**AUTOMATIC BRAKING AND SPEED CONTROL SYSTEM USING DEEP NEURAL NETWORKS**

***Mini Project Report Submitted by***

**Chethan Eastan Saldanha**

**(4NM19ME025) (4NM19ME036)**

**Under The Guidance of**

**Mr. Adarsh Rai**

Assistant Professor

***in partial fulfillment of the requirements for the award of the Degree of***

***Bachelor of Engineering (Mechanical)***

***from***

***Visvesvaraya Technological University, Belgaum***



**DEPARTMENT OF MECHANICAL ENGINEERING**

(AICTE approved, NBA Accredited, ISO 9001:2008 Certified)

**ACKNOWLEDGEMENT**

We take this opportunity to express my deep sense of gratitude to my guide Mr. Adarsh Rai, Asst Professor, Department of Mechanical Engineering, NMAMIT, Nitte for his whole-hearted support and valuable guidance he has given us during the Project part 1.

We convey our sincere regards to **Dr. Shrinivasa Pai**, Professor & Head, Department of Mechanical Engineering, NMAMIT, Nitte for encouraging us during the project tenure.

Our grateful thanks to **Dr. Niranjan N. Chiplunkar,** Principal, NMAMIT, Nitte for his inspiration and support while carrying out the project work.

We cordially acknowledge and give due credit to our friends and family for the constant support they have extended to we and the faith they have shown in us during the tenure of this mini project and finally to God who made all the things possible for us till the end.

Chethan (4NM19ME025) and Eastan (4NM19ME036)

**Contents**

|  |  |  |
| --- | --- | --- |
|  | | Page No. |
| **Acknowledgement** | | 2 |
|  | |  |
| **Chapter 1** | **Introduction** | 4 |
| **Chapter 2** | **Literature Review** | 7 |
| **Chapter 3** | **Problem definition** | 11 |
| **Chapter 4** | **Objectives** | 12 |
| **Chapter 5** | **Methodology of proposed work** | 13 |
| **Chapter 6** | **Work Done** | 20 |
| **Chapter 7** | **Results and discussions** | 30 |
|  | **References** | 32 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

This project gives a brief introduction of Automatic Braking and Speed Control System based on Convolution Deep Neural Network and how to effectively use it to control and regulate the speed of a vehicle effectively. At the end of the chapter objectives of the project, methodology and the organization of the report have been discussed.

**1.2 Automatic Braking and Speed Control System (ABSCS)**

Safety is a necessary part of man’s life. Due to the accident cases reported daily on the major roads in all parts of the countries, more attention is needed for research in the designing an efficient car driving aiding system. According to the report of NCRB of India 3,54,796 cases of road accidents during 2020 in which 1,33,201 people died and 3,35,201 were injured and more than 60% of road accidents were caused due to over-speeding, accounting for 75,333 deaths and 2,09,736 injuries. In many road accident cases, a major cause of the accident is the driver distraction and failure to react in time. Advanced system of auxiliary functions has been developed to help avoid such accident and minimize the effects of collision in effect. This was achieved by reducing the total stopping distance through works done by researchers in the past. Lately some of the works were used in many car brake system developments deploying electronic brake control system which has led to significant safety in driving. We have come up with an Automatic Braking Systems which uses more sophisticated deep learning methods like convolutional neural network for the hurdle detection in the path of the vehicle and correspondingly controlling the speed of the vehicle.

**1.3 CONVOLUTION NEURAL NETWORK**

Neural networks are a subset of machine learning, and they are at the heart of deep learning algorithms. They are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have two main types of layers, which are:

1. Convolutional layer
2. Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

**1.4 OBJECTIVE**

The purpose of this project is to develop an automated control system based on deep neural networks that would maintain a safe driving distance from obstacles while driving which prevents high speed collisions or avoids collision. Since high speed crashes are more likely to be fatal than low speed collisions, automatic braking systems can save lives and reduce the amount of property damage that occurs during an accident. It focuses on passenger safety and providing stress-free driving to the driver as well as the passengers while minimizing property damage of the passengers.

**1.5 METHODOLOGY**

Automatic braking and speed control System (ABSC) basically controls the speed of the vehicle by continuously feeding the driving atmosphere to the pre-trained deep neural network as digital image captured by the camera sensor​.

The Neural Network predicts the desired speed for that instant and the controller unit will send signal to the actuator unit to control the vehicle speed based on the extent with which the vehicle is going.

**1.6 ORGANISATION OF REPORT**

Chapter 1 gives the brief introduction to Deep Neural network, Automatic Braking and Speed Control System and objectives of the project.

Chapter 2 explains the background theory and the literature review related to the Automatic Braking and Speed Control System.

Chapter 3 describes the main problem definition of this mini project.

Chapter 4 Provides an overview of the project's objectives.

Chapter 5 provides a summary of the methodology we incorporated into the project.

Chapter 6 runs through all the work done by us beginning from data collection to training the code.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

This chapter provides details of observations from the literature review that has been carried out.

**Rizianiza1 et all,** Institute of Technology, Balikpapan, Indonesia have used Fuzzy Logic method. This research focuses on design of an autonomous car control system that can reduce the rate of traffic accident. Car prototype is designed by using Fuzzy Logic Control. Fuzzy logic is as main component of artificial intelligence has significantly influenced the design-controlled system. Fuzzy logic system is integrated with the Arduino Mega 2560 microcontroller. The Fuzzy logic used Mamdani method and 28 rule bases. Fuzzy logic has a variable input and output variables. Input variables used in this research is the distance between car and obstacle; the speed of car prototype and the output variable is the brake angle.

**Hyunmin Chae et all** from Hanyang University with his teammates has proposed a new autonomous braking system in the year 2017 based on deep reinforcement learning. The proposed autonomous braking system automatically decides whether to apply the brake at each time step when confronting the risk of collision using the information on the obstacle obtained by the sensors. The problem of designing brake control is formulated as searching for the optimal policy in Markov decision process (MDP) model where the state is given by the relative position of the obstacle and the vehicle's speed, and the action space is defined as whether brake is stepped or not.

**Mariusz Bojarski et all** in the year 2016 proposed end-to-end deep learning model for self-driving car. The article describes the use of convolutional neural networks (CNNs) to map the raw pixels from a front-facing camera to the steering commands for a self-driving car. This powerful end-to-end approach means that with minimum training data from humans, the system learns to steer, with or without lane markings, on both local roads and highways. The system can also operate in areas with unclear visual guidance such as parking lots or unpaved roads.

**Jiri David** from the Department of Mechanical and Electrical Engineering, SKODA AUTO University has used Adaptive Cruise Control with Deep learning for the efficient brake control in the year 2021. The article describes the basic functionality of Adaptive cruise control and creates a mathematical model of braking, which is one of the basic functions of adaptive cruise control. The advantage of this approach is the original use of neural networks, which refines the determination of the deceleration value of the vehicle in front of a static or dynamic obstacle, while including a number of influences that affect the braking process and thus increase driving safety. The NN model predicts the deceleration value of the vehicle based on them and provide this value to the ACC control unit. The article represents a comprehensive approach, which is focused on ACC.

**Deepanshu Prashar et all** Research Scholar, Automobile Department, Gulzar Group of Institute, Ludhiana, Punjab have implemented a technology where they have used deep neural network or Convolutional Neural network, which can classify signs or signals present in the image into different categories. They introduced a real time traffic sign/light detecting system using Open CV and CNN. With this approach, they could able to recognize different kinds of traffic lights/signs and with their output, the accuracy of the braking system has increased. This approach is mainly based on implementation of the Automatic brake actuating system and Automatic throttle controlling system during detecting the signals. They have used a camera vision sensor for detecting the traffic lights/signs and they have also combined the Radar sensors for detecting the other obstacles like pedestrians as well as other vehicles and their distance.

**Salameh A. Sawalha**, faulty of Engineering Technology, Al- Balqa Applied University- Jordan has proposed special braking control system. It is a system that aims to control vehicle's speed in accordance with the street and road speed limit. The system finds out the whereabouts of the vehicle then it identifies the allowed speed for that particular place. After this, it starts decreasing the speed gradually until it reaches the desirable limit. It uses GPS to fin the location of the vehicle (longitude and latitude). If it has been found that the vehicle's speed exceeds the limit, a signal will be sent to the brake system to reduce the speed until it drops to the limits.

**Pushkar P. Bhat,** Department of Mechanical Engineering, Faculty of Technology & Engineering, M. S. University, Baroda, Gujarat along with their subordinates have introduced A.O.C.S system, which stands for Automatic Over Speed Control System. It basically controls the speed of the vehicle continuously checking it through sensors which sends the signal to the input of microcontroller. Whenever the speed is increased above the predefined critical speed, the microcontroller through its logic circuits sends the signals to the actuators via its output pins. The actuators apply necessary force on the brakes and reduce the speed.

**Dhanya K. R.** has proposed a technology of an advance automatic braking system with sensor fusion concept. In this they used the properties of both capacitive & ultrasonic sensor for detecting the obstacle & also for calculating the distance between the vehicle & the obstacle. This distance measurement is used to control automatic braking system for safety application. In this system they used the 32-bit microcontroller with ARM processor (LPC2138) as the brain of this system for controlling process. The programming is done by using c-language. The additional feature included in the system are automatic retarding & automatic horn disabling in restricted area, this is done through RF signal communication. The RF transmitter is placed in restricted area, where the speed is limited & horn is restricted.

**S. P. Bhumkar** presents a system of about accidents avoidance & detection on highways. This system is about advance technology in cars for making it more intelligent & interactive for avoiding accidents on roads. ARM7 is using for making this system more efficient, reliable & effective. In this system, they have described real-time online safety prototypes that control the vehicle speed under driver fatigue. The purpose of this system is to avoid accidents. The main component of this system consists of number of real time sensor like gas, eye blink, alcohol, fuel impact sensor & a software is interface with GPS & Google maps APIs for location. Through this research work, they have proposed an intelligent car system for accident prevention & making the world a much better & safe place to live.

**2.2 OBSERVATIONS FROM LITERATURE SURVEY**

* The CNN is a widely used in developing automatic braking or speed control system.
* Emphasis on driver safety must be given while developing automatic braking system.
* A large collection of data in every possible situation is required to train the system.
* Optical sensors are widely used to detect/sense the objects or obstacles.

**CHAPTER 3**

**PROBLEM DEFINITION**

Road accident is most unwanted thing to happen to a road user, though they happen quite often. The most unfortunate thing is that people don't learn from their mistakes on road. Most of the road users are quite well aware of the general rules and safety measures while using roads but it is only the laxity on part of road users, which cause accidents and crashes. Main cause of accidents and crashes are due to human errors. Various national and international researchers have found some of the most common behavior of Road drivers, which leads to accidents. These include,

1. Over Speeding

Most of the fatal accidents occur due to over speeding. It is a natural psyche of humans to excel. But when we are sharing the road with other users we will always remain behind some or another vehicle. Increase in speed multiplies the risk of accident and severity of injury during accident. Faster vehicles are more prone to accident than the slower one and the severity of accident will also be more in case of faster the severity of accident will also be more in case of faster vehicles.

1. Distraction to Driver:

Though distraction while driving could be minor but it can cause major accidents. Distractions could be outside or inside the vehicle. The major distraction now a days is talking on mobile phone while driving. Act of talking on phone occupies major portion of brain and the smaller part handles the driving skills. This division of brain hampers reaction time and ability of judgement. Adjusting mirrors while driving, Stereo/Radio in vehicle, Animals on the road, Banners and billboards are the reasons for this.

**CHAPTER 4**

**OBJECTIVES**

The purpose of this project is to develop an automated control system based on deep neural networks that would maintain a safe driving distance from obstacles while driving which prevents high speed collisions or avoids collision. Since high speed crashes are more likely to be fatal than low speed collisions, automatic braking systems can save lives and reduce the amount of property damage that occurs during an accident. It focuses on passenger safety and providing stress-free driving to the driver as well as the passengers while minimizing property damage of the passengers. The model building focused on,

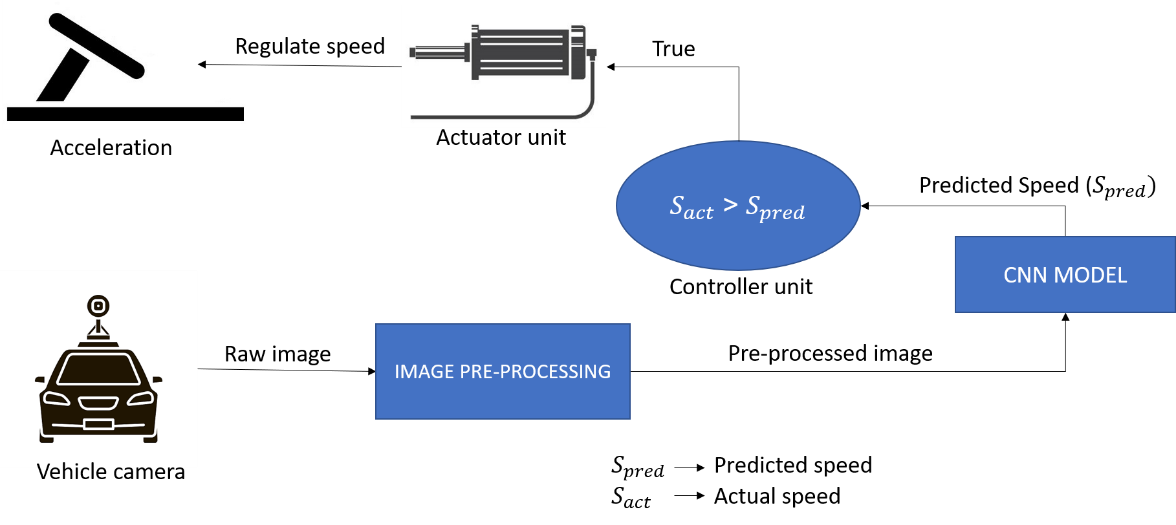
1. Appropriate placement of camera sensor on the vehicle
2. Desired amount of data required for the training as well as for testing
3. Selection of image preprocessing steps
4. Selection of right number of layers and appropriate activation function, loss function and optimization algorithm
5. Optimal tuning of model hyper-parameters for optimizing the performance of the model
6. Other methods required for the improvement of model performance

**CHAPTER 5**

**METHODOLOGY OF PROPOSED WORK**

* 1. **METHODOLOGY**

Automatic braking and speed control System (ABSC) basically controls the speed of the vehicle by continuously feeding the driving atmosphere to the pre-trained deep neural network as digital image captured by the camera sensor​.



*Fig. 5.1 System Architecture*

The Neural Network predicts the desired speed for that instant and the controller unit will send signal to the actuator unit to control the vehicle speed based on the extent with which the vehicle is going. If the current speed of the vehicle is more than the predicted speed, the vehicle’s embedded unit automatically alerts the driver to reduce the speed according to the prediction and waits for the user response. If it doesn’t get any input from the user, the controller will calculate the change in speed between previous and current instants and correspondingly the acceleration and then the controller will send signals the actuator to move the accelerate peddle to attain speed predicted by the Neural Network. The proposed work is likely to control the speed of vehicle and the process consist of following steps,

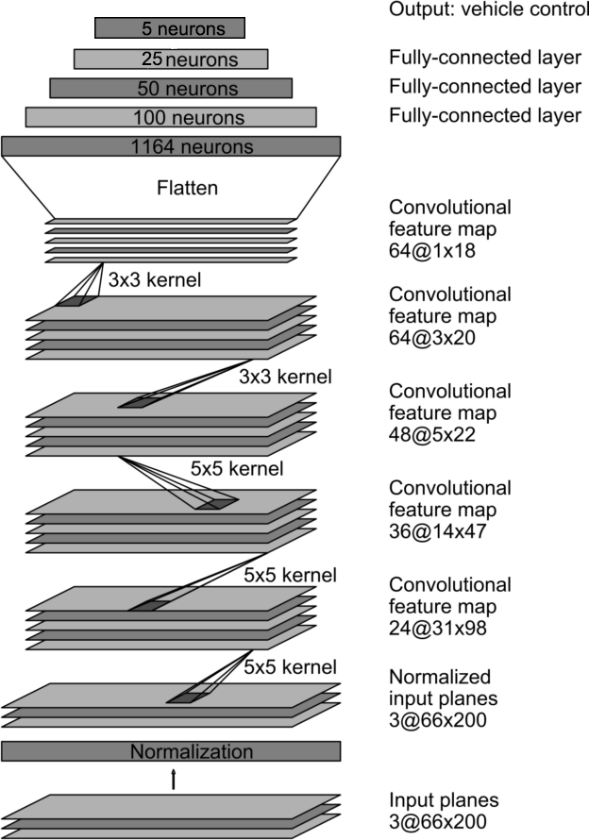
1. Capture the driving atmosphere / Collect the environment data using Camera sensor

2. Pre-process the collected image

3. Feed pre-processed image to the AI model

4. Maintain / reduce / increase the speed as predicted by the model

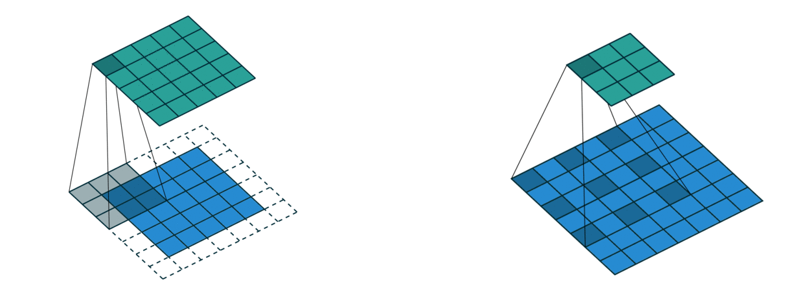
* 1. **MODEL ARCHITECTURE**

****A CNN typically has two layers: a convolutional layer and a fully connected layer. The network architecture consists of 9 layers, including a normalization layer, 5 convolutional layers, and 4 fully connected layers. The network has about 27 million connections and 250 thousand parameters. The convolutional layers are designed to perform feature extraction, and are chosen empirically through a series of experiments that vary layer configurations. We then use strided convolutions in the first three convolutional layers with a 2×2 stride and a 5×5 kernel, and a non-strided convolution with a 3×3 kernel size in the final two convolutional layers. Each layer is introduced with non-linearity with ReLU activation function. The output layer consists of log SoftMax function for multiclass classification. The output will be any one of the ranges between 0-5, 5-15, 15-25, 25- 35, more than 35.

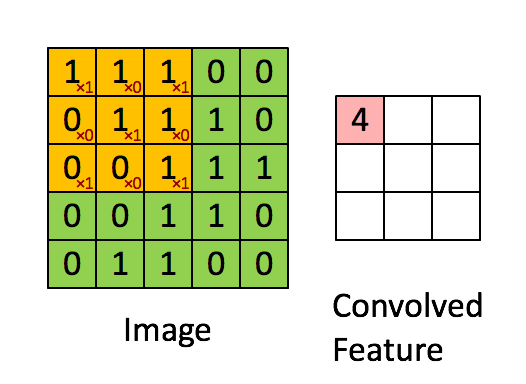
*Fig. 5.2 Model Architecture*

**5.2 KEY ELEMENTS INVOLVED IN CNN**

**5.2.1 CONVOLUTION**

The human brain processes a huge amount of information the second we see an image. Each neuron works in its own receptive field and is connected to other neurons in a way that they cover the entire visual field. Just as each neuron responds to stimuli only in the restricted region, CNN processes data only in its receptive field as well. The layers are arranged in such a way so that they detect simpler patterns first (lines, curves, etc.) and more complex patterns (faces, objects, etc.) further along. By using a CNN, one can enable sight to computers.

*Fig. 5.2.1.1 Kernel Fig. 5.2.1.2 Kernel*

It carries the main portion of the network’s computational load. This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. If the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels. During the forward pass, the kernel slides across the height and width of the image producing the image representation of that receptive region. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride. If the size of the input image is , K is number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula,

*Fig. 5.2.1.3 Convolution Process*

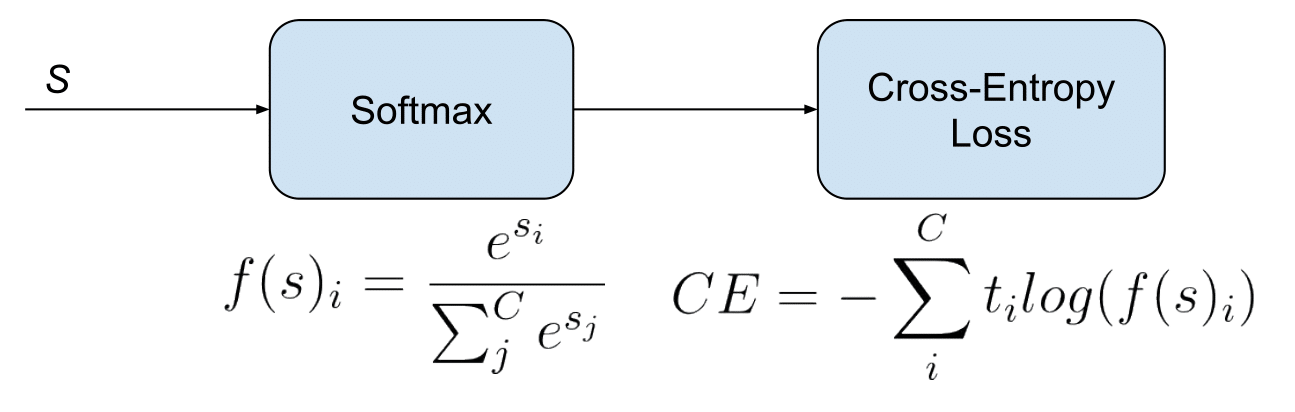
**5.2.2. LOSS FUNCTION**

The loss function in a neural network quantifies the difference between the expected outcome and the outcome produced by the machine learning model. The loss function is used to derive the gradients which are used to update the weights. The average over all losses constitutes the cost. Some of the loss functions include,

• Mean Squared Error (MSE)

• Binary Cross entropy (BCE)

• Categorical Cross entropy (CC)

 • Sparse Categorical Cross entropy (SCC)

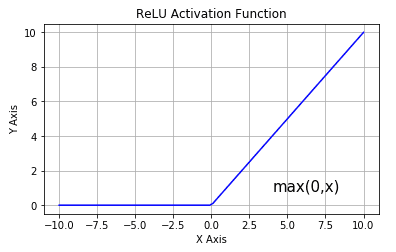
*Fig. 5.2.2.1 Cross Entropy Loss Function*

Cross-entropy-based loss functions are commonly used in classification scenarios where as MSE is used for regression scenarios. Cross entropy is a measure of the difference between two probability distributions. In a machine learning maximum likelihood estimation is used to calculate the difference

between the probability distribution produced by the data generating process (the expected outcome) and the distribution represented by our model of that process. The resulting difference produced is called the loss. It increases exponentially as the prediction diverges from the actual outcome. If the actual outcome is 1, the model should produce a probability estimate that is as close as possible to 1 to reduce the loss as much as possible.

**5.2.3 ACTIVATION FUNCTIONS**

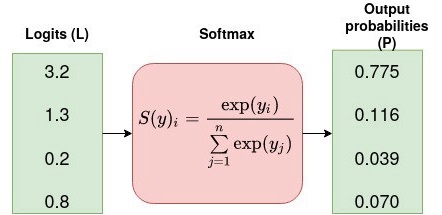
1. **ReLU**

The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function, In other words, the activation is simply threshold at zero. In comparison to other activations functions like sigmoid and tanh, ReLU is more reliable and accelerates the convergence by six times. Unfortunately, a con is that ReLU can be fragile during training. A large gradient flowing through it can update it in such a way that the neuron will never get further updated. However, we can work with this by setting a proper learning rate.

*Fig. 5.2.3.1 ReLU Activation Function*

1. **Softmax**

The SoftMax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, SoftMax is used as for multi-class classification problems where class membership is required on more than two class labels.



*Fig. 5.2.3.2 Softmax Function*

The other activation functions include Sigmoid, tanH, ELU, etc.

**5.2.4. OPTIMIZATION ALGORITHM**

An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate. Thus, it helps in reducing the overall loss and improve the accuracy. The problem of choosing the right weights for the model is a daunting task, as a deep learning model generally consists of millions of parameters. It raises the need to choose a suitable optimization algorithm for particular application. The terms related to the optimization process include,

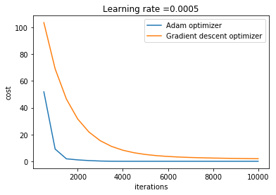
**1. Epoch** – The number of times the algorithm runs on the whole training dataset.

**2. Sample** – A single row of a dataset.

**3. Batch** – It denotes the number of samples to be taken to for updating the model parameters.

**4. Learning rate** – It is a parameter that provides the model a scale of how much model weights should be updated.

**5. Cost Function/Loss Function** – A cost function is used to calculate the cost that is the difference between the predicted value and the actual value.

**6. Weights/ Bias** – The learnable parameters in a model that controls the signal between two neurons.

*Fig. 5.2.4 Optimization Algorithm Characteristics*

The main optimization algorithms are,

• Gradient Descent

• Stochastic Gradient Descent

• Adam etc.

The optimization algorithm used here is **Adam**. Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. Straightforward to implement, computationally efficient, little memory requirements, well suited for problems that are large in terms of data and/or parameters, hyper-parameters have intuitive interpretation and typically require little tuning. Adam as combining the advantages of **AdaGrad** that maintains a per-parameter learning rate that improves performance on problems with sparse gradients and **RMSProp** that also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the. This means the algorithm does well on online and non-stationary problems.

**CHAPTER 6**

**WORK DONE**

**6.1 INTRODUCTION**

This chapter includes all the work that we have done towards developing an Automatic braking and Speed Control System which involves developing a code, data collection, preprocessing of the images and training the code.

**6.2 DATA COLLECTION/GENERATION**

Training data has to be collected by driving on a wide variety of roads and in a diverse set of lighting and weather conditions but due to the time limitations we were only able to collect data by driving our vehicle about 5-6 km in a bitumen road. The data was acquired using **Maruthi Suzuki Baleno** as test vehicle with camera was placed at the middle position. But this system has no dependencies on any particular vehicle make or model. Other road types include two-lane roads (with and without lane markings), residential roads with parked cars, tunnels, and unpaved roads.

****Data was collected in clear weather but it has to be collected in cloudy, foggy, snowy, and rainy weather, both day and night. In some instances, the sun was low in the sky, resulting in glare reflecting from the road surface and scattering from the windshield which was cleared later. Driver was encouraged to maintain full attentiveness.

*Fig. 6.2 Image of vehicle used for data collection*

To train the model, we simply selected data where the driver was staying in a lane, and discard the rest. We then sampled the video at 10 FPS because a higher sampling rate would include images that are highly similar, and thus not provide much additional useful information. To remove a bias towards driving straight the training data includes a higher proportion of frames that represent road curves.The aim of the preprocessing is to enhance the image features to avoid the distortion.

**6.3 IMAGE PRE-PROCESSING**

Image preprocessing is very necessary aspect as the image should not have any impurities, and it is accomplished to be better for the further process such as segmentation, feature extraction, and classification. Collection of data is challenging because of the type of device the image captured and factors like climatic condition, etc. The factors like noise and other aspects need to be eliminated. It is necessary for the preprocessing to remove noise. Preprocessing is the process which is used to enhance the evaluation and interpretation of an image. Image preprocessing is executed systematically; then, the further detection of disease in plants will be successful. The steps involved in the image processing are as follows,



1. Windowing or cropping
2. Data size reduction
3. Label encoding
4. Normalization

*Fig 6.3 Raw image from the vehicle camera*

**6.3.1 Windowing / Cropping**

Image cropping or Windowing is a technique that is used to select the most relevant areas of an image, discarding the useless ones. Handmade selection, especially in case of large photo collections, is a time-consuming task. So, we have used Automatic image cropping technique to extract the part of the image that was most relevant, according to specific requirements.

*Fig. 6.3.1 Image after windowing*

**

**6.3.2 Data size reduction**

The Gaussian pyramid was used to scale down the image to smaller size. It is a technique in image processing that breaks down an image into successively smaller groups of pixels to blur it. This type of precise mathematical blurring is used extensively in artificially intelligent computer vision as a pre-processing step. When a digital photograph is blurred in this way, edges of objects are easier to detect, enabling a computer to identify them automatically. The "pyramid" is constructed by repeatedly calculating a weighted average of the neighboring pixels of a source image and scaling the image down. It can be visualized by stacking progressively smaller versions of the image on top of one another. This process creates a pyramid shape with the base as the original image and the tip a single pixel representing the average value of the entire image.

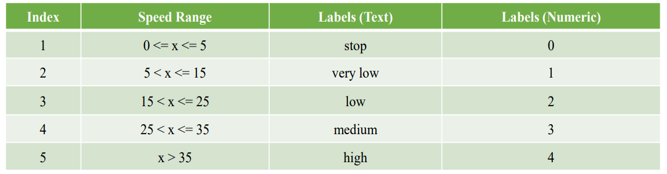
**

*Fig. 6.3.2.2 Image with low quality*

*Fig. 6.3.2.1 Image with high quality*

**6.3.3 Label encoding (Converting text label to numeric label)**

In machine learning, we usually deal with datasets that contain multiple labels in one or more than one column. To make the data understandable or in human-readable form, the training data is often labelled in words. Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning



*Table. 6.3.3 Label encoding data range table*

**6.3.4 Normalization**

Normalization is a data preparation technique that is frequently used in machine learning. The process of transforming the columns in a dataset to the same scale is referred to as normalization. Every dataset does not need to be normalized for machine learning. It is required only when features of machine learning models have different ranges. The most widely used types of normalization in machine learning are,

**1. Min-Max Scaling**

• Subtract the minimum value from each column’s highest value and divide by the range

• Each new column has a minimum value of 0 and a maximum value of 1

**2. Standardization Scaling**

• The term “standardization” refers to the process of centering a variable at zero and standardizing the variance at one

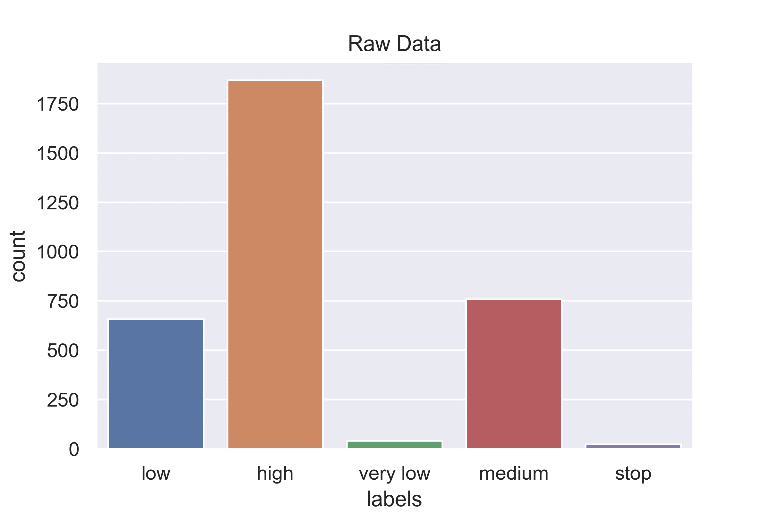
• Subtracting the mean of each observation and then dividing by the standard deviation is the procedure

**6.4 SPLITTING THE DATA**

The data collected has to be split into three groups namely,

1. Training Dataset

2. Testing Dataset

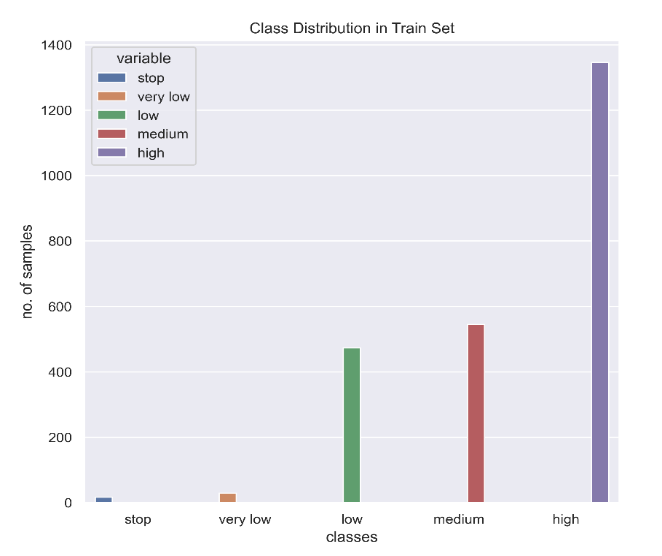
3. Validation Dataset

*Fig. 6.4 Bar chart of raw data*

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not

*Fig 6.4 Bar chart of validation data*

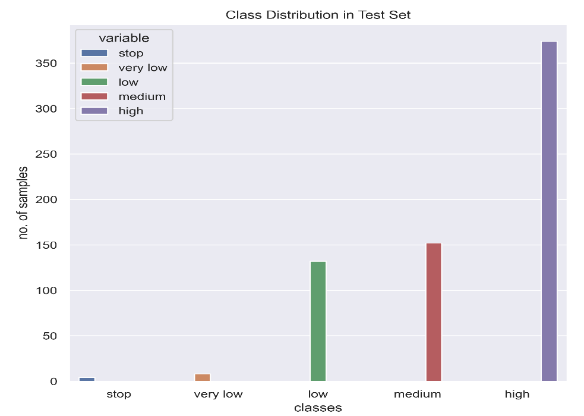
used to train the model. It can be used for classification or regression problems and can be used for any supervised learning algorithm. It is a fast and easy procedure to perform, the results of which allow to compare the performance of machine learning algorithms for predictive modeling problem. The train-test split procedure is appropriate when you have a very large dataset, a costly model to train, or require a good estimate of model performance quickly. Samples from the original training dataset were split into the two subsets using random selection and the training dataset was again split into train and validation sets.

**6.4.1 Training Dataset:**

*Fig 6.4.1 Bar chart of training data*

The sample of data used to fit the model. The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model sees and learns from this data.

**6.4.2 Validation Dataset**

The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. Hence the model occasionally sees this data, but never learns from this. So, the validation**** set affects a model, but only indirectly. This makes sense since this dataset helps during the “development” stage of the model.

*Fig 6.4.2 Bar chart of validation data*

**6.4.3 Test Dataset**

*Fig 6.4.3 Bar chart of test data*

The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. The test dataset provides the gold standard used to evaluate the model.It is only used once a model is completely trained (using the train and validation sets). The test set is generally what is used to evaluate competing models. It contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

**6.5 TRAINING THE MODEL**

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. This iterative process is called “model fitting”. The accuracy of the training dataset or the validation dataset is critical for the precision of the model. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning. Supervised learning is possible when the training data contains both the input and output values. Unsupervised learning involves determining patterns in the data. There are a few definitions related to this section are:

1. Parameter

A model parameter is a configuration variable that is internal to the model and whose value can be estimated from data. They are required by the model when making predictions. They are estimated or learned from data

1. Hyperparameter

A model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data. They are often used in processes to help estimate model parameters. They are often specified by the practitioner. They are often tuned for a given predictive modeling problem.

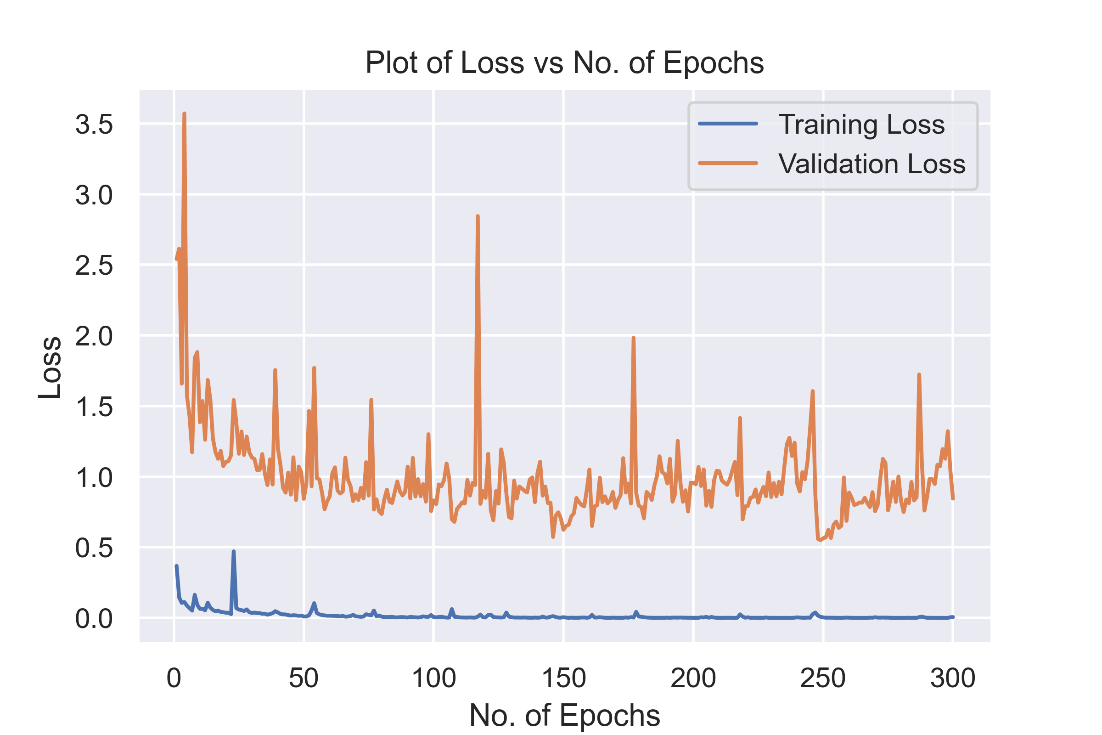
The plot of the variation of the training loss with the epoch is shown below. The variation of train accuracy and validation accuracy was also plotted against

number of epochs. At the beginning the training loss and validation loss were high. As the model was introduced with more and more training samples and the number of epochs increases the training loss got reduced along with the validation. The min loss was about 0.03 and the train accuracy was 99%. The validation loss was 0.55 and the validation accuracy was 89%. After epoch 100, the model was come to saturation, after which the testing accuracy remained the same and the validation loss started fluctuating.

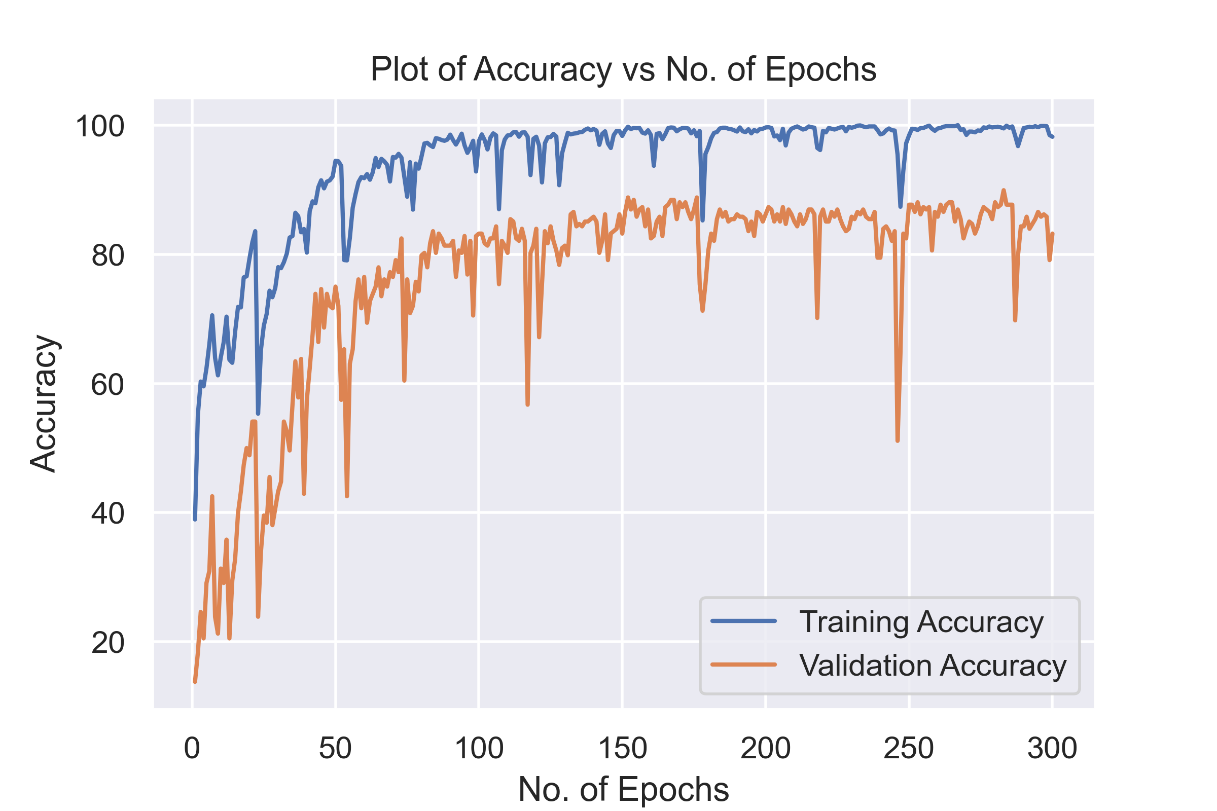
Hyperparameters Used:

• Batch size: 16

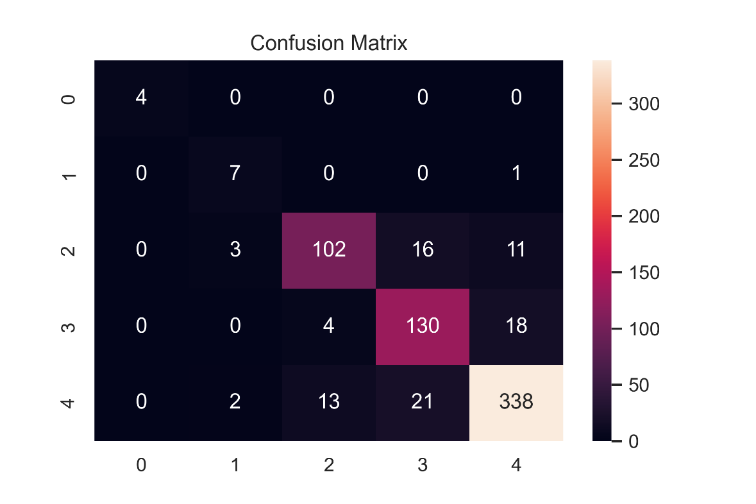
• Number of epochs: 150

 • Learning rate: 0.001

*Fig 6.5.1 Plot of Loss vs Number of epochs*



*Fig 6.5.2 Plot of Accuracy vs Number of epochs*

**6.6 TESTING AND EVALUATION**

Model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. The testing set consisting of a set of testing samples should be separated from the both training and validation sets but it should follow the same probability distribution as the training set. Model testing involves explicit checks for behaviors that we expect our model to follow. The testing was done on the test images and the test accuracy was about 87%. The confusion matrix for the test results is shown below and the model is evaluated based on that results.

*Fig 6.6 Confusion matrix for test data*

**CHAPTER 7**

**RESULT AND CONCLUSION**

**7.1 RESULTS**

Model training was done and the train loss was about 0.03 and the train accuracy was 98% after 150 epochs. The validation loss was 0.55 and the validation accuracy was 89% after 150 epochs. After epoch 100, the model was come to saturation, after which the testing accuracy remained the same and the validation loss started fluctuating. The testing was done on the test images and the test accuracy was about 87%. The confusion matrix for the test results was visualized and the model is evaluated based on that results. We could see that the model got confused near higher speed regions. This was because the speed data that we collected from the vehicle was not optimal because of some disturbances during data acquisition. We have trained the model by tuning various model hyper-parameters like batch size, learning rate, number of epochs etc. We have gone up to epochs of 300 and realized though the training loss was too low but the validation loss was fluctuating and it was started decreasing. The learning rate was decreased from 0.007 to 0.001 but we were only able to increase the test accuracy by 1%. At the end ,we concluded that the training should require more training data for getting good results.

**7.2 CONCLUSION**

* The braking system based on the deep learning is intelligent way of brake control which exhibits desirable and consistent brake control behavior for various scenarios where behavior of the pedestrian is uncertain
* It can reduce the velocity of the vehicle automatically when a threatening obstacle is detected
* The autonomous braking offers safe and comfortable brake control without exhibiting too early or too late braking
* The proposed system is more intelligent and prominent over conventional rule-based autonomous braking systems and servers as a principled and goal-oriented system
* The main challenge here is that the model has to be trained for lots of kms. with different driving and weather conditions to get best results

**REFERENCES**

* <https://pytorch.org/docs/stable/index.html>
* <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
* <https://developer.nvidia.com/blog/deep-learning-self-driving-cars/>
* <https://en.wikipedia.org/wiki/Convolutional_neural_network>
* <https://www.geeksforgeeks.org/activation-functions/>
* <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>