJECTIVE

Passenger safety is a top priority in the automotive industry today. Participants throughout the vehicle value chain acknowledge the importance of passenger/passenger safety and continually improve their delivery to provide failed safety technologies that will protect passengers and pedestrians. The implementation of policy and consumer awareness has played a key role in making popular car safety systems popular. Vehicle safety learning and operation design, construction, equipment, and controls to reduce the occurrence and consequences of motor vehicle accidents.

IV.METHODOLOGY

Baseline

Neural Network is a machine learning algorithm, based on the organization and operation of neural networks. The idea arose in an effort to mimic brain processes by Warren McCulloch and Walter Pitts in 1943.Neural networks consist of individual units called neurons. Neurons are found in a series of groups - layers. Every neuron in each layer will be connected to neurons of the next layer. The data comes from the input layer in the output layer next to these compounds. Each node performs a simple statistical calculation. When transferring its data to all nodes connected to it.Convolutional neural networks (CNN) are a unique structure for artificial neural networks, proposed by Yann LeCun in 1988. CNN uses other features of the visual cortex. One of the most popular of these architectures is image separation. For example, Facebook uses CNN in automated tagging algorithms, and Amazon - for making product recommendations with Google - to search between user photos.CNN has broken the mold and ascended the throne to become a state-of-the-art computer vision system. Among the various types of neural networks (some including recurrent neural networks (RNN), long-term memory (LSTM), artificial neural networks (ANN), etc.), CNN's are easily recognizable. in the image data space. They are very good at computer viewing tasks such as image classification, object detection, image, etc.

Image classification involves the removal of elements from an image to see other patterns in the database. The use of ANN for the purpose of image separation can end up being more expensive in terms of calculation as the training parameters become extremely large. For example, on having a 50 X 50 cat image, and want to train our traditional ANN on that image to separate it from a dog or cat trainees (50 * 50) * 100 pixels of image enhanced by a hidden layer + 100 bias + 2 * 100 output neurons + 2 bias = 2,50,302.Filters are used when using CNN. Filters have many different types in terms of their purpose. Filters help us to apply the location of a particular image by forcing a local connection pattern between neurons. The Convolution layer is always first. The image (matrix with pixel values) is embedded in it. Assume that the input matrix reading starts at the top left of the image. Next, the software selects a small matrix there, called a filter (or neuron, or total). After that, the filter produces a resolution, e.g. It goes with the input image. The function of the filter is to multiply its values by actual pixel values. All of these repetitions are summarized. Number one is finally found. As the filter reads the image in the top left corner, it moves forward to the right with 1 unit doing the same function. After passing the filter to all positions, a matrix is obtained, but smaller than the input matrix.This functionality, from a human point of view, is similar to pointing out simple borders and colors in an image. But in order to see high-quality structures like the trunk or large ears the whole network is needed. The network will have several convolutional networks integrated with offline and integration layers. When an image passes through a single layer of convolution, the removal of the first layer becomes the input of the second layer. And this happens with every other layer of clarity.

A non-linear layer is added after each convolution function. It has an activation function, which brings non-linear assets. Without this structure, the network would not be powerful enough and would not be able to mimic the flexibility of the a class label). The pooling layer follows an indirect layer. It works with the width and height of the image and does the work of lowering the sample to them. The final result is that the image volume is reduced. This means that once certain features (e.g. parameters) have already been identified in the previous convolution operation, the detailed image is no longer required to continue functioning, and is compressed into less detailed images.After the completion of a series of convolutional, nonlinear, and pooling layers, it is necessary to attach a completely connected layer. This layer captures extraction data from convolutional networks. Attach a fully connected layer at end of the network results in the N vector, where N is the number of classes.Classification of the vehicle from the given video footage, moving objects are detected. Object detection model named YOLOv3 is used to classify those moving objects into respective classes. YOLOv3 is the third object detection algorithm in YOLO (You Only Look Once) family. This has indeed increased the accuracy by using many tricks and is more capable of detecting objects. The above classifier model is being built with the use of Darknet-53 architecture.

The dataset has four junctions, which are treated separately for modeling. Then, the data is split into 85% of the training set and 25% of the testing set, as the dataset has to be trained as much as possible. Then, the chosen model for this project is the Gated Recurrent Unit (GRU). The important features are Gated Recurrent Unit (GRU): ● Construct six layers. ● Use tanh as the activation function. ● For performance optimization, stochastic gradient descent has been used. GRU or Gated Recurrent Unit is the development of a common RNN. GRUs are very similar to Short-Term Memory (LSTM). Like LSTM, GRU uses gateways to control the flow of information. They are new compared to LSTM. This is the reason why they offer some improvements with LSTM and have simple structures. Another interesting thing about GRU is that, unlike LSTM, it does not have a separate cell (Ct) status. Contains only hidden mode (Ht). Due to the simple structures, GRUs are quick to train.

Model Process

To reduce the noise, the system needs to blur the framed image with Gaussian Blur and turn it on a grayscale. Find the straight edges on the picture. To reveal the plate, the picture must be taken in pairs. For this use Otsu's Thresholding on the straight edge of the image. For other mitigation methods, it has to select the limit value to make the image double but Otsu's Thresholding determines the value automatically. Apply Closing morphological Transformation in the border image. Closing helps to fill small black circles between white circles in the cut image.Displays a rectangular white plate for the license plate. To find the plate the system needs to find the links in the picture. Resize the image before getting the concerts to find the most appropriate and minimum number of concerts in the photo. Now find the minimum rectangle of the area closed by each contract and confirm their side measurements with the area. It is being defined as the minimum and maximum area of the plate as 4500 and 30000 respectively.

Now find the contour in the designated area and confirm the side measurements and location of the largest contour rectangle in that region. After verification, you will receive the appropriate contour of the license plate. Now remove that contour from the first image. To identify the license characters accurately, image separation is done. That first step is to extract the value channel in the HSV format of the image of the plate. It may look like it. Image to split it in half and highlight the characters. A plate image can have different lighting conditions in different areas, in which case the variable threshold may be more suitable for binarization because it uses different values of different regions based on pixel intensity in the surrounding region. After combining both, usage of bitwise and not, it will work on the image to get the parts connected to the image so that the removal of the character candidates is easier.

With a webcam, it will take pictures as input. So, in order to access the webcam, endless loops have been created that will capture each frame. It uses OpenCV, cv2.VideoCapture (0) to access the camera and set the recording object The same facial recognition process is used to see the eyes. First, it is important to place the cascade of the eye on the eye and the rear respectively, and then find the eyes using left_eye = leye. detectMultiScale(gray) Now it will have to extract the optical data only from the full image. This can be achieved by removing the eye border box so that it is easier to extract the eye image from the frame with this code. L_eye contains only eye image data. This will be included in our CNN article which will predict that the eyes are open or closed. Similarly, the removal of the right eye from the r_eye. The CNN section is used to predict eye conditions.

In order to upload our image to a model, it has to perform certain tasks because the model needs the right size to start with. First, the conversion of the color image to grayscale using r_eye = cv2.cvtColor (r_eye, cv2.COLOR_BGR2GRAY). Then, the image is adjusted to the size of 24 * 24 pixels as our model trained on 24 * 24-pixel cv2.resize images (r_eye, (24,24)). Own data is being made so that it fits better r_eye = r_eye / 255 (All values will be between 0-1). Extend size to feed to our separator. The model downloaded is loaded using, model = load_model (‘models / cnnCat2.h5’).Prediction of each eye with the model is being done. lpred = model. predict_classes (l_eye). If the value of lpred [0] = 1, it says the eyes are open, if the value lpred [0] = 0 then, it means the eyes are closed.Points are actually the amount that is used to determine how long a person has closed their eyes. So, if both eyes are closed, it will continue to increase points and when the eyes are open, it will reduce the points. Draw the effect on the screen using the cv2.putText () function which will show the real-time status of the person. A threshold is defined for example when points become greater than 15 which means that a person's eyes are closed for a long time. This is where the sound of an alarm is played using sound. play ().

Model Implementation

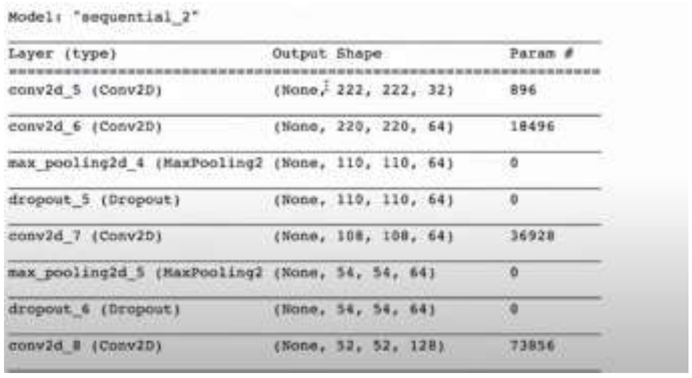
The first convolutional layer of CNN has a layer with 32 filters. The number of filters is small in the beginning because the lower layers detect the features in very small parts of the image. The model gets the very minute features like straight lines and smoothened curves as it is trying to learn the patterns that are hidden. Looking deep into the network the connections of the CNN model keep increasing. That means the feature that the layers extract is a bigger portion of the image. 32 filters mean extracting 32 different features extracted in the first layer. It's a 2D convolution with the kernel size as 3x3 and the activation as 'relu'. An additional convolutional layer with 64 filters and size 3x3 with activation as relu is also added.

Then the addition of a max pooling layer with the default size as 2x2. Then, by adding many layers in the convolutional neural network, the input for the next layer will be the output of the first layer and so on, which yields such an effective output. Then a dropout layer is added to find the overfitting. The same is repeated for another layer by increasing the filter size by 64 and 128. Then the addition of the flatten and the dense layer is done. The model uses Flattering as a layer to change the dataset into a one-dimensional array in-order to store the output for the next layer. By flattering the outcome of the layers in CNN will give a long single featured vector. It is combined with the concluding classification model, which is known as a fully connected convolutional layer. High and deep layers that are connected give us the features of learnings from all the joint features from the previous layer, whereas a convolutional layer, relies on consistent features with a small repetitive field. The output layer is added with a sigmoid activation function to predict the probability between 1 to 0.

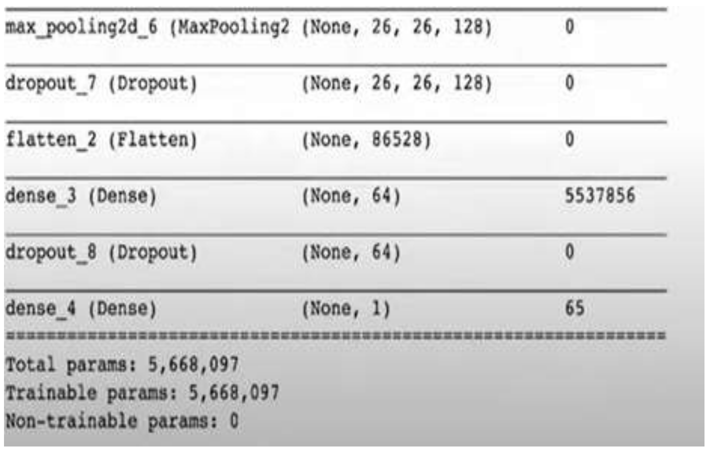
The model is compiled with the 'binary cross entropy loss' as it compares each of the predicted probabilities to the actual class output which can be either 0 or 1. The model then finds the actual score that reduces the chance probabilities which is based on the expected value's distance. With this, it is seen how close or far from the actual value is. The usage of the Adam optimizer combines the best properties of the AdaGrad and an algorithm named RMSProp helps to get an optimizing algorithm that is able to reduce the sparse gradients on noisy problems. And the metric to evaluate our model is accurate.

Model testing

The data was split into training and validation which was a 90-10 split. The best training model was saved based on the idea that the best model should have a validation loss less than the previous epoch so that only the best model is safe and not all the models.



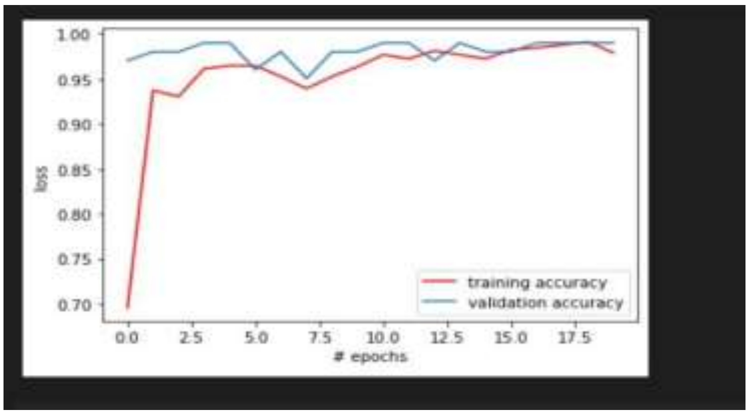
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 222, 222, 32)      896
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 220, 220, 64)      18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 110, 110, 64)      0
_________________________________________________________________
dropout_5 (Dropout)          (None, 110, 110, 64)      0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 108, 108, 64)      36928
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_6 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
```

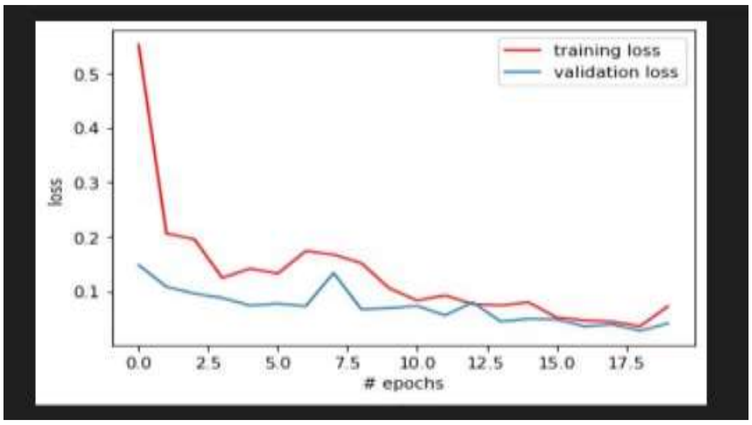


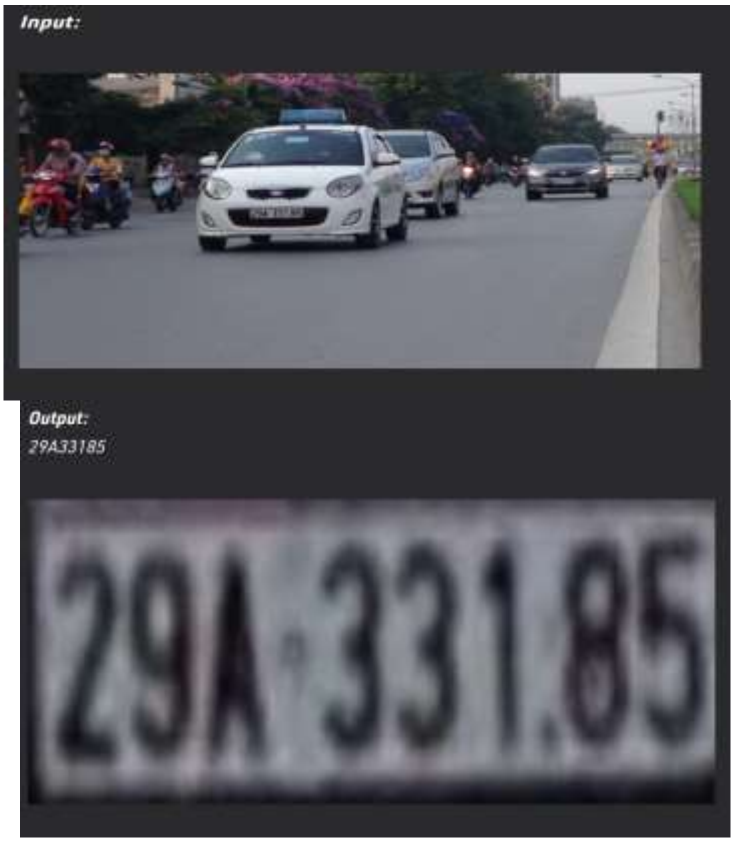
```
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_7 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
flatten_2 (Flatten)          (None, 86528)             0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                5537856
_________________________________________________________________
dropout_8 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 65
=================================================================
Total params: 5,668,097
Trainable params: 5,668,097
Non-trainable params: 0
```

V.RESULTS & DISCUSSION

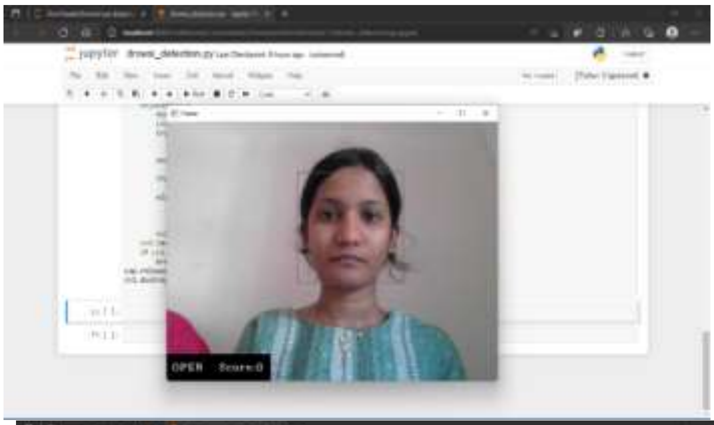
The model has performed well as the training loss eventually decreases, even though the validation loss fluctuates it settles at a decreasing rate. From the graph, it is clear that there is not much evidence for overfitting so it can be concluded by saying it is a quite good model. The best thing is that the loss and accuracy are quite similar.

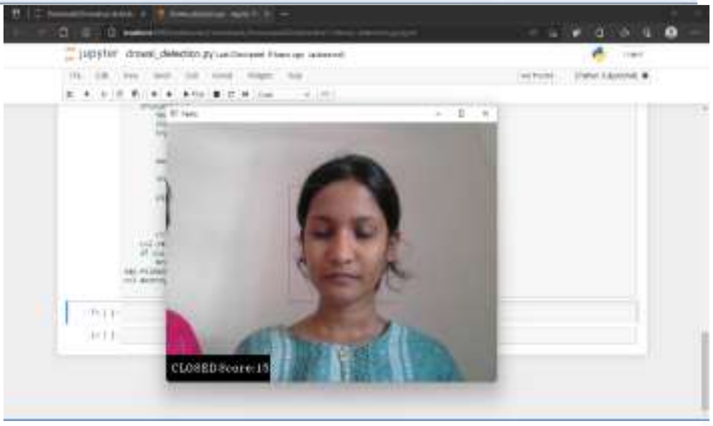


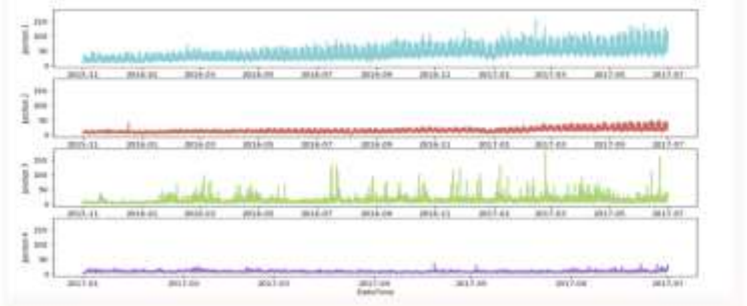


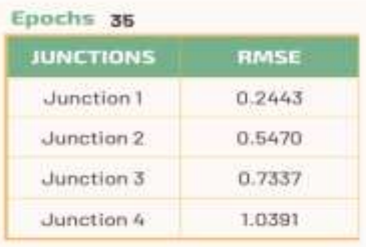

Input:

Output:
29A33185

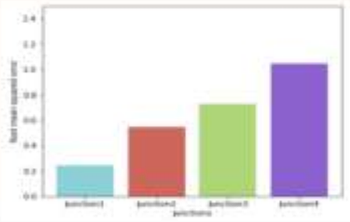


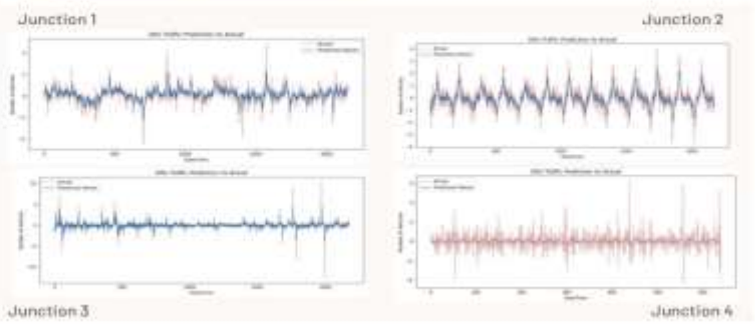


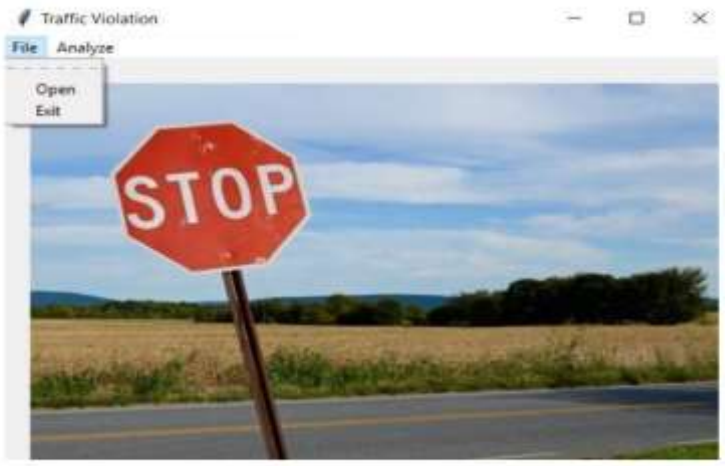

Timeseries of traffic for each junction over the year.

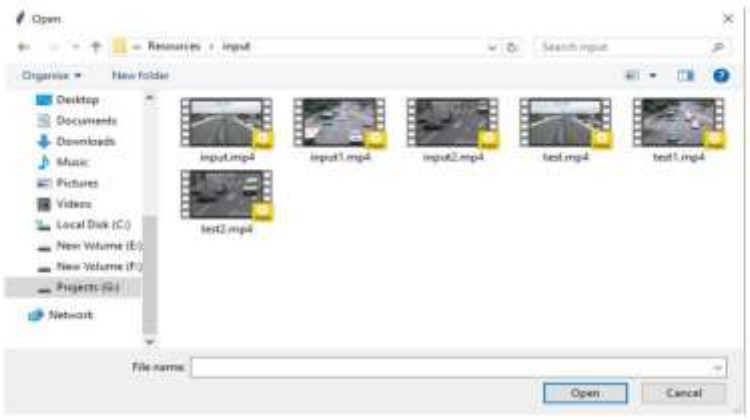

Epochs 35

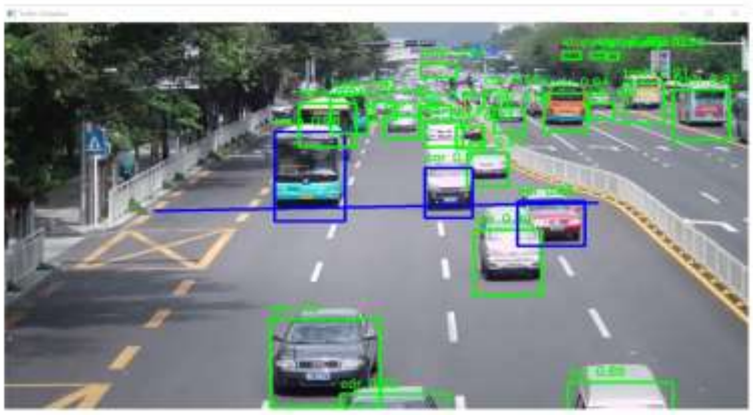
JUNCTIONS	RMSE
Junction 1	0.2443
Junction 2	0.5470
Junction 3	0.7337
Junction 4	1.0391











VI.LIMITATIONS OF THE STUDY

Attackers may gain access to either open areas or areas not covered by the security system, or have the technical ability to bypass the system. System setup is a costly task as it will require advanced configuration equipment to get accurate results. Lack of knowledge is a major problem to consider. The data you need with all the required parameters is hard to find.

VII.FUTURE SCOPE OF THE STUDY

The combination of the traffic violation detection mentioned is not the same, as there is a different boundary condition. The system provides for the detection of traffic signal violations. In addition, the system is capable of processing one data at a time. Also, the operating time of the system is relatively slow and can be improved by using a computer with a high-speed processor or GPU specification. Future research on the use of an algorithm designed for other advanced image processing techniques. As such, this may improve the operating time of the system by ignoring other unnecessary steps performed in the form of a background difference. A computer vision algorithm can be developed instead of providing a lot of ingenuity in the system.

VIII.CONCLUSION

Although there are many license plate recognition systems, drowsiness systems, and traffic violation systems existing, the proposed system is the first of its kind of an integrated module of most of the automobile security features. The system is tested over a large number of vehicle datasets and is proved to be 90% accurate. The developed system is a non-invasive system. The system can be further developed by adding various types of sensors. The system is aimed at assisting drivers, and more particularly at reminding them of the presence of some specific traffic signs on the road. The system is built especially for the traffic police authorities to manage the traffic violations happening in the real world.

Reference

[1]. Https://Journals.Sagepub.Com/Doi/Full/10.117 7/2472630320958376
[2]. Https://Arxiv.Org/Abs/2005.01578
[3]. Https://Ieeexplore.Ieee.Org/Document/9231235
[4]. Https://Www.Ncbi.Nlm.Nih.Gov/Pmc/Articles/P Mc7874961/
[5]. Https://Www.Sciencedirect.Com/Science/Articl E/Abs/Pii/S1361841520301584
[6]. Https://Www.Analyticsvidhya.Com/Blog/2021/ 01/Image-Classification-Using-Convolutional-
[7]. Neural-Networks-A-Step-By-Step-Guide/
[8].Https://Medium.Com/@Raghavprabhu/Understanding-Of-Convolutional-Neural-Network-Cnn-Deep-Learning-997608 35f148