| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

Deep Learning and Cloud based Driver Monitoring System

Karthik Raj S S[1], Jaffar Aleem[2], Gagan L Naik[3], Likith S[4], K Sujatha[5]

UG Student, Dept. of ECE, BMS College of Engineering, Bangalore, Karnataka, India[1]

UG Student, Dept. of ECE, BMS College of Engineering, Bangalore, Karnataka, India[2]

UG Student, Dept. of ECE, BMS College of Engineering, Bangalore, Karnataka, India[3]

UG Student, Dept. of ECE, BMS College of Engineering, Bangalore, Karnataka, India[4]

Associate Professor, Dept. of ECE, BMS College of Engineering, Bangalore, Karnataka, India[5]

ABSTRACT: Every day thousands of people die due to road accidents and many suffer permanent injuries; Driver distraction is a significant factor in a large number of these motor-vehicle accidents. In this paper, we have presented a methodology to detect driver inattention and alert him on the same. A trained and tested deep learning model is implemented to detect any distractions like talking & texting on phone, reaching radio, turning behind, hair & makeup. At the time of negligence driver is alerted and the violation information like time & type of violation, driver ID, name of driver, GPS coordinates, and a snapshot of violation is uploaded to cloud storage. This would ensure easier law enforcement and safe driving.

KEYWORDS: Deep Learning, Convolutional Neural Network (CNN), Computer Vision, Cloud Computing.

I.INTRODUCTION

It is within the law that drivers must be focused during driving and any secondary distractions could lead to fatal instances not only for passengers within the vehicle but also to the by-passers. These can lead to fatal accidents. A survey conducted by SaveLIFE foundation from 2010 to 2019 tells that over the decade, 1.3 million Indians have been killed and 5.3 million have been disabled for life; out of which 140 thousand are children. It is concluded that driver distraction accounts for 84% of the accidents and usage of mobile phones as a cause tops the list at 47% of total distracted accidents.

This paper is mainly aimed that bringing those numbers to a minimum by use of a computer vision system and machine learning using the dataset available in Kaggle to alert the driver at the time of distraction and prevent road accidents. This data set was released by an insurance company called State Farm. The data set consists of images that were taken inside a car with the driver performing some actions. The actions were divided into classes such as cell phones for texting/calling, drinking, operating the radio, adjusting hair/makeup, turning behind, and talking to passengers. The entire lifecycle of the project has been phased into data collection, data pre-processing, training, evaluation considering both the false positives and false negatives. To observe the visual patterns like hand movement, head rotation or phone usage Convolution Neural Network (CNN) has been used to develop the neural network architecture. Also, an alarm is sounded to the driver when he is performing secondary tasks and the data is also simultaneously uploaded to a cloud platform.

This can be used on a personal scale to get an alert when the driver is distracted due to any secondary causes and help focus on driving. At the end of the journey, the driver can assess his driving abilities based on the data collected. This can be extremely useful to cab aggregators to monitor their drivers to ensure passenger safety. A data monitoring person can observe the cloud data and warn the drivers appropriately.The integration of this alert system with the existing technology infrastructure of respective transport departments of countries can aid the government in keeping off unsafe drivers from the road and also for smooth road law enforcement. The integration of this alert system by car manufacturers can aid in improving car safety and in the overall development of the car.

II.LITERATURE SURVEY

Sleep and fatigue are the main reasons for major accidents on the high way, but there are driver assistant systems to alert the driver for this purpose. This system focuses on the face of the driver with a camera on the dashboard to

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

generate a live video stream[1]. There are face recognition systems that recognize if the driver's eyes are open or closed to warn the driver with an alarm system. These systems are coupled with other safety features that are incorporated in advanced driver assistant systems[2]. For early detection of sleep and fatigue a yawn detection system is used [3]. Neural Networks are used for image recognition in all these systems [4]. Adaptive voice alerts can be used to alert according to the moods of the driver [5]. Cognitive emotion detection systems are developed for this purpose.Using a mobile phone is another main distraction for accidents. CNN can similarly be used for this purpose also [6]. Efficient way to detect secondary distractions is by detecting body postures [7], CNN can be used for the same, provided a good data set is present [8]. A good data set would have a sufficient amount of data equally distributed among all classes. There is also a need to increase the number of distractions that need to be detected. For this purpose, we use the Kaggle data set[9] which contains around 22424 images in the training set and 79792 images in the test set. Each image has a resolution of 480x640 with 10 different classes. [10] Has implemented machine learning models using the above-mentioned data set. CNN seems to be the best method for the proposed solution. Also, we could enhance the entire system by using the cloud to have a centralized monitoring system.

III.SYSTEM DESIGNAND IMPLEMENTATION

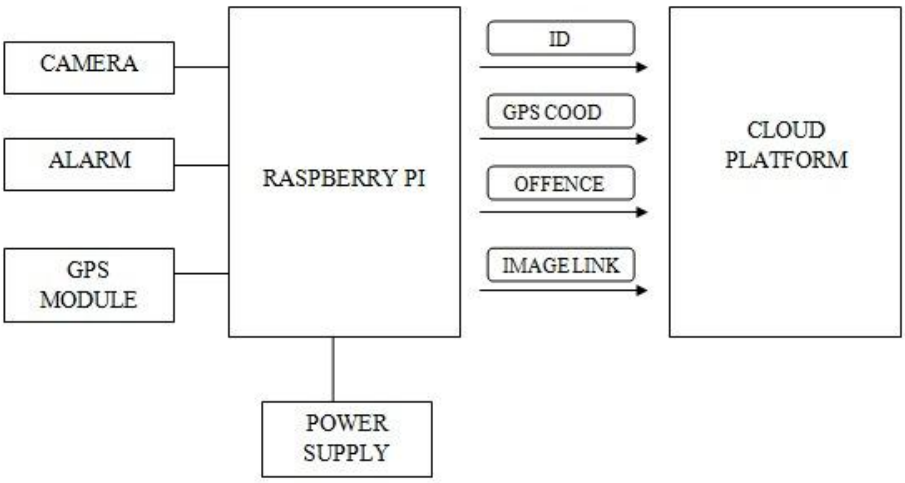

Fig. 1 Block Diagram of the system

The overall block diagram of the system and implementation flow chart is illustrated in fig 1 and 2 respectively. Raspberry Pi Model B is used as the main controller. This provides a Raspbian OS, a Linux distribution that enables us to run CNN models. A standard power supply of 5V and 2A is used to power the controller. A webcam is used to observe the live feed of the driver and predict the driver actions. When a violation is observed a speaker is used to sound a warning to the driver, the GPS module is to sample coordinates to the cloud. Along with this the driver ID, the type of offense, and the image are also sampled as proof. The controller is connected to the Internet to enable data transfer to the cloud. These are also stored locally to act like a black box for the vehicle.

The system implementation can be divided into 2 parts, the CNN model and the cloud part. The entire system is triggered when the vehicle engine is switched on. Firstly the camera is activated to provide a live video feed to the model. The model predicts on these frames if a violation is predicted a warning alarm is sounded to the driver and also the data (Driver ID, Co-ordinates, type of offense, and Image) is sampled to cloud. This data is also simultaneously stored in the local memory. If there is no violation detected the system goes back to predict the live video feed until a violation is detected.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

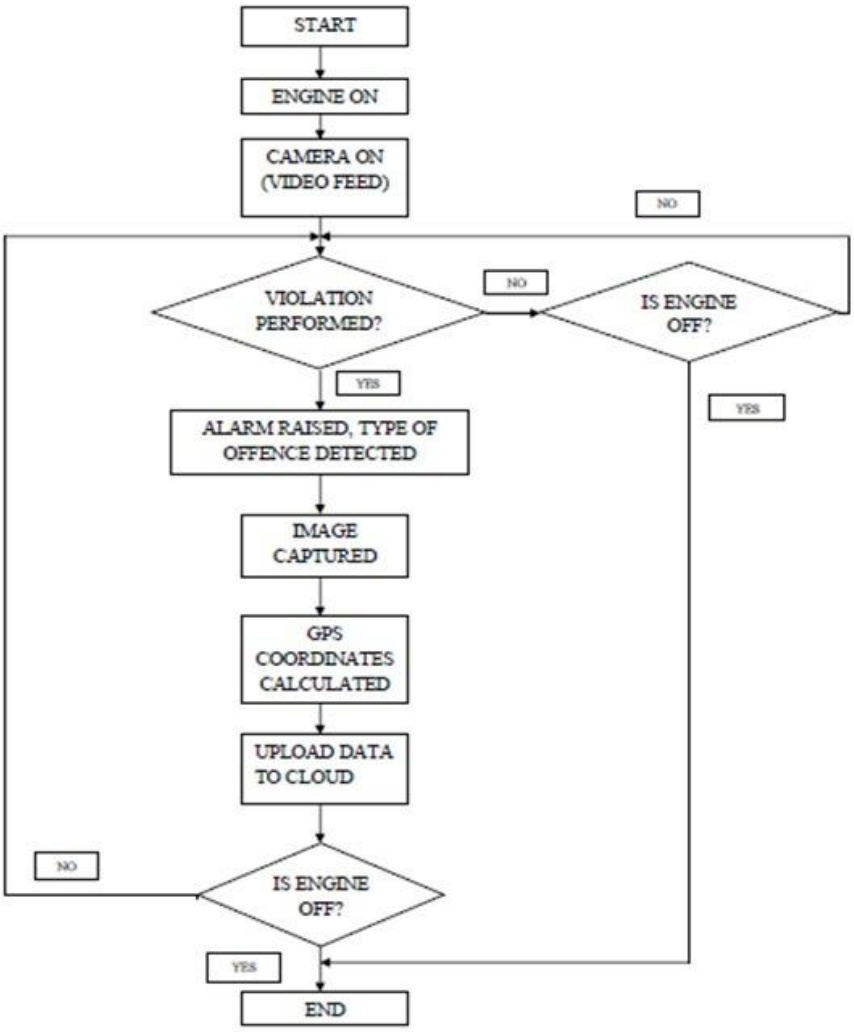

Fig. 2 Implementation Flow Chart

IV.CONVOLUTIONAL NEURAL NETWORK

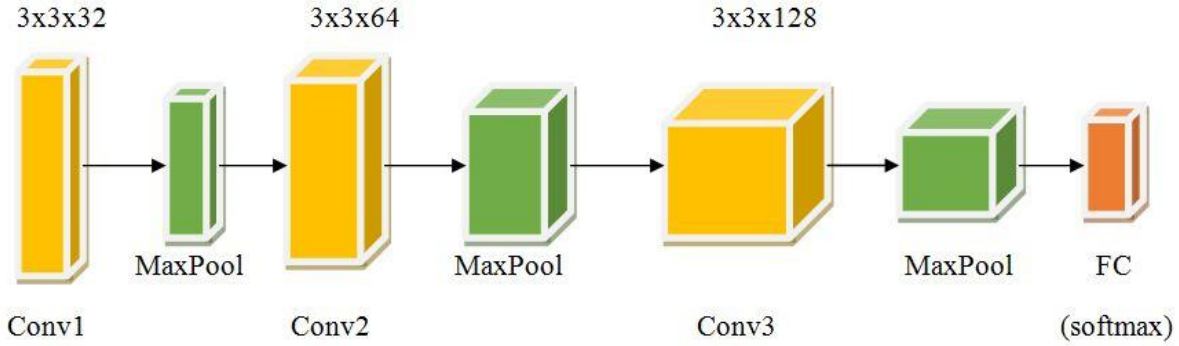

Fig. 3 CNN model

Firstly, during data pre-processing, we convert all the training images from RGB to a greyscale to reduce the number of computations. To further reduce the computations and speed up the process we reduce the resolution of the training images to 224x224 and normalize the data by a value of 255. Out of 22424 training images, we split it in such a way that 17939 images are used during the training process and 4485 images were used for validation. The data set contains

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

only images for left-hand side driving cars. For our country utilization, we horizontally flipped the images during pre-processing to create a separate model for the same. We also employ image augmentation techniques to ensure a better model development. Open CV library was used for image pre-processing and further implementation was done by using Keras API.

A small neural network model with just one input layer and an output layer with a fully connected softmax layer was constructed. This was just to give an idea about how to train the model in the future and to have a look at the factors affecting the model in its performance. As expected the model performed very poorly. The model gave an accuracy of about 30 % on training and the test set. Then a new network was designed as shown in fig 3. The convolution layerperforms convolution operation on the set of an image matrix. This operation extracts the high-level features of an image.The first layer is typically responsible for capturing features such as the edges of an image.The architecture, with added layers, adapts to capture the image's high-level features, giving us a deep neural network with a deep understanding of images in the dataset, similar to how a human brain would identify an image. To prevent overfitting dropout regularization technique was used.

Activation functions are mathematical functions that are used to observe the output at the particular node. It is a non-linear transformation that we do over the input. For intermediate nodes, we use the Relu activation function and the last layer is a softmax layer.Relu is the most commonly used activation function in neural networks, as neural networks often learn faster with Relu compared to other activation functions, and the reason is that there is less of a slope effect that slows learning to zero. Even though the slope is zero when z=0, but the probability of z being 0 or less than 0 is very low.

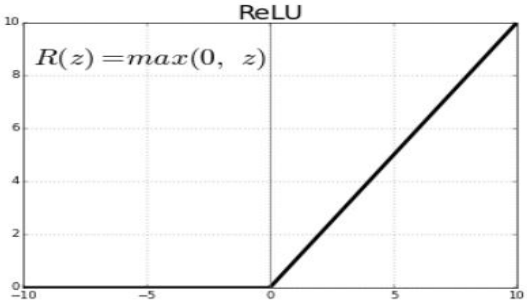

Fig. 4 Relu activation function

Softmax is a type of activation function which is applied to the last layer. It is used when we want to classify an image with more than two classes.It helps change the last linear layer of a multiclass classification network into probabilities by taking the exponent of each output and then standardizing by the sum of all the exponents. The first node produces the probability of first class image, the second node gives the probability of second class image, and so on.It got its name softmax because unlike hardmax where the output is in the form of 1 or 0, in softmax, we get the probabilities of an image belonging to different classes. Since the model is a 10 class classification system, there are 10 nodes in the output layer.

SOFTMAX

$$S(y_i) = \frac{e^{y_i}}{\sum e^{y_j}}$$

Fig. 5 Softmax activation function

During Forward Propagation pixels of images are fed in the form of numerical values.Weights, biases, and filters are initialised randomly.These values are considered parameters of the convolution neural network algorithm. The model attempts to update the parameters in the backward propagation process, so that the overall predictions are more accurate. We use gradient descent to update the backward parameters.Convnets also use pooling layers to reduce the representation size, speed up the computation and make some of the features that it detects a little more robust. Max Pooling is one of the types of pooling layers. In maxpooling, we observe the fixed number of input pixels and choose the maximum value pixel out of it and fill that pixel value in the output. This fixed number is decided by filter size and

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

stride value assigned to max pooling. One interesting thing about Pooling layers is that they do not have and parameters to be learned during backpropagation.

It is not possible to pass the entire batch of the training data all at once to the neural network; hence we divide it into batches. Batch size indicates the number of training examples in each batch. An epoch is the number of times we pass through a data set. We need multiple epochs because each time the weights of the neural network change leading it from an underfitting model to an optimum one. The batch size is set at 32 and the epoch number is set at 20.

IV.CLOUD CONNECTIVITY

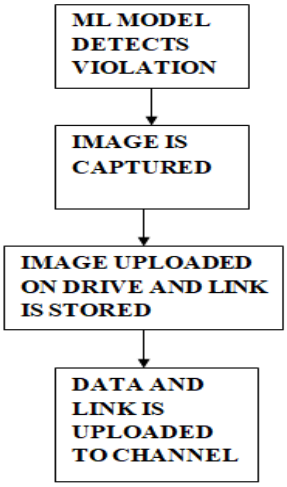

Fig 6Cloud Implementation

When the Deep Learning model detects a violation it must be recorded. To store all the violations of a user and allow authorities to view and analyze the data, Thingspeak, a cloud-based service offered by MATLAB is used. The data includes driver name, GPS coordinates where the violation was done, the type of offense, and the image link is uploaded on the cloud. To perform this task we used Python's URL library features. Initially, a channel on Thingspeak was setup. A private channel was created and 4 fields were defined. After creating the channel, by using the write API key and URL library features data uploading was possible.

Unfortunately, Thingspeak Cloud Platform does not have the capability of storing images. Hence to store images we tried two approaches. The first approach was to convert the image to base64 format. The base64 format converts images to encrypted text. However, these texts would be very long (usually 200,000 characters long). Uploading such large text on Thingspeak was not possible. The second approach was to upload the image onto a DRIVE folder and then share the link to Thingspeak channel. This approach would also help to store the images in an organized manner. If a violation is committed than the image would be stored in a drive folder under that driver's ID and then the link would be sent to the Thingspeak channel. When the excel file from Thingspeak is downloaded upon clicking the link the image would be displayed.The cloud service used to store the images was Mega Cloud which is a New Zealand based cloud service. They have APIs which enable us to upload images without having to manually upload the image. They have a higher level of security and also provide more storage space compared to other cloud services.

V. RESULT AND DISCUSSION

After training the model for 20 epochs, the model gave us an accuracy of 91.07% and validation accuracy of 98.08%. Log loss is used to observe the performance of the model.We were able to achieve a log loss of 0.0746. This is illustrated in figure 7. The model although is slightly overfitting and can be further improved to obtain a better model.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

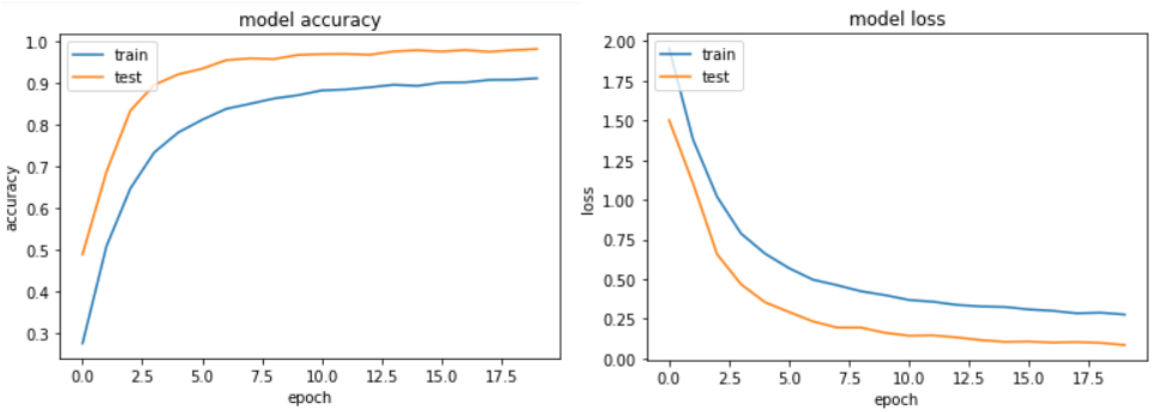

Fig. 7Accuracy and loss graph

A virtual environment in Raspberry Pi was treated to run the model. Fig 8 shows the output log of the model running on Raspberry Pi which was provided with an input video stream. All the violations recorded are stored in the cloud. Fig 9 represents the image storage in the Mega cloud. Fig 10 and 11 represent the data collected on the ThingsSpeak platform.

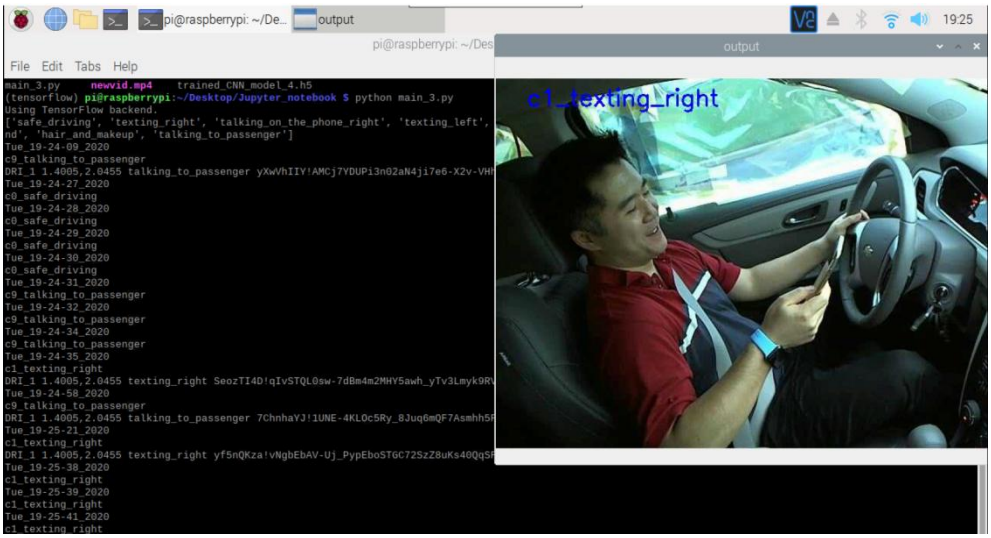

Fig. 8 Output log of the model running on Raspberry Pi 3

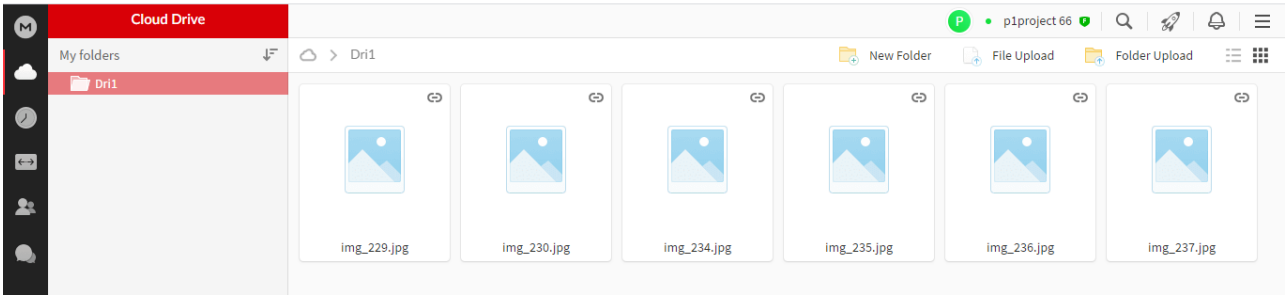

Fig. 9 Images stored in Mega cloud platform

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

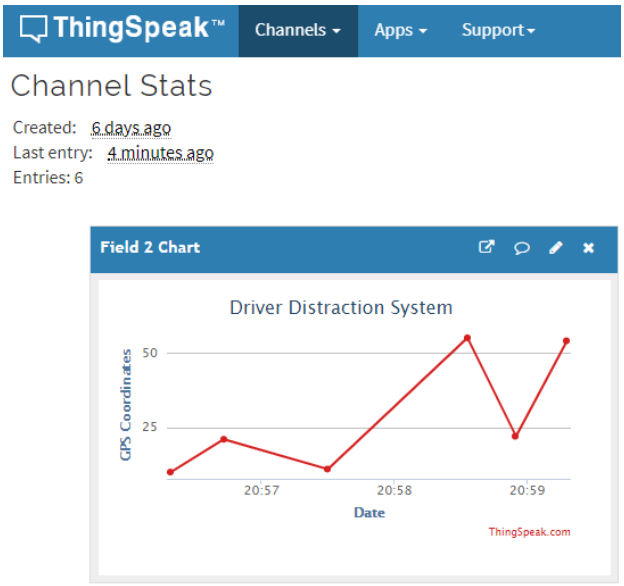

Fig. 10 ThingsSpeak Channel

Time	No	Driver ID	GPS	Offense	Image Link
2020-06-09 13:49:11 UTC	1	DRI_1	13.011768,77.62789	texting_right	https://mega.co.nz/#!7KohlQAB!Mt9Rvm6G9LtMmzXT3FgHXUSFQfTWtqNnQUGnKp8W
2020-06-09 13:49:39 UTC	2	DRI_1	13.011768,77.62789	talking_on_the_phone_right	https://mega.co.nz/#!ef4RUaTB!oG_rTwHHaisH9ylNwCrOZPVVaC7bceWByuVOrl973zU
2020-06-09 13:50:06 UTC	3	DRI_1	13.011768,77.62789	texting_left	https://mega.co.nz/#!qTgngCgB!IFrqWoa4_6sWu180fLewY3y-wZTzSAM5nXEEsVqiKdA
2020-06-09 13:50:33 UTC	4	DRI_1	13.011768,77.62789	talking_on_the_phone_left	https://mega.co.nz/#!rK5z3YKa!U6kUY4UQCrlAg2qnFKlQEoQB4wOxxDKarNnFNwEpoI0
2020-06-09 13:54:25 UTC	5	DRI_1	13.011768,77.62789	talking_to_passenger	https://mega.co.nz/#!yXwVhIIY!AMCj7YDUPi3n02aN4ji7e6-X2v-VHhpXcK_dkoyRprM
2020-06-09 13:54:57 UTC	6	DRI_1	13.011768,77.62789	texting_right	https://mega.co.nz/#!SeozTI4D!qIvSTQL0sw-7dBm4m2MHY5awh_yTv3Lmyk9RVIIgTbg

Fig. 11 Sample of data collected in the cloud

Image predictions on both the left hand and right-hand side drive are illustrated in Figures 6 and 7. The percentage of the image shows the probability of the predicted image.

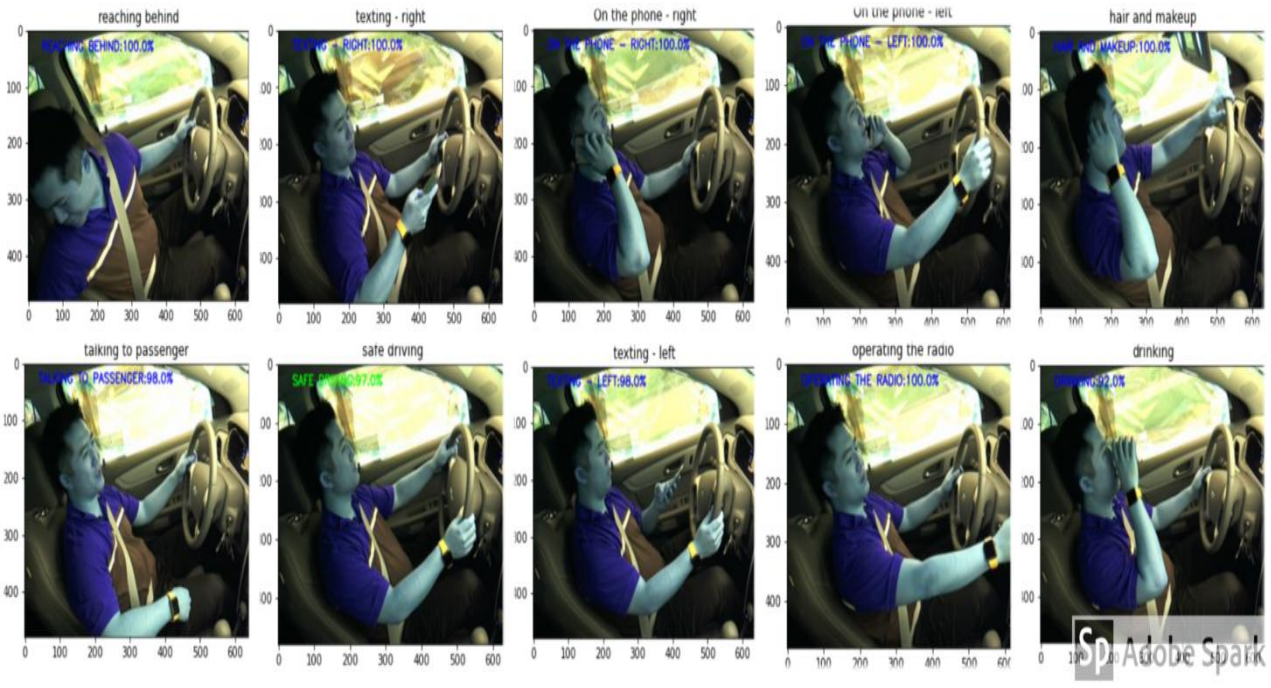

Fig. 12 LHS driving predictions

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

Fig. 13 RHS driving predictions

VI.CONCLUSION AND FUTURE ENHANCEMENTS

The model has achieved good accuracy but still needs to be optimized for better results. There is slight overfitting in the model. Transfer learning can be tried to train the model.Transfer learning can be used to speed up neural network training as either a weight initialization scheme or as a feature extraction technique. Images from a night vision camera can be included in the training set to train the model so that it operates seamlessly even during the night. Using Raspberry Pi gave us a frame rate of 1fps. When the model is used on hardware like the older version of raspberry pi which has limited hardware capabilities, it will also affect the performance of our neural network. New versions of raspberry pi which have better hardware capabilities can be used. To increase the feature detection capabilities of the neural network we can increase the depth of the model by extracting only the required features (hands, head, etc.) for computation. Other cloud services (like AWS, Google Cloud, Azure, etc.) can be used for better data analytics.

REFERENCES

[1] Y. Torres-Berru and P. Torres-Carrion, "Development of Machine Learning Model for Mobile Advanced Driver Assistance (ADA)," 2019 International Conference on Information Systems and Software Technologies (ICI2ST), Quito, Ecuador, 2019, pp. 162-167.

[2] S. Kailasam, M. Priyadarshini, M. Karthiga, K. Anithadevi and R. K. Kartheeban " An Accident Alert System for Driver Using Face Recognition," in 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), in Tamilnadu, India, 2019, pp. 1-4.

[3] M. Ali, S. Abdullah, C. S. Raizal, K. F. Rohith, and V. G. Menon, "A Novel and Efficient Real-Time Driver Fatigue and Yawn Detection-Alert System," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 687-691.

[4] M. Kumari, C. V. Hari, and P. Sankaran, "Driver Distraction Analysis Using Convolutional Neural Networks," 2018 International Conference on Data Science and Engineering (ICDSE), Kochi, 2018, pp. 1-5.

[5] S. M. Sarala, D. H. SharathYadav, and A. Ansari, "Emotionally Adaptive Driver Voice Alert System for Advanced Driver Assistance System (ADAS) Applications," 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2018, pp. 509-512.

[6] Q. Xiong, J. Lin, W. Yue, S. Liu, Y. Liu and C. Ding, "A Deep Learning Approach to Driver Distraction Detection of Using Mobile Phone," 2019 IEEE Vehicle Power and Propulsion Conference (VPPC), Hanoi, Vietnam, 2019, pp. 1-5.

[7] M. Martin, J. Popp, M. Anneken, M. Voit and R. Stiefelhagen, "Body Pose and Context Information for Driver Secondary Task Detection," 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, 2018, pp. 2015-2021

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

||Volume 9, Issue 5, May 2020||

[8] Y. Xing et al, " The Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition," in IEEE Transactions on Computational Social Systems, in vol. 5, no. 1, pp. 95-108, March 2018.

[9] Kaggle Competition Site: https://www.kaggle.com/c/state-farm-distracted-driverdetection/

[10] Ben(Yundong)Zhang. "Apply and Compare Different Classical Image Classification Method: Detect Distracted Driver," in CSS229 Stanford.edu, April 2016.