Improved Safety of Self-Driving Car using Voice Recognition through CNN

Varun Totakura
UG Scholar, CSE,
Guru Nanak Institutions Technical Campus, Hyderabad, India
totakura.varun@gmail.com
ORCID:0000-0002-5114-5205

Madhusudhana Reddy. E
Professor, CSE,
Guru Nanak Institutions Technical Campus, Hyderabad, India
e_mreddy@yahoo.com
ORCID:0000-0001-5316-5798

Devasekhar. V
Professor, CSE,
Guru Nanak Institutions Technical Campus, Hyderabad, India
hodcse2.gnitc@gniindia.org

***Abstract*— Deployment of the Artificial Intelligence (AI) in the daily life will show a new way to future, the complex task will become easy. One of such evolution of AI technology is the creation of the Self-Driving cars. But according the statistics, there are some cases where the Self-Driving cars killed people due to the failure in the prediction or hardware problem. So, to overcome those kind of situations the AI models should be accurate and also humans should take a part in making them perfect. For example, when there are some people in a self-driving car and the is car about to meet with an accident and there might be any fault in the hardware of the computer and it lost control over the car then the humans in the car should be able to instruct the car using voice commands to overcome that situation. The main focus of this paper rely on the development of Self-Driving Car, drawbacks and solution for the drawbacks. The Convolutional Neural Network (CNN) models are used to develop the models as the advancement of Computer Vision (CV) these days has grown up beyond imagination and CNN is the best algorithm to process the image data. The Self-Driving car CNN model was trained using the Asphalt-8 game data and the Voice command prediction CNN model was trained with 3 different persons (1-Kid, 1-Man, 1-Woman) voices. The accuracy of both the CNN models were 99% and were tested on the same game where they have produced the best results.**

Keywords*— *Convolutional Neural Network, Deep Learning, Voice Commands, Self-Driving, Autonomous Vehicle.

I. Introduction

According to a survey in traffic accidents, there are more than 150,000 fatalities caused each year in India. That's about 400 fatalities a day and far higher than developed auto markets like the US, which in 2016 logged about 40,000. Nearly 55% of these involve 4 wheel vehicles or buses. The idea of a self-driving car is an effort to minimize the accidents caused by careless and violent driving of 4 wheelers [1]. But there are some of the accidents that were also caused by Self-Driving cars. To overcome those kind of situations the proposed methodology will be used and it uses CV to work.

Computer Vision (CV) is one of the area of AI where the usage of image data will be high for the feature extraction from those images. And due to its accuracy and easy usage many new innovations are rising and the people are expecting more. Many papers were published in that field, especially in related to Deep Learning (DL) and CNN. The CNN algorithm is exploited in various fields to get the accurate results like image processing, video analysis, and much more [2]. Research and development in the field of CV and more precisely CV using DL lead to many discoveries and practical applications in different domains. And thus the automotive industry and the development of fully autonomous vehicles became easy by using CV. In parallel with the development of the Self-Driving cars for the human transportation, the development of various automotive platforms like delivery vehicles, robotic vehicles used in industries for the transportation of goods from one place to another is the current trend [3]. In the same way, CNN is also used in the human voice classification where the human voice signals are taken as the input and converted into images to feed to the CNN and the output will be the classification of the human voice. In keeping all the advantages of the CV by using CNN the proposed methodology is built.

This paper consists of the following topics: The methodology to build the Self-Driving car by using the Asphalt-8 game data is discussed, along with that the layers and the building of the CNN model is mentioned. And building of the Voice recognition model is mentioned. The combination of the both the model and their deployment in the game for the safe driving of the Self-Driving car is discussed. And the results of the models along with the conclusions & future work were discussed in the last section of the paper.

II. Literature Review

Eadhunath et Al. [1] has published a paper on a Self-Driving Car using Convolutional Neural Networks by testing it on the prototype model. The advantages are they have added GPS location system to monitor the route and also to navigation. They have added some other sensors to the car for the better working.

In the paper published by Brilian et Al. [2], they have used pre-trained YOLOv1 model, road lane detector and based on the observations the steering angle can be controlled automatically. Their proposed method is suitable for highways but not for city or dirt roads. They have used many filters to achieve good accuracy.

Jelena et Al. [3] have proposed a novel machine learning model named a J-Net. Their model is a light architecture which has less complexity and very less latency. Due to the problems in the other approaches mentioned in their paper, they have created J-Net. For 10 successful laps it's latency is around 24ms with 44 frames per second. Due to the usage of this model the autonomous vehicular model can be embedded with less cost, low-power hardware and size.

Truong-Dong et Al. [4] mentioned their paper by proposing a model that has two phases, one is to predict the angle of steering and the other is for the road lane detection, traffic signal identification and navigation. They have mentioned that their method was tested on a RC car using Raspberry Pi and various other sensors. They have even mentioned about the study of their proposed model that has

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©2020 IEEE

installed on the RC car on the various experimental tracks and various traffic signals placed on the track.

Wael Farag [5] has published a paper where the concept of the paper is helpful to the autonomous vehicle or self-driving car. The main theme of the paper is building an CNN based classifier called "WAF-LeNet". The concept of the paper is the identification of the traffic signal using the proposed CNN model. German Traffic dataset was used for the training of the model and the model has produced over 96.5% of accuracy on the test dataset and about 100% accuracy on the robust data.

Vishal et Al. [6] has published a chapter in a book named as Convolutional Neural Networks for Raw Speech recognition. They have covered the methodology for the conversion of the voice to images and recognition of the voice through the converted images by using the word image database. They have used Mel-Frequency Cepstral Coefficients (MFCC) which is more useful for the identical and is most commonly using for the speech recognition or mood recognition. The musical notes generation or identifications also done using the MFCC images produced out of the voice signals.

Sanguk et Al. [7] has mentioned about the theoretical implications as well as practical implications about the design of the autonomous vehicle voice agents (AVVA). Based on the gender of the people in the self-driving car the response to the conversation made by the virtual assistant will effect. They have mentioned about the stereotypical expectation of the social role (informative male AVVA and social female AVVA).

III. Problem Statement

The main problem is that the self-driven cars that are in the present world are not 100% accurate in some of the times. The accuracy will depend on various factors like road condition, GPS signals, weather conditions, and many more. A small error produced by the model will lead to the terrible effects to the people present in the car. To overcome this situation, the person who sits in the vehicle should be always alert and should know how to drive that vehicle. But there might be some passengers who might not able to drive that particular type of vehicle, then there will be a problem and it can lead to very uncertain conditions. The present self-driven cars are coming with no driving compartment and only a screen is present inside the vehicle to give the instructions and navigations to the passengers. For instance, let us suppose that the self-driving car is a taxi and there is no driver compartment and if the machine fails due to some technical problem then the people inside the car can get into any unpleasant situation.

To overcome all the above problems, we propose a methodology which can be used effectively in all kind of situations and also can be installed in all the self-driven cars for the passenger safety. We represent the solution for the above mentioned problems in proposed methodology phase and the results or other findings are discussed in the result and conclusion phase.

IV. Proposed Methodology

The Convolutional Neural Networks are the game changers to obtain the best solutions. They have wide usage in providing the solutions to the problems related to computer vision. In the same way, by using the CNN the solution for the above mentioned problems is mentioned below. Generally, a CNN will consists of multiple layers and multiple phases as in shown in the Figure - 1. The input consists of image where the pooling is applied to that image where a particular area in that image will be taken and the max of the pixel values will be updated in that image for the next pooling. In the same way, multiple image will undergo continuous pooling and maxing effects in various convolutional layers and finally the weights which are obtained after the process will be stored. After certain iterations or epochs, the CNN will learn and stores the weights in a file and using those weights we can predict the test data. We have used python as the platform to develop the CNN model for providing the solution for the problems mentioned in the problem statement phase.

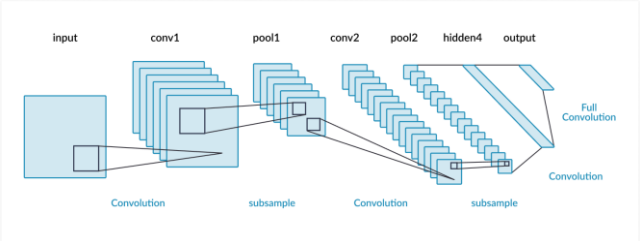

Figure – 1: Convolutional Neural Netwok

In our proposed methodology, there are two CNN models. One model is for the self-driving car and the other is for the voice recognition. By combining those two models we can get the optimal results. The system architecture of the proposed methodology is as shown in Figure – 2.

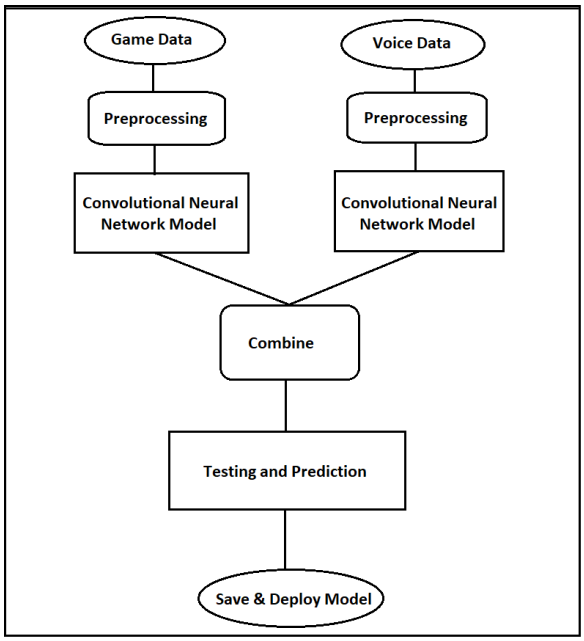

Figure – 2: System Architecture

The input game data is as shown in the Figure – 3. Along with the set of those kind of images, the keystrokes for every move are combined and stored as the multidimensional array as [Image, Key Stroke]. Before storing the images, those images will undergo into the pre-processing segment.

Pre-Processing the Data:

1. Primary step is to take all the images at once and put them in a folder.
2. A blank array is defined to store the images and respective classes which are multi-dimensional.
3. Read the images using OpenCV, which converts the images into numerical arrays.

Figure – 3: Asphalt - 8 Game data.

4. A NumPy file will be saved for every set of 100 images by combining the [image_array, class].
5. Now, NumPy file is used to read all the images which exists along with their respective class.
6. The images are in the order of audio files. This may impact the efficiency of the model, hence the NumPy file is shuffled to make the confused over an order, but not original data.
7. This NumPy file which is shuffled is saved as another file. The same process is conducted for the train dataset and test dataset too.

The input for the voice data is generated using the wav file. From the wav file the voice signal can be generated as shown in the Figure – 4. By the audio signals we can generate two types of images. One is by generating the specgrams as shown in the Figure - 6 and the other is generating the MFCC related image as shown in the Figure – 5.

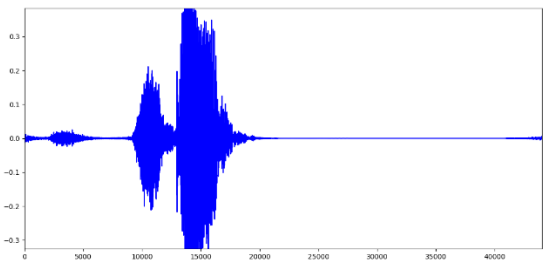

Figure – 4: Sound Wave

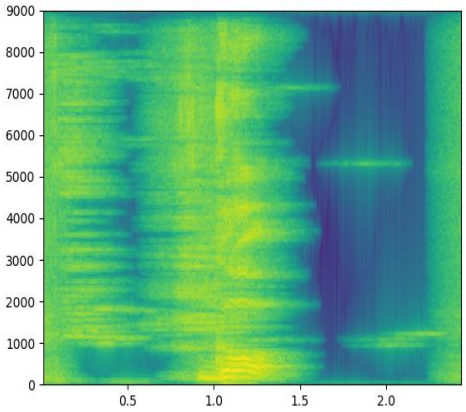

Figure – 5: Voice to Specgram Image

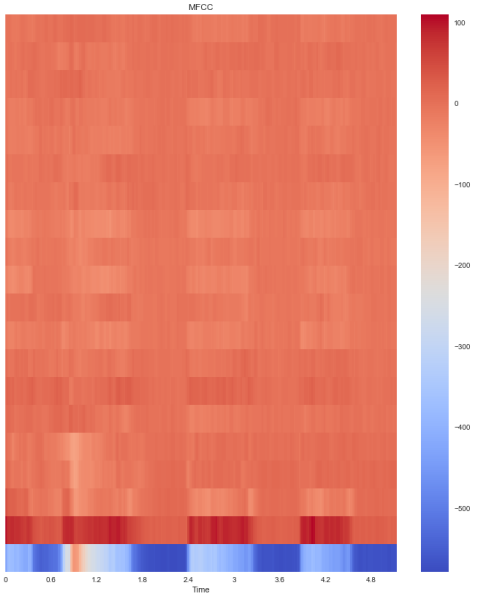

Figure – 6: Voice to MFCC Image

Any of the two types of Voice signal converted images can be used to train the Voice recognition CNN model to get a decent accuracy. Here, we use Specgrams images as our input to the Voice recognition model as the variation between the images for every class is much greater than the MFCC images.

After saving all the images in the specified format the CNN model is trained using those images as the input. The specifications and the details of the CNN models for both the models i.e., Self-Driving car model and the Voice recognition model is as mentioned below.

Building the Convolutional Neural Network for the Self-Driving car:

1. The Numpy file which is created above is imported and size adjustment is done from multidimensional array to single dimensional array.
2. A sequential network of 5 layers is built with 3 layers as the hidden layers.
3. First layer is known as the flatten layer, this layer used to resize the data.

4. Second layer, which is first hidden layer, is with neural network nodes and rectified linear activation function.
5. Third layer, which is second hidden layer, is built with 512 neural network nodes and relu activation function.
6. Fourth layer (third hidden layer) with 128 nodes neural network and relu as activation function.
7. The last layer is the layer with the same number of nodes as the classes that are available and activation function is soft max (normalized exponential function to normalizes k real numbers into a probability distribution consisting of K probabilities).
8. The model needs to be compiled using Adam Optimizer.
9. The loss in the network can be found with sparse categorical cross entropy.
10. The constructed model is trained with 15 epochs as a result it produces 99.7% accuracy.

Building the Convolutional Neural Network for the Voice Recognition model:

- The layers of the CNN are similar to the layers in the Self-driving car.
- But the constructed Voice recognition model is trained with 100 epochs as a result it produces 100% accuracy.

Testing of the CNN:

After the training of both the models they are saved with the (.model) file extension. The testing can be done using those files. The input to the saved files are processed in prior to avoid all the errors and after that the predictions from those files are used. The basic architecture of the testing phase is as shown in the Figure – 7. Both models are combined and are given with the same input but the output from those models are used as per the requirement. In the Figure – 8, the predictions of the given input images obtained using the trained self-driving car model are displayed. In the Figure – 9, the predictions of the voice recognition model along with the self-driving car model is shown. As per the voice commands given the self-driving car model is activated or the voice recognition model is activated.

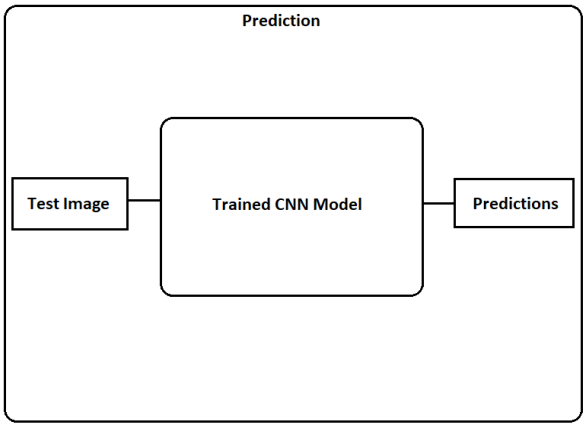

Figure – 7: Testing Phase Architecture

```
Moves:  [0.0, 0.0, 1.0, 0.0] Predictions:  [0. 0. 1. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [0.0, 0.0, 1.0, 0.0] Predictions:  [0. 0. 1. 0.]
Moves:  [0.0, 0.0, 1.0, 0.0] Predictions:  [0. 0. 1. 0.]
Moves:  [0.0, 0.0, 1.0, 0.0] Predictions:  [0. 0. 1. 0.]
Moves:  [0.0, 0.0, 0.0, 1.0] Predictions:  [0. 0. 0. 1.]
Moves:  [0.0, 1.0, 0.0, 0.0] Predictions:  [0. 1. 0. 0.]
Moves:  [0.0, 1.0, 0.0, 0.0] Predictions:  [0. 1. 0. 0.]
Moves:  [0.0, 1.0, 0.0, 0.0] Predictions:  [0. 1. 0. 0.]
```

Figure – 8: Predicions of Self-Driving Car

```
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Moves:  [1.0, 0.0, 0.0, 0.0] Predictions:  [1. 0. 0. 0.]
Set minimum energy threshold to 1666.975392341361
Say Something
Time over, Thanks
```

Figure – 9:Predictions of Voice Recognition Model

The obtained results are in the one hot format that means in an array with the zero values only the value pertained to the specific class is marked as 1. It can be explained as shown in the Figure – 10 which is the predictions of the Self-Driving car model. If the predicted label has the first value of the array as 1 then the "straight" function is activated. In the same way all the other functions are activated according to the obtained predicted labels.

```
if predicted_label == [1.0, 0.0, 0.0, 0.0]:
    straight()
elif predicted_label == [0.0, 1.0, 0.0, 0.0]:
    reverse()
elif predicted_label == [0.0, 0.0, 1.0, 0.0]:
    left()
elif predicted_label == [0.0, 0.0, 0.0, 1.0]:
    right()
```

Figure – 10: Using the Predicted label obtained from the Self-Driving Car Model

The Figure – 11 will show the output or the prediction which are obtained using the Voice Recognition CNN model, where the predicted label is labelled as text as it is the voice of the human being.

```
if text in ['go', 'straight', 'start', 'up']:
    predicted_label = [1.0, 0.0, 0.0, 0.0]
elif text in ['stop', 'reverse', 'back', 'halt']:
    predicted_label = [0.0, 1.0, 0.0, 0.0]
elif text in ['left', 'turn left']:
    predicted_label = [0.0, 0.0, 1.0, 0.0]
elif text in ['right', 'turn right']:
    predicted_label = [0.0, 0.0, 0.0, 1.0]
```

Figure – 11: Using the Predicted label obtained from the Voice Recognition Model

So, in this paper we propose a voice over vehicular control model that will help the passengers in that vehicle to instruct the vehicle what to do in those kind of situations. The main advantage of this model is that the passengers need not to have any driving experience, they can use the voice commands like "Go Straight", "Turn Left", "Turn Right", "Stop", and "Reverse" to control the vehicle in that kind of situations.

V. Results And Discussion

The Self-Driving Car CNN model and Voice recognition CNN model has produced very good results. The models are now ready to be tested which states the respective label of the image of classified with the existing label in the training set else the image is categorized as unclassified. After the test phase the predicted is checked and accuracy is calculated. In the Figure – 12, the specifications or details in the layers of the Convolutional Neural Network used for the Self-Driving car as mentioned in the above proposed methodology phase which is used for the training of the Self-Driving car model is shown, there are three columns in the details, the first column is the type of the layer used, second column is the shape of the input given to the particular layer used is displayed and in the third layer the parameters or the number of neurons used in each layer is shown. Along with the specification or details the training accuracy and the time taken for the training process is also displayed.

The Figure – 13, which give us the information about the accuracy of the model in each epoch or iteration along with the loss per every epoch or iteration is plotted in the graph. The X-axis of the graph is the Epochs and the Y-axis is the percentage of the result. As shown in the graph, the Self-Driving has an accuracy of 99.7% over the 15 epoch or iterations and the loss gradually decreases from 70% to almost 0%. As shown in the legend present in the graph, the blue line will give the information about the accuracy and orange will give the information about the loss for every epoch.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 99, 149, 32)	416
max_pooling2d (MaxPooling2D)	(None, 49, 74, 32)	0
conv2d_1 (Conv2D)	(None, 47, 72, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 23, 36, 64)	0
conv2d_2 (Conv2D)	(None, 21, 34, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 10, 17, 64)	0
dropout (Dropout)	(None, 10, 17, 64)	0
flatten (Flatten)	(None, 10880)	0
dense (Dense)	(None, 512)	5571072
dense_1 (Dense)	(None, 128)	65664
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 4)	516

Total params: 5,709,604
Trainable params: 5,709,604
Non-trainable params: 0

9584/9584 [==============================] - 235s 24ms/step
Test accuracy: 0.9974958263772955
Total Time: 3.45 Hours
Accuracy: 99.7%

Figure – 12: The Self-Drving Car CNN Model and Training Accuracy

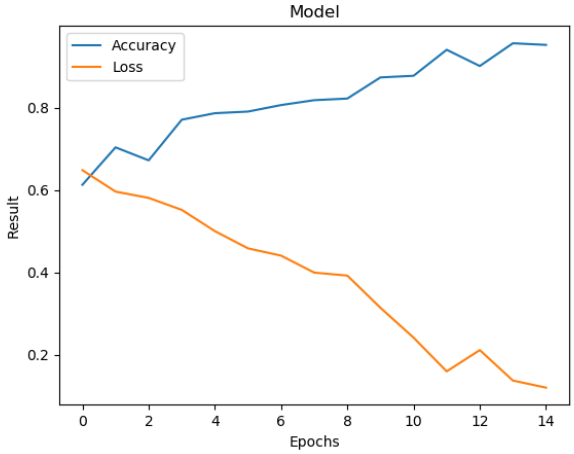

Figure – 13: The Accracy and Loss graph of Self-Driving Car CNN Model

In the Figure – 14, the specifications or details in the layers of the Convolutional Neural Network used in the Voice Recognition Model as mentioned in the above proposed methodology phase is also similar to the specifications or details of the layers used for the Self-Driving car CNN Model which is used for the training of the model is shown. Along with the specification or details the training accuracy and the time taken for the training process is also displayed.

The Figure – 15, which give us the information about the accuracy of the model in each epoch or iteration along with the loss per every epoch or iteration is plotted in the graph. The X-axis of the graph is the Epochs and the Y-axis is the percentage of the result. As shown in the graph, the Self-Driving has an accuracy of 100% over the 100 epochs or iterations which was increased gradually from 30% to 100% and the loss gradually decreases to almost 0%. But, the plot seems to be uneven as the training data is much lesser and the images are likely to be similar. So, it took lot of time for the model to reach a decent accuracy.

Layer (type)	Output Shape	Param #
conv2d_33 (Conv2D)	(None, 59, 79, 32)	160
max_pooling2d_33 (MaxPooling	(None, 29, 39, 32)	0
conv2d_34 (Conv2D)	(None, 27, 37, 64)	18496
max_pooling2d_34 (MaxPooling	(None, 13, 18, 64)	0
conv2d_35 (Conv2D)	(None, 11, 16, 64)	36928
max_pooling2d_35 (MaxPooling	(None, 5, 8, 64)	0
dropout_11 (Dropout)	(None, 5, 8, 64)	0
flatten_11 (Flatten)	(None, 2560)	0
dense_68 (Dense)	(None, 512)	1311232
dense_69 (Dense)	(None, 128)	65664
dense_70 (Dense)	(None, 128)	16512
dense_71 (Dense)	(None, 4)	516

Total params: 1,449,508
Trainable params: 1,449,508
Non-trainable params: 0

8/8 [==============================] - 0s 21ms/sample - loss: 0.1311 - acc: 1.0000
Test accuracy: 100.0 %
Predictions : D (Turn Right)
Actual : D (Turn Right)

Figure – 14: The Voice Recognition CNN Model and Training Accuracy

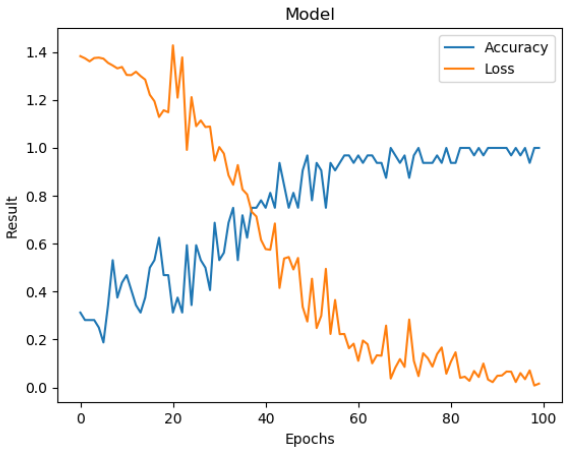

Figure – 15: The Accracy and Loss graph of Voice Recognition CNN Model

VI. Conclusion And Future Work

The research paper will give the keen insight and the advantages of using the self-driving car in the real world along with the voice recognition model. Using the proposed methodology, it can be concluded that all the problems caused by the self-driving car in all the cases which were mentioned in the problem statement are solved.

In future, there is a chance of creating more secured self-driving autonomous vehicles which can show the path to the world. The safe and secured autonomous vehicles can be used in all the sectors widely as the usage is very easy. So, the safety and security in using the self-driving vehicles should be increased and make them to achieve 100% success in all aspects.

References

[1] Eadhunath V, Amir Suhail, Jayant Waghmare, Rishab Mishra, Prof. K. U. Jadhav, "Self-Driving Car using Convolutional Neural Network and Road Lane Detector", International Journal of Engineering Research & Technology (May - 2019), Volume – 8, Issue – 5, pp. 951-954.

[2] Brilian Tafjira Nugraha, Shun-Feng Su, Fahmizal, "Towards Self-Driving Car using Convolutional Neural Network and Road Lane Detector", International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), October 23, 2017, Jakarta, Indonesia, pp. 65-70.

[3] Jelena Kocic , Nenad Jovic˘ic and Vujo Drndarevic, "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms", Sensors 2019, 19, 2064.

[4] Truong-Dong Do, Minh-Thien Duong, Quoc-Vu Dang and My-Ha Le, "Real-Time Self-Driving Car Navigation Using Deep Neural Network", 2018 4th International Conference on Green Technology and Sustainable Development (GTSD), pp. 7-13.

[5] Wael Farag, "Recognition of traffic signs by convolutional neural nets for self-driving vehicles", International Journal of Knowledge-based and Intelligent Engineering Systems, Vol. 22, (2018), pp. 205–214.

[6] Vishal Passricha and Rajesh Kumar Aggarwal, "Convolutional Neural Networks for Raw Speech Recognition", From Natural to Artificial Intelligence, IntechOpen, Chapter 2, 2018, 10.5772/intechopen.80026

[7] Sanguk Lee, Rabindra Ratan and Taiwoo Park, "The Voice Makes the Car: Enhancing Autonomous Vehicle Perceptions and Adoption Intention through Voice Agent Gender and Style", Multimodal Technol. Interact. 2019, 3, 20.