# SELF DRIVING AUTONOMOUS VEHICLE

Major Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

Anil Kushwaha (181245)

Under the supervision of

Dr. Vipul Kumar Sharma

to

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh**

# CERTIFICATE

## Candidate's Declaration

I hereby declare that the work presented in this report entitled " **SELF DRIVING AUTONOMOUS VEHICLE**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **Dr. Vipul kumar Sharma**(Assistant Professor in Computer Science).
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Anil Kushwaha

Anil Kushwaha, 181245.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Dr. Vipul Kumar Sharma
Assistant Professor
Computer Science
Dated:25/05/2022

## AKCNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **dr. Vipul Kumar Sharma Assistant Professor** Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of Web Development to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Vipul Kumar Sharma** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my par

Anil kushwaha

Anil kushwaha

# TABLE OF CONTENTS

# LIST OF ABBREVIATIONS

1) ML: Machine Learning
2) CNN: Convolutional Neural Network
3) DL: Deep Learning
4) CSV: Comma Separated Values
5) API: Application Program Interface
6) FC: Fully connected
7) CLI: Command Line Interface
8) DNN: Deep Neural Network
9) XML: Extensible Markup Language
10) PCA: Principal Component Analysis
11) MAE: Mean Absolute Error

# LIST OF FIGURES

## LIST OF TABLES

## ABSTRACT

We trained a Convolutional Neural Network (CNN) to map the raw pixels of a single front camera directly to control commands. This comprehensive the approach was surprisingly effective. By minimizing human training data, the learns to drive in heavy traffic on the highway as well as local roads with and without markings, and works in places with blurry visual guidance, such as parking lots and dirt roads. The system automatically stores internal representations of necessary processing steps (), such as detecting useful road features using only the person's steering angle () as a training signal. For example, it has never been specifically trained to detect road contours. Compared to explicit task decomposition such as lane marking detection, route planning and control, our end-to-end system optimizes all processing steps simultaneously. We argue that this will ultimately lead to better performance on the and smaller systems. The result will be better performance as the's internals will self-optimize to maximize overall system performance instead of optimizing, a human-chosen intermediate criterion. g., lane detection. These criteria have been chosen for easy human interpretation and do not automatically guarantee maximum system performance. Smaller networks are possible because the system learns to solve the problem with at least processing steps. We used NVIDIA DevBox and Torch 7 for training, and an NVIDIA DRIVETM PX autopilot computer running Torch 7 as well to determine the direction of movement. The system runs at 30 frames per second (FPS).

# CHAPTER-01: INTRODUCTION

## 1.1 Introduction

Every year, traffic accidents account for 2.6% of global deaths. Which nearly counts and stacks up to roughly 1.37 million a year — 3,343 a day. On top of this, some 20–50 million people are seriously injured and also many lost their lives in auto-related accidents each year. What is the root of these accidents? Human error.

From distracted driving to drunk driving to reckless driving to careless driving, all of this are related to one another. One poor or inattentive decision could be the difference between a typical drive and a life-threatening situation. But what if we could neutralize human error from the equation? Or in simpler words, we can say that what is car is driven by it's own??

"Autonomous cars are no longer beholden to Hollywood sci-fi films" — Elon Musk, the founder of Tesla Inc. and SpaceX believes within a decade, self-driving cars will be as common as elevators and now in today's world, we can see that Tesla is as common as in other countries like Maruti in India.

Our Project is all about this, a Self driving autonomous vehicle which would drive by itself on any random track, different factors are taken into consideration while preparing this model and this model is purely made for research purpose.

## 1.2 Problem Statement

As we already discussed the need of self-driving cars, for designing the model we need two different parts, first one is the software part, which we will do in this model and the other is the hardware part, in which the results of the first part are applied actually on that. Now for designing the software part, various factors should be taken into consideration that include lane markings, the distance from any obstacle, and the most challenging situation is weather conditions which are mainly rainy and snowy weather. The other factor which should be taken into consideration is the traffic lights, proper training of that should also be given so that no confusion arises in the final results. In this project a low cost prototype of self driving car is proposed and implemented. The car will have a cameras on board and with the feed video the analyser computer can detect traffic signal (turn right , turn left , stop) and give correct decisions to the car based on a particular model

## 1.3 Objective

The main objective of this project is to train autonomous vehicle and make a model that has an ability to drove by itself in any random track. We trained a Convolutional Neural

Network (CNN) to map raw pixels from a single front-facing camera directly to control commands. This end-to-end approach has been surprisingly effective. With minimal human training, the

learns to drive in traffic jams on highways as well as local roads with and without markings. It also works in areas with poor visual guidance, such as

parking lots and dirt roads. The system automatically learns the internal representations of necessary processing steps, such as detecting useful road features, using only the person's steering angle as a learning signal. For example, it has never been explicitly trained to detect the contours of a road.

## 1.4 Methodology

Many different approaches were used to categorize the self-driving autonomous vehicle. A machine learning technology called convolution neural network CNN was used in this study to construct a model that could be used to predict the outcome using image as an input. Just a few of the techniques that were employed to execute this project are listed below. One of the difficulties in self driving autonomous vehicle is taking into consideration the pixels of images, the accurate distance from the obstacle particularly in monsoon and snow weather conditions. In order to resolve these challenges, practical frameworks must be trustworthy enough.
The means available for the proposed strategy are as follows:

- Detect pixels from images in the dataset using pre trained ‘OpenCV’ function.
- Delete unwanted and outlier images which are unclear and with poor resolution.
- Read pixels from images.
- Develop CNN model.
- Create training data and testing data from variables.
- Train model and testing it.
- Finalizing and packing model for deployment.
- Deploy model for research purpose.

## 1.5 Organization

This report is divided into five modules, and a full description of each module of this project has been provided for the sake of clarity and comprehension. Module 1: This module serves as a formal introduction to the project and contains all of the necessary information. In this section, we introduce the reader to the project's numerous terms while also outlining the problem or motive for undertaking the project in the first CHAPTER-01: INTRODUCTION Page | 4 place. Along with this, we're also getting started on our project's purpose and the methods that will be used to carry it out. Module 2: This module contains a collection of current research studies that have been conducted in connection with our project. In this section, we place a greater emphasis on the methodology that the articles employed. In addition to this, we're keeping an eye on the results of their various initiatives as well. Module 3: We will go through the many stages of our development in this module, and we will learn about the design and implementation of our algorithm. In this section, we will also construct the model and attempt to represent it from a variety of perspectives, including analytical, computational, experimental, mathematical, and statistical perspectives, among others. Module 4: In this module, we will examine the performance of our project and provide recommendations for improvement. Module 5: This will be our last module, in which we will discuss the outcome of our research as well as examine our findings and conclusions. In addition to this, we will examine the project's future scope as well as any enhancements that may be feasible in the near future. In addition, we will describe some of the applications in which the system might be beneficial.

# CHAPTER-02 :LITERATURE SURVEY

## 2.1 Literature Survey

We trained a Convolutional Neural Network (CNN) to map the raw pixels of a single front camera directly to control commands. This comprehensive
approach has proven to be surprisingly robust. By minimizing human training data, the learns to drive in heavy traffic on the
highway as well as local roads with and without markings, and works in places with blurry visual guidance, such as
parking lots and dirt roads. Compared to explicit task decomposition such as lane marking detection, route planning and control, our end-to-end system optimizes all
processing steps simultaneously. We argue that this will ultimately lead to better

performance on
and lower systems. The result will be better performance as the
's internals will self-optimize to maximize overall system performance instead of optimizing
, a human-chosen intermediate criterion. g., lane detection. These criteria have been chosen for easy human interpretation and do not automatically guarantee maximum system performance. Smaller networks are possible because the system learns to solve the problem with at least
processing steps. CNNs with learned features have been in commercial use for over 20 years [3], but two recent developments have resulted in a rapid increase in adoption in the past few years. First, large, labeled datasets such as the Large Scale Visual Recognition Challenge (ILSVRC) were made available for training and validation. Second, CNN learning algorithms are implemented on massively parallel graphics processing units (GPUs) to significantly speed up training and inference. The DAVE2 is inspired by the pioneering work of Pomelo, who created the
Autonomous Land Vehicle in a Neural Network (ALVINN) system in 1989.

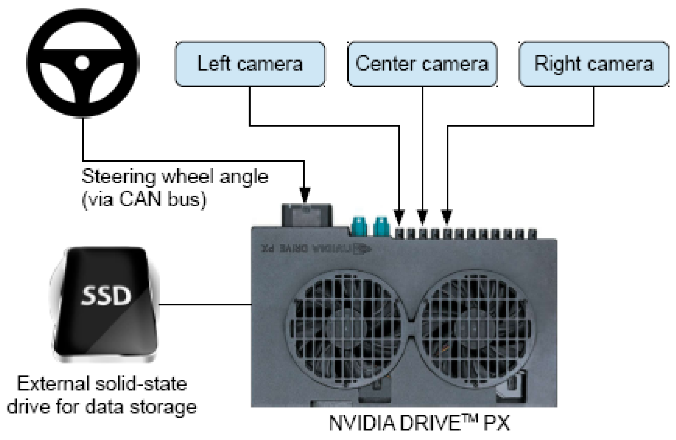


Figure 1: High-level view of the data collection system.

Training data was collected by driving on a wide variety of roads and in a diverse set of lighting
and weather conditions. Most road data was collected in central New Jersey, although highway data was also collected from Illinois, Michigan, Pennsylvania, and New York. 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, 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.

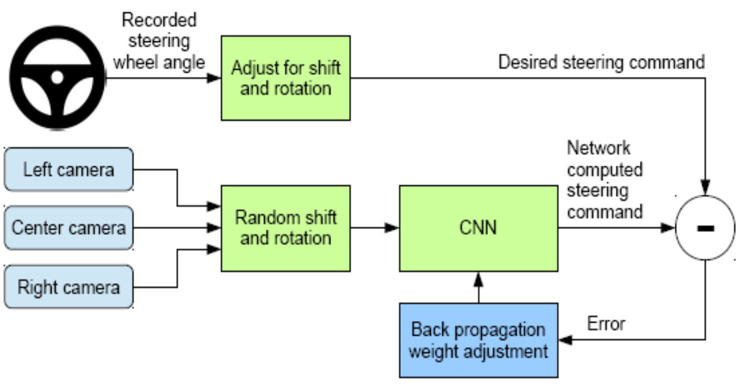

Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
-
Back propagation weight adjustment
Error

Figure 2: Training the neural network.

Data was acquired using either our drive-by-wire test vehicle, which is a 2016 Lincoln MKZ, or using a 2013 Ford Focus with cameras placed in similar positions to those in the Lincoln. The system has no dependencies on any particular vehicle make or model. Drivers were encouraged to maintain full attentiveness, but otherwise drive as they usually do. As of March 28, 2016, about 72 hours of driving data was collected.
We train the weights of our network to minimize the mean squared error between the steering command output by the network and the command of either the human driver, or the adjusted steering command for off-center and rotated images Our network architecture is shown in Figure 4. The network consists of 9 layers, including a normalization layer, 5 convolutional layers and 3 fully connected layers. The input image is split into YUV planes and passed to the network.

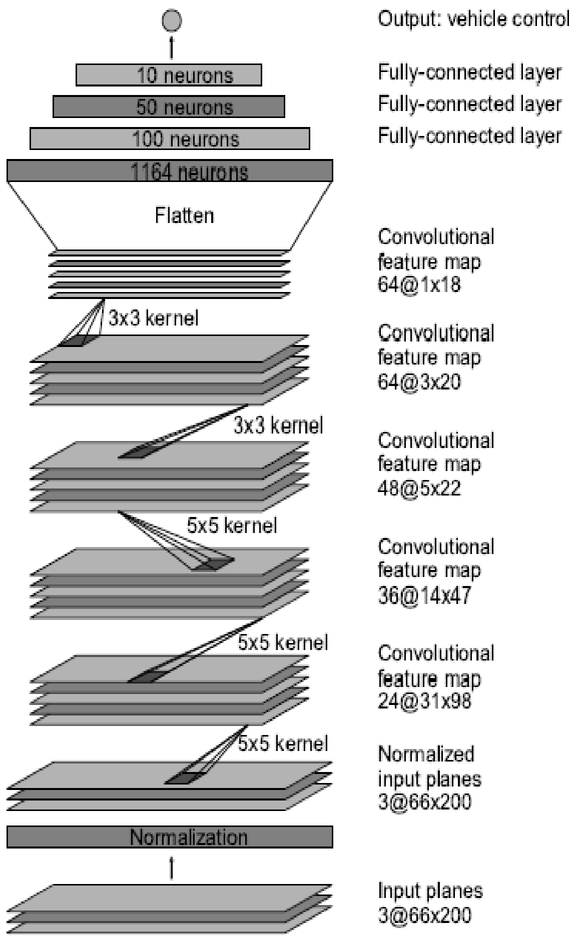


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

The first layer of the network performs image normalization. Normalizers are hard-coded and are not adjusted during training. Performing normalization on the network can change the normalization scheme and speed up GPU processing depending on the network architecture. The

convolutional layer was designed for feature extraction and was heuristically selected from a series of experiments to change the layer configuration. We use step convolution in the first three convolution layers with stride of 2_2 and kernel 5_5, and non-step convolution with kernel size 3_3 in the last two convolution layers.

Follows 5 convolutional layers with 3 fully connected layers, resulting in an output control value that is the inverse of the radius of rotation. The fully connected layer is designed to act as a controller for governance, but it is impossible to clearly demarcate which parts of the network primarily function as feature extractors and controllers through end-to-end systems. learning.

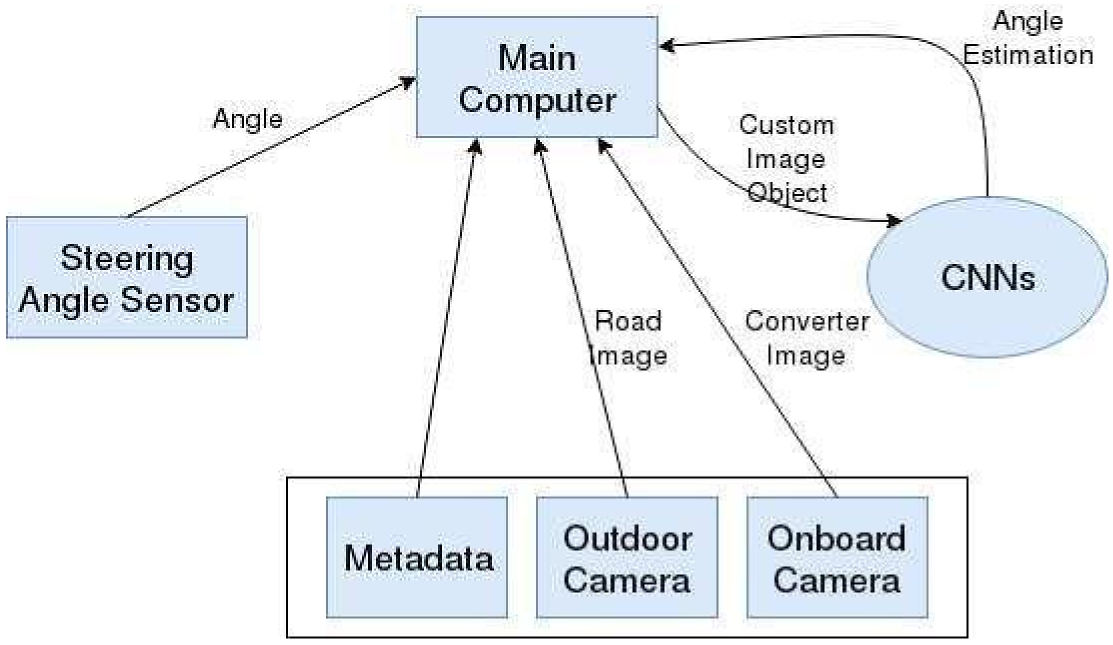
The first step in training a neural network is to choose which frames to use.
data collected by us tagged by road type, weather conditions and driver behavior (lane keeping, lane change, turns, etc.). To teach the CNN to follow a lane, it selects only the data that the driver stayed in the lane and discards the rest. Then we sample this video at 10 frames per second. Higher sampling rates contain very similar images and are not very useful. To eliminate linear bias, the training data contains many frames representing the curves of the road.
: After selecting the final frameset, augment the data by adding artificial translations and rotations to teach the network to recover from the wrong position or orientation. The magnitude of these perturbations is randomly chosen from a normal distribution. The distribution has a mean of zero and the standard deviation is twice the standard deviation measured by human drivers. Data padding adds unwanted artifacts as values increase.
Before road-testing a trained CNN, we first evaluate the network's performance in simulation. A simplified block diagram of the simulation system is shown in Figure 5.
The simulator takes pre-recorded videos from a forward-facing on-board camera on a human-driven data-collection vehicle and generates images that approximate what would appear if the CNN were, instead, steering the vehicle. These test videos are time-synchronized with recorded steering commands generated by the human driver.
Since human drivers might not be driving in the center of the lane all the time, we manually calibrate the lane center associated with each frame in the video used by the simulator. We call this position the “ground truth”.
The simulator transforms the original images to account for departures from the ground truth. Note that this transformation also includes any discrepancy between the human driven path and the ground truth. The transformation is accomplished by the same methods described in Section 2. The simulator accesses the recorded test video along with the synchronized steering commands that occurred when the video was captured. The simulator sends the first frame of the chosen test video, adjusted for any departures from the ground truth, to the input of the trained CNN. The CNN then returns a steering command for that frame. The CNN steering commands as well as the recorded human-driver commands are fed into the dynamic model [8] of the vehicle to update the position and orientation of the simulated vehicle.
The simulator then modifies the next frame in the test video so that the image appears as if the vehicle were at the position that resulted by following steering commands from the

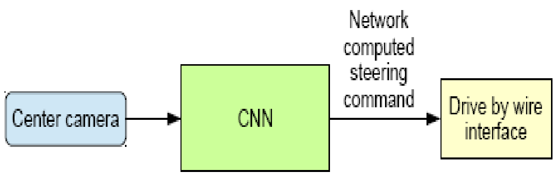
CNN. This new image is then fed to the CNN and the process repeats.
The simulator records the off-center distance (distance from the car to the lane center), the yaw, and the distance traveled by the virtual car. When the off-center distance exceeds one meter, a virtual human intervention is triggered, and the virtual vehicle position and orientation is reset to match the ground truth of the corresponding frame of the original test video.
Evaluating our networks is done in two steps, first in simulation, and then in on-road tests.
In simulation we have the networks provide steering commands in our simulator to an ensemble of prerecorded test routes that correspond to about a total of three hours and 100 miles of driving in Monmouth County, NJ. The test data was taken in diverse lighting and weather conditions and includes highways, local roads, and residential streets.

# CHAPTER-03 :SYSTEM DEVELOPMENT

## 3.1 Analysis/Design/Development/Algorithm

This project used a Udacity driving simulator with two different tracks. One of them was used to collect training data and the other was used to replace the test set not recognized by the model. The driving simulator records the data from the vehicle's perspective by storing images from three front "cameras". In addition, it provides various driving statistics such as throttle, speed and steering angle. We use the camera data as input to the model and expect to predict the handle angle in the range [-1, 1].





A) Firstly CNN model was chosen because it outperformed other approaches in terms of accuracy, quickness, operating in real-time upon CPU, and detecting faces of varied sizes and alignments. The picture was first extracted from the dataset and transformed to grayscale. The OpenCV cascade classifier was then utilized, and the pixels were sent through the OpenCV pretrained model ("camera_imgs.xml"), which generated the corner points for the rectangle where the obstacles and the lanes was detected. Finally, the pixels was stored into a separate folder for future processing. The photos were then run through a size filter function, which excluded outlier images (less than 5kb in size), and the final image produced was utilized as the dataset for model training.

B) When the images were captured by the camera, the images are sent to CNN model for further processing, when the images reaches the model, it was firstly checked that if that image has poor resolution , or the image was ruptured from somewhere, and if any of the points were found to be there, then the image is simply discarded because if the image is not clear, it would cause disturbance in identifying pixels and also lots of computations occur which would particularly waste time and lots of power.

C)After generating pixels, the main work comes into play, i.e reading of that pixels to make proper judgement on that basis, so when image reaches the CNN model and when the model generates pixels, the reading part comes, In that part, the lane detection, means to identify a particular track that where the track is moving and in which direction, and another thing which is most important is any obstacle in the path, because if any obstacle is founded, the results would change accordingly which is particularly based on size of obstacle, the direction in which it is present and whether that obstacle is moving or it is static.

D) Convolutional Neural Networks (CNNs) are used to model spatial information such as images. CNNs are very good at extracting features from images and are often considered general-purpose approximators of nonlinear functions. The

CNN can capture different patterns as the depth of the network increases. For example, a layer at the beginning of a network captures edges, while a deeper layer captures more complex features, such as the shape of an object (a leaf on a tree or a tire on a vehicle). This is why CNNs are the main algorithm for autonomous vehicles. A key component of the CNN is the convolutional layer itself. There is a convolutional kernel called the filter matrix. The filter matrix is convolved with the local region of the input image, which can

$$y_j = \sum w_{ij} * x + b_j,$$

be defined as

Where:

- the operator * represents the convolution operation,
- w is the filter matrix and b is the bias,
- x is the input,
- y is the output.

In practice, the size of the filter matrix is usually 3X3 or 5X5. During the learning process, the filter matrix is continuously updated to obtain reasonable weights. One of the properties of CNNs is that they share weights. You can represent two different transformations in the network using the same weight parameters. Common parameters save a lot of processing space. They can generate more diverse functional representations learned by the network. The output of a
CNN is usually fed to a nonlinear activation function. Activation functions can be used to solve inseparable linear problems in networks, and these functions can represent high-dimensional manifolds from low-dimensional manifolds. Commonly used activation functions are Sigmoid, Tanh, and ReLU listed below.

$$\begin{cases} \text{Sigmoid} : R = \frac{1}{1+e^{-y}} \\ \text{Tanh} : R = \frac{e^{y}-e^{-y}}{e^{y}+e^{-y}} \\ \text{ReLU} : R = \max\ (0, y) \end{cases}$$

It is worth noting that ReLU is the preferred activation function because it converges faster compared to other activation functions. The output of the convolution layer is also modified by the max pooling layer, which stores additional information about the input image such as background and texture.The three important CNN properties that make them versatile and a primary component of self-driving cars are:

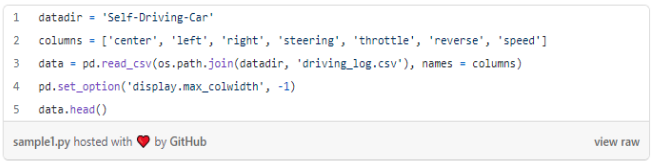
- **local receptive fields,**
- **shared weights,**
- **spatial sampling**.

These properties reduce overfitting and store representations and features that are vital for image classification, segmentation, localization, and more.

## 3.2 Model Development

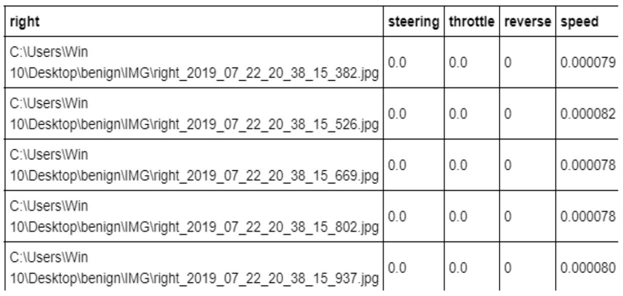
### 1) Dataset Used

The dataset has 6 columns — center, left, right (camera image paths), steering, throttle, reverse, speed (values). I have used pandas dataframe to display the first five rows in the dataset. In this project, Udacity driving simulator has been used which has two different tracks. One of them was used for collecting training data, and the other one — never seen by the model — as a substitute for the test set.

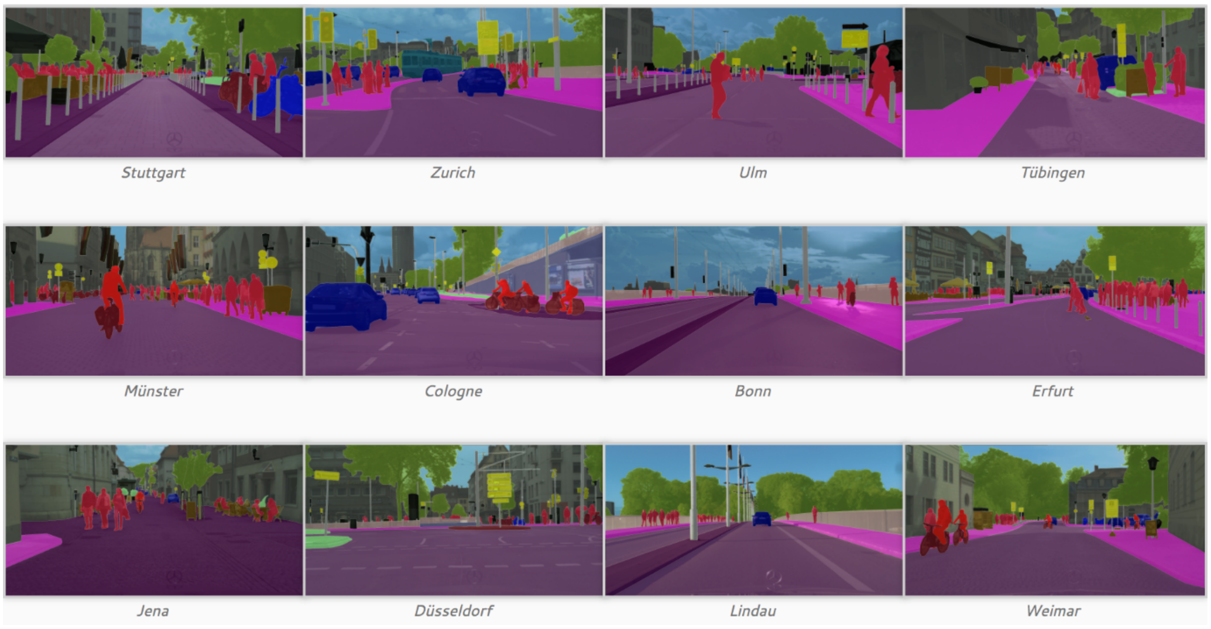


```python
datadir = 'Self-Driving-Car'
columns = ['center', 'left', 'right', 'steering', 'throttle', 'reverse', 'speed']
data = pd.read_csv(os.path.join(datadir, 'driving_log.csv'), names = columns)
pd.set_option('display.max_colwidth', -1)
data.head()
```

sample1.py hosted with ❤ by GitHub view raw



| right | steering | throttle | reverse | speed |
|---|---|---|---|---|
| C:\Users\Win 10\Desktop\benign\IMG\right_2019_07_22_20_38_15_382.jpg | 0.0 | 0.0 | 0 | 0.000079 |
| C:\Users\Win 10\Desktop\benign\IMG\right_2019_07_22_20_38_15_526.jpg | 0.0 | 0.0 | 0 | 0.000082 |
| C:\Users\Win 10\Desktop\benign\IMG\right_2019_07_22_20_38_15_669.jpg | 0.0 | 0.0 | 0 | 0.000078 |
| C:\Users\Win 10\Desktop\benign\IMG\right_2019_07_22_20_38_15_802.jpg | 0.0 | 0.0 | 0 | 0.000078 |
| C:\Users\Win 10\Desktop\benign\IMG\right_2019_07_22_20_38_15_937.jpg | 0.0 | 0.0 | 0 | 0.000080 |



`

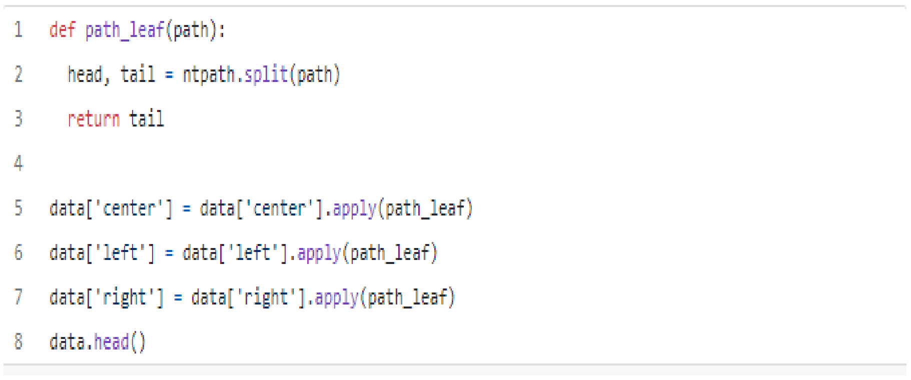
Since the prefix of the left, right and center image paths was the same for all the rows so I decided to remove the prefix part throughout the dataset

## 2) Preprocessing of the Dataset

The first leg of the test was based on traffic data from El Camino Real (small curve, mostly straight traffic) and the second leg was based on traffic data from San Mateo to Half Moon Bay (bend and highway traffic). I ended up downloading over 100GB of driving data, but the driving time was about an hour, which is clearly not enough to create an all-in-one end-to-end solution. We tried to preprocess the data in a way that makes it easier to model motion in the network. We had high hopes for the optical flow feature, but it didn't seem to help.

Firstly the path of the images is given so long and all the images has same prefix part, the onl difference in the name of the images is the suffix part, so we create a function called path_leaf that removes all the prefix part which is common among all the images names.

Then, In the end, I scaled the input image to 256 x 192, converted it to grayscale, calculated the difference between frames with a delay of 1, and took two consecutive difference images as input. For example, we took [x_{t} - x_{t1}, x_{t1} - x_{t2}] as input at time t. where x corresponds to a grayscale image. Future frames were not used to predict the current steering angle.



```python
def path_leaf(path):
  head, tail = ntpath.split(path)
  return tail

data['center'] = data['center'].apply(path_leaf)
data['left'] = data['left'].apply(path_leaf)
data['right'] = data['right'].apply(path_leaf)
data.head()
```

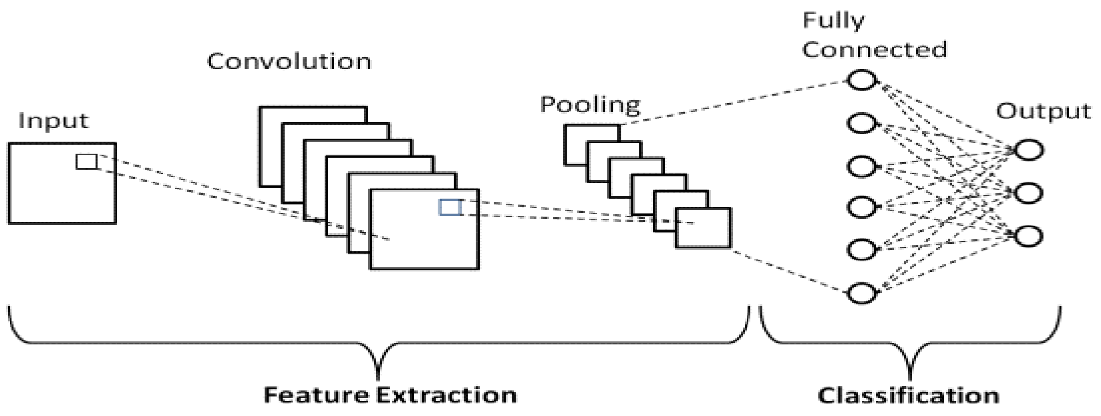
3.) Visualization of dataset and creating training and testing data-

Various bargraphs, countplots, and line graphs were used to examine the data's consistency. The opencv imshow function was used to verify for accurate pixel alignment. Importing the necessary tensorflow and keras libraries and preparing training and testing data for gender and age.

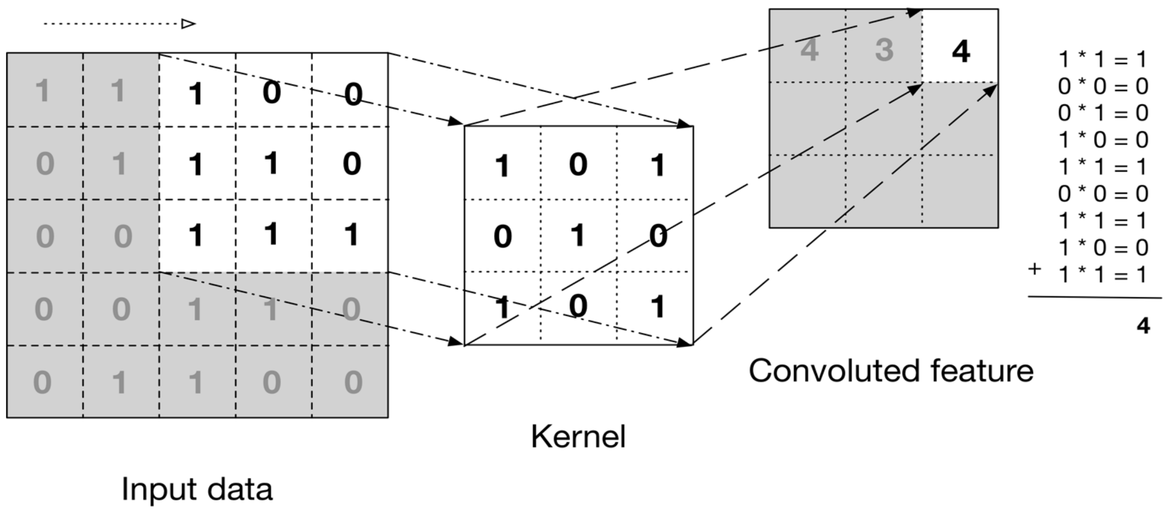
4.) Convolution neural network-

In the field of artificial intelligence, CNNs are a kind of ‘Deep Neural Network’ (DNN) that can identify and categorize certain aspects in pictures and are commonly used to analyze visual data. Its uses span from video and picture identification to image classification, image analysis in medicine, and computer vision and natural language processing. When two functions are multiplied together, the result is a third function that reflects how the form of one function is altered by the other. Multiplying two matrices-representable photos produces an output from which features from the image may be extracted.

Feature Extraction is a convolution technique that separates and identifies the image's distinct features for analysis. Convolution output is fed into a fully connected layer that uses the information gathered in earlier rounds to forecast the image's class. The CNN has three layers: convolutional, pooling, and fully-connected (FC) layers. A CNN architecture will be constructed when these layers are layered.



A) Convolution layer- The input pictures are initially processed via a convolutional layer, which is utilized to extract different characteristics. The input picture is processed by this layer, which extracts characteristics from it. This layer contains the picture matrix and the kernel/filter that extracts the image's features. It is possible to get an image feature map of dimension (a-(fh+1), b-(fw+1),1)) from an image matrix of volume (a). In this layer, the convolution of the input picture with a filter of a certain size is carried out. Dot products are calculated by multiplying the input image's pixels by the filter's pixel size. We get information about the image's corners and edges from this layer's output, which is called a Feature map. Afterwards, this feature map is supplied to further layers in order to learn more about the input picture.



**Convolutional Layer**

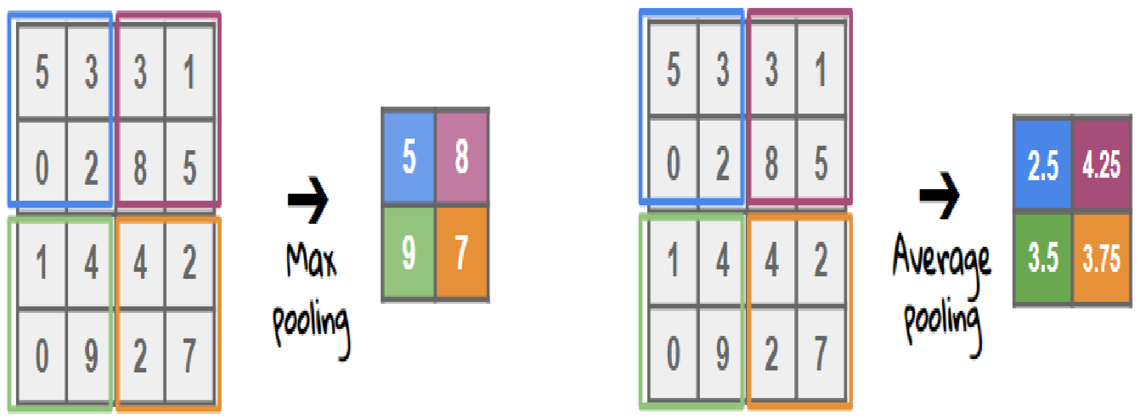
B) Pooling layer- This layer's major goal is to lower the size of the convolved feature map in order to reduce the computational expenses of the algorithm. When the photos, like those we used in the project, are too huge, the pooling layer decreases the number of parameters or blocks in the feature matrix.

Max Pooling: the filter area is pooled to its fullest extent.

Average Pooling: pools the average of the elements.

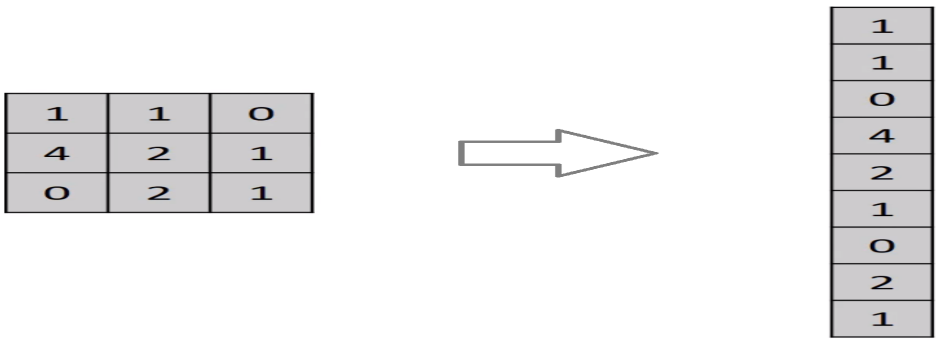
Sum Pooling: The total of the components in the filter is gathered in a pool called the sum pooling.

The pooled feature map was created using Max Pooling for this project. The biggest element on the feature map is used in Max Pooling.



**Pooling Layer**

C) Flattening layer- Flattening is the process of transforming a dataset into a one-dimensional array that may be used as input to the following layer. Convolutional layers' output is flattened into a single feature vector. Also known as a fully connected layer, it is linked to the final classification model.



**Flattening Layer**

D) Fully Connected Layer- The Fully Connected (FC) layer is utilized to link neurons between multiple layers and comprises of weights and biases along with neurons. It is common practice to position the last few layers of a CNN architecture before the output layer.

E) The dropout layer and activation function are the two additional factors that must be taken into consideration in addition to these three layers.

- Dropout- Overfitting may be avoided by using a dropout layer, which removes some neurons from the neural network during training, resulting in a smaller model. The neural network loses 30% of its nodes when it reaches a dropout of 0.3.
- Activation Functions- It is possible to learn and estimate any form of continuous and complicated connection between variables in a network using an Activation Function, which is a function of the network's variables. To put it another way, this function determines which information should be transported to the output and which information should not. The activation functions that we utilized were as follows:

  ReLU (Rectified Linear Unit): When utilized in our model, it is one of the most often used activation functions since it ensures non-linearity by changing all negative values in the matrix to 0 while leaving all positive numbers as they are.

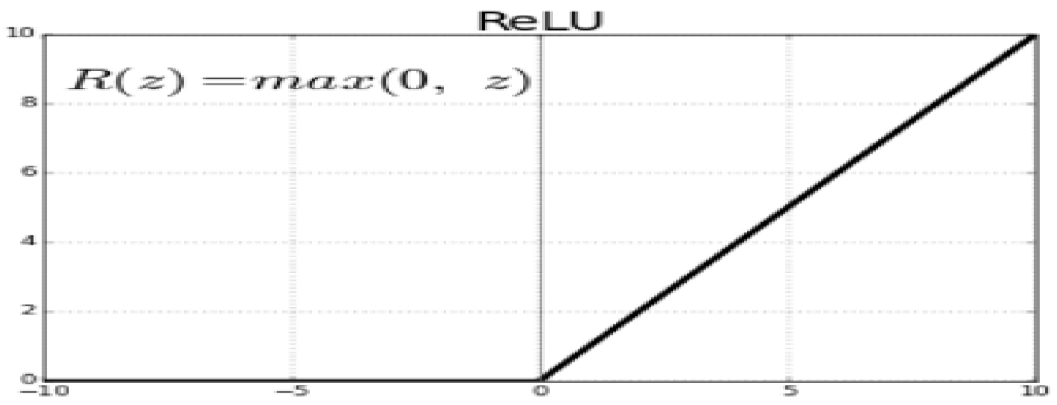
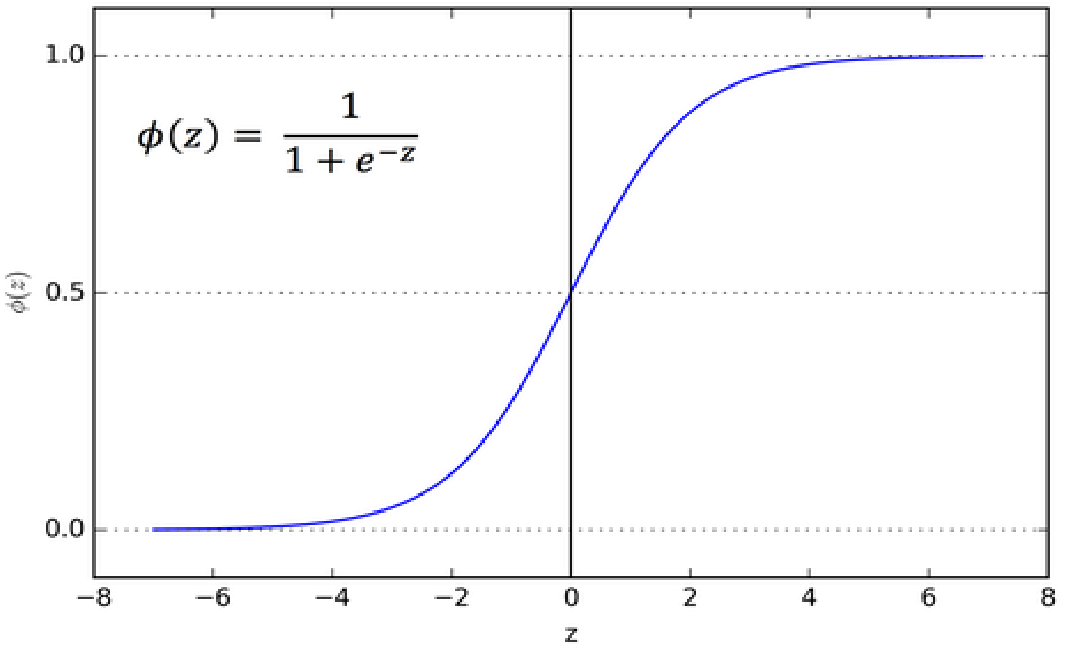


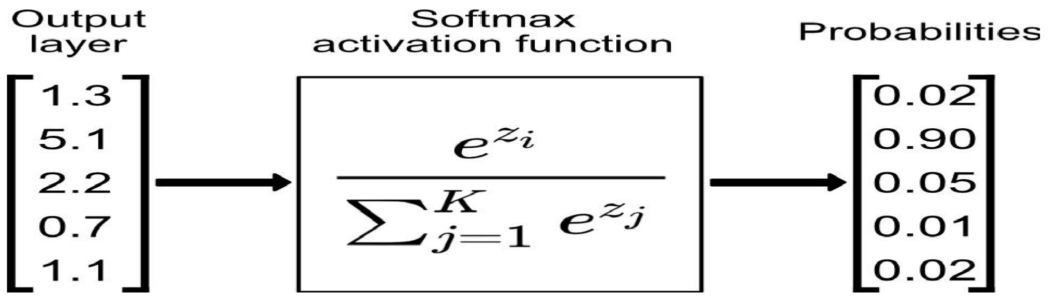
**Fig.10 Relu function**

Sigmoid function- The major reason we pick the sigmoid function is because it occurs between and inside the interval (0 to 1). As a result, it is particularly useful for models in which we must forecast the probability as an output. Because the probability of anything exists only between 0 and 1, the sigmoid function is the most appropriate option. It is possible to differentiate the function. That is, we can determine the slope of the sigmoid curve at any two locations on the curve. The function is monotonic; however, the derivative of the function is not. The logistic sigmoid function has the potential to cause a neural network to get stuck during its training phase.



**Sigmoid function**

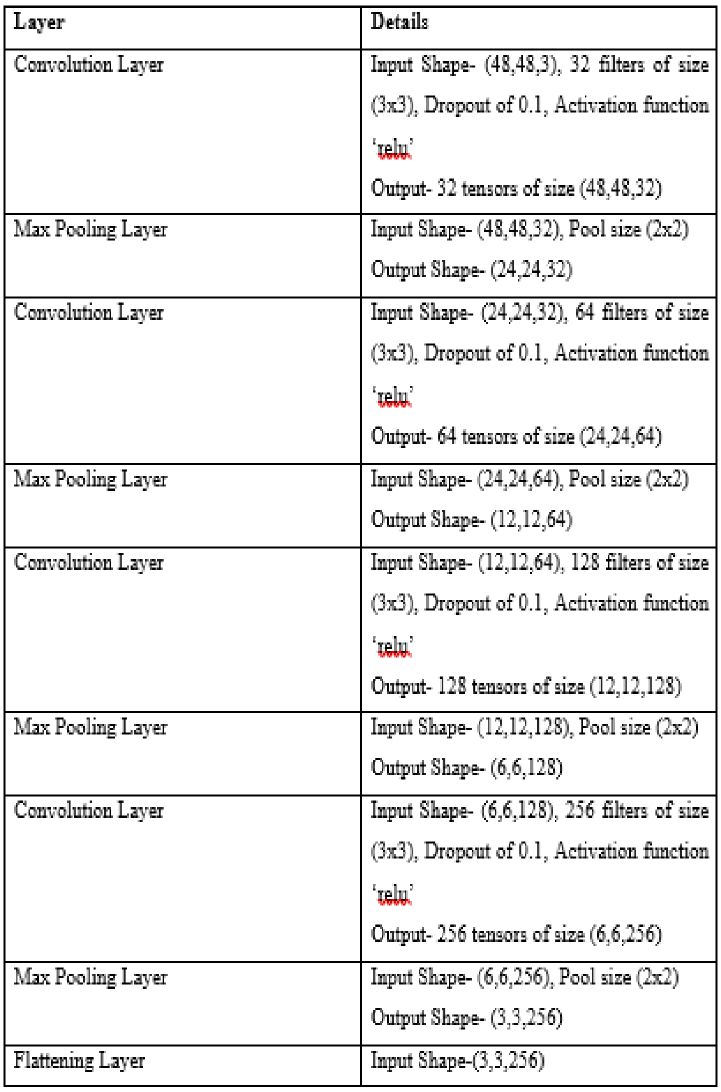
Softmax function- This is an activation function that turns a vector of integers into a vector of likelihood. Our model is designed to produce 24 values corresponding to various age classes. The Softmax function normalizes the outputs and changes them from weighted sum values to probability values adding to one. The Softmax function is depicted in the diagram below.



Output layer
Softmax activation function
Probabilities
$\begin{bmatrix}1.3\\5.1\\2.2\\0.7\\1.1\end{bmatrix}$
$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
$\begin{bmatrix}0.02\\0.90\\0.05\\0.01\\0.02\end{bmatrix}$

**Fig.12 Softmax function**

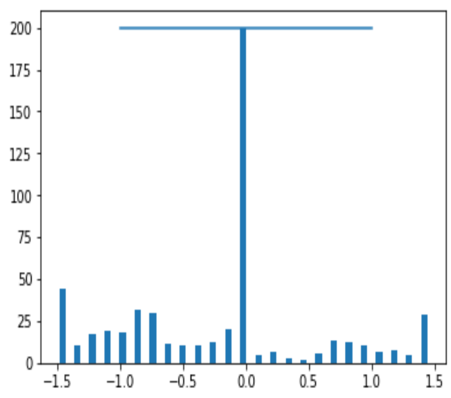
5.) Creating CNN model for self driving autonomous vehicle:-



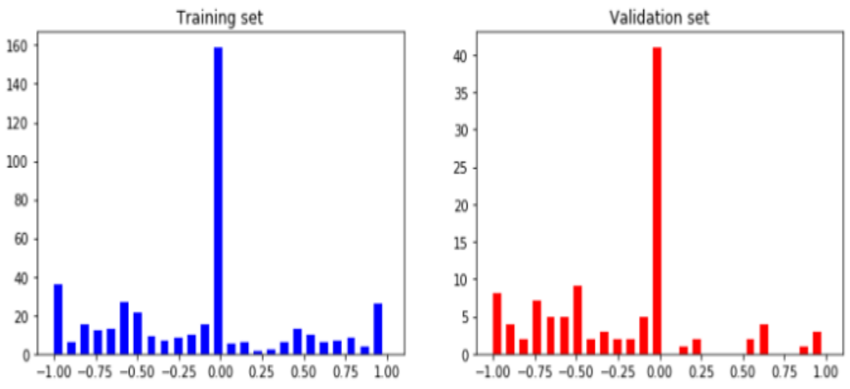
| Layer | Details |
|---|---|
| Convolution Layer | Input Shape- (48,48,3), 32 filters of size (3x3), Dropout of 0.1, Activation function 'relu'<br>Output- 32 tensors of size (48,48,32) |
| Max Pooling Layer | Input Shape- (48,48,32), Pool size (2x2)<br>Output Shape- (24,24,32) |
| Convolution Layer | Input Shape- (24,24,32), 64 filters of size (3x3), Dropout of 0.1, Activation function 'relu'<br>Output- 64 tensors of size (24,24,64) |
| Max Pooling Layer | Input Shape- (24,24,64), Pool size (2x2)<br>Output Shape- (12,12,64) |
| Convolution Layer | Input Shape- (12,12,64), 128 filters of size (3x3), Dropout of 0.1, Activation function 'relu'<br>Output- 128 tensors of size (12,12,128) |
| Max Pooling Layer | Input Shape- (12,12,128), Pool size (2x2)<br>Output Shape- (6,6,128) |
| Convolution Layer | Input Shape- (6,6,128), 256 filters of size (3x3), Dropout of 0.1, Activation function 'relu'<br>Output- 256 tensors of size (6,6,256) |
| Max Pooling Layer | Input Shape- (6,6,256), Pool size (2x2)<br>Output Shape- (3,3,256) |
| Flattening Layer | Input Shape-(3,3,256) |

# CHAPTER-04 : PERFORMANCE ANALYTICS

## 4.1 Outputs At Various Stages

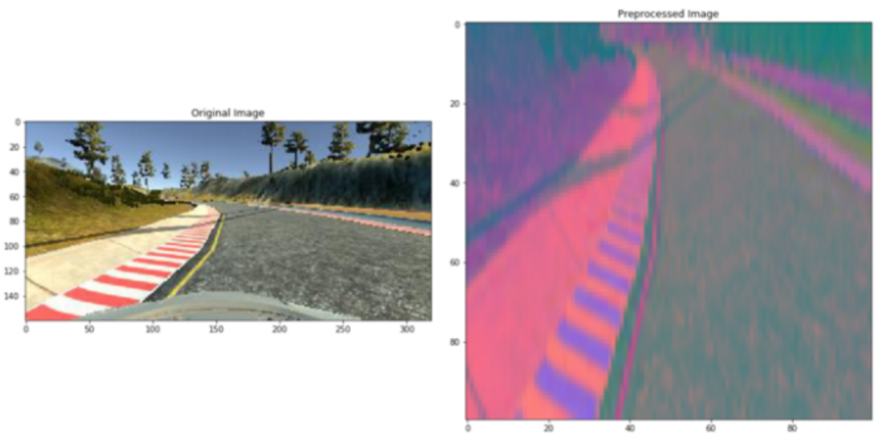


We draw the steering angle values graph, and we can see a huge spike near zero, which clearly indicates that maximum times, car is driving straight.



Before this, data is split up using 80-20 rule, which means 80% of the data is used in training set and the rest 20% of data is used in the validation set.

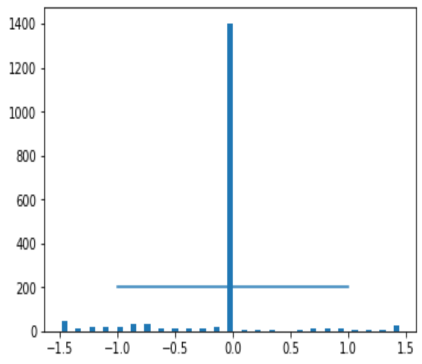
After that we plotted sample training and validation steering angle distributions and we can see that both the graphs are same in nature and both having huge spike near zero, which means maximum times car is driving straight



In this part, we have shown the raw and the processed image, means the image on the left side was a preprocessed image which is captured by a camera in real time and on the right side is the processed image, which is obtained using CNN which can be easily understood by the model we prepare and in this, we can clearly see the lane and also the distance between the lane and the car and the other important thing is the direction, we can see that the track is diverting towards the left.

```
## Visualize data
num_bins = 25
samples_per_bin = 200
hist, bins = np.histogram(data['steering'], num_bins)
center = bins[:-1] + bins[1:] * 0.5  # center the bins to 0

## Plot
plt.bar(center, hist, width=0.05)
plt.plot((np.min(data['steering']), np.max(data['steering'])), (samples_per_bin, samples_per_bin))
```

```
[<matplotlib.lines.Line2D at 0x7f17f8175860>]
```



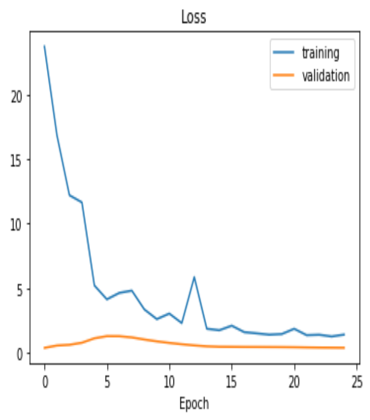
```
W0825 19:24:11.704631 139742454921088 deprecation.py:506] From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3445: calling
dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
W0825 19:24:11.817271 139742454921088 deprecation_wrapper.py:119] From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The name tf.trai
n.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 4, 4, 2048)        23587712
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 2048)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 32768)             0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               3276900
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dropout_3 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
_________________________________________________________________
dropout_4 (Dropout)          (None, 10)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 11
=================================================================
Total params: 26,870,183
Trainable params: 4,337,191
Non-trainable params: 22,532,992
_________________________________________________________________
None
```

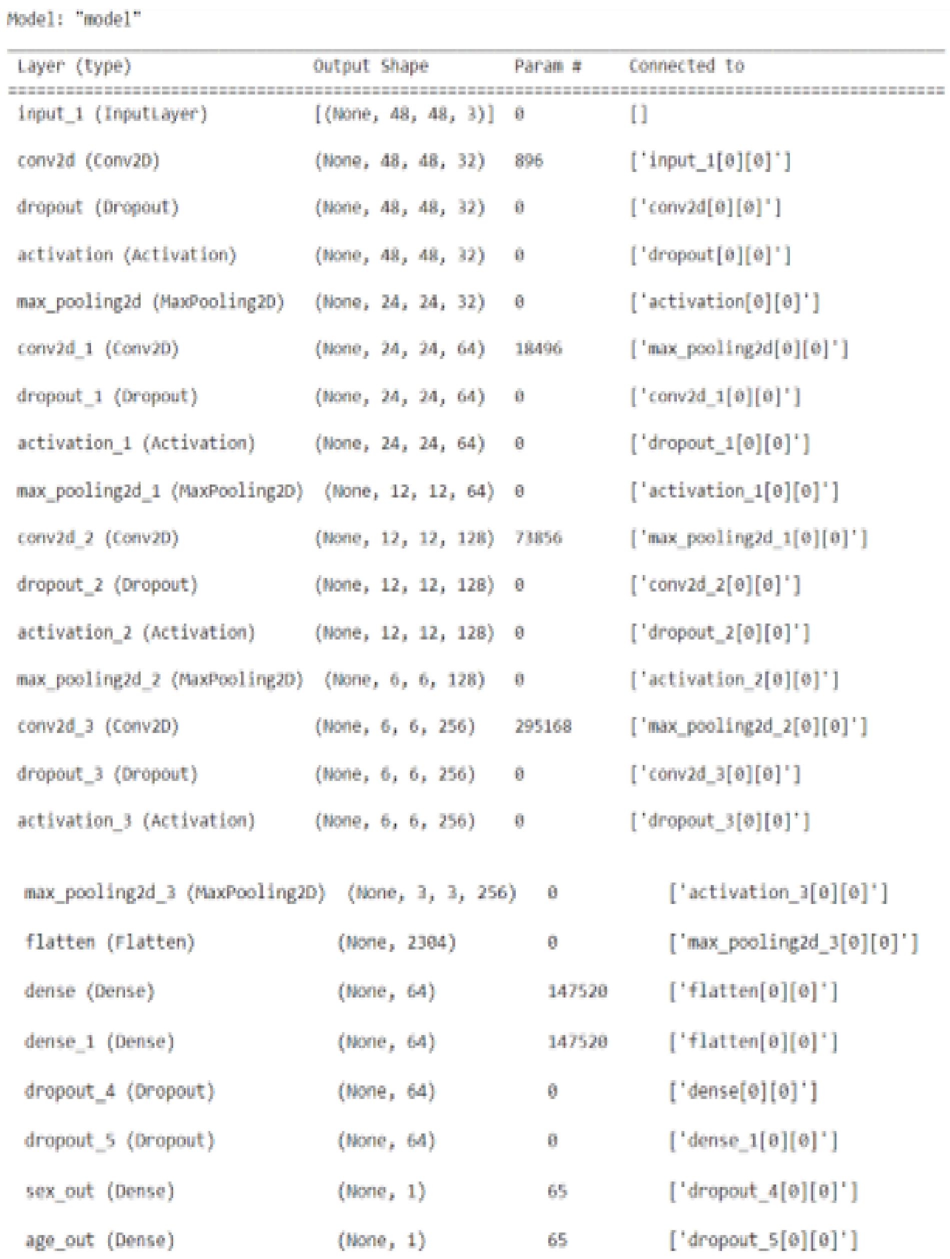
In [27]:

```python
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training', 'validation'])
plt.title('Loss')
plt.xlabel('Epoch')
```

Out[27]:

```
Text(0.5, 0, 'Epoch')
```

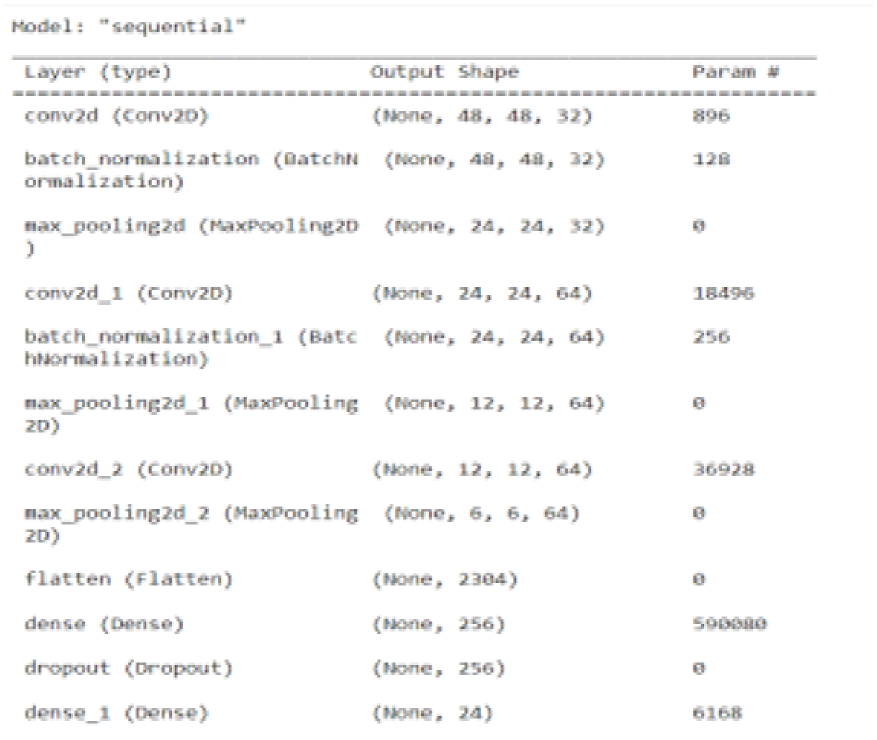




```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 48, 48, 3)]  0           []

 conv2d (Conv2D)                (None, 48, 48, 32)   896         ['input_1[0][0]']

 dropout (Dropout)              (None, 48, 48, 32)   0           ['conv2d[0][0]']

 activation (Activation)        (None, 48, 48, 32)   0           ['dropout[0][0]']

 max_pooling2d (MaxPooling2D)   (None, 24, 24, 32)   0           ['activation[0][0]']

 conv2d_1 (Conv2D)              (None, 24, 24, 64)   18496       ['max_pooling2d[0][0]']

 dropout_1 (Dropout)            (None, 24, 24, 64)   0           ['conv2d_1[0][0]']

 activation_1 (Activation)      (None, 24, 24, 64)   0           ['dropout_1[0][0]']

 max_pooling2d_1 (MaxPooling2D)  (None, 12, 12, 64)  0           ['activation_1[0][0]']

 conv2d_2 (Conv2D)              (None, 12, 12, 128)  73856       ['max_pooling2d_1[0][0]']

 dropout_2 (Dropout)            (None, 12, 12, 128)  0           ['conv2d_2[0][0]']

 activation_2 (Activation)      (None, 12, 12, 128)  0           ['dropout_2[0][0]']

 max_pooling2d_2 (MaxPooling2D)  (None, 6, 6, 128)   0           ['activation_2[0][0]']

 conv2d_3 (Conv2D)              (None, 6, 6, 256)    295168      ['max_pooling2d_2[0][0]']

 dropout_3 (Dropout)            (None, 6, 6, 256)    0           ['conv2d_3[0][0]']

 activation_3 (Activation)      (None, 6, 6, 256)    0           ['dropout_3[0][0]']

 max_pooling2d_3 (MaxPooling2D)  (None, 3, 3, 256)   0           ['activation_3[0][0]']

 flatten (Flatten)              (None, 2304)         0           ['max_pooling2d_3[0][0]']

 dense (Dense)                  (None, 64)           147520      ['flatten[0][0]']

 dense_1 (Dense)                (None, 64)           147520      ['flatten[0][0]']

 dropout_4 (Dropout)            (None, 64)           0           ['dense[0][0]']

 dropout_5 (Dropout)            (None, 64)           0           ['dense_1[0][0]']

 sex_out (Dense)                (None, 1)            65          ['dropout_4[0][0]']

 age_out (Dense)                (None, 1)            65          ['dropout_5[0][0]']
```

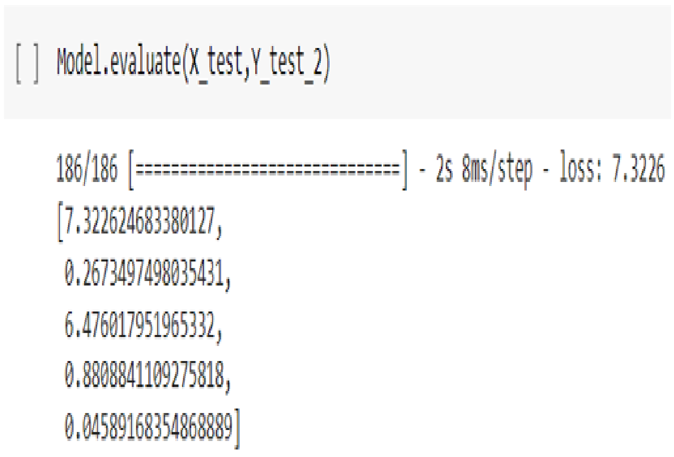
This is the visulizations of the different layers in CNN, the pooling layers, the droupout layers, convolutions layers.

Fig.23 Model-1 Summary

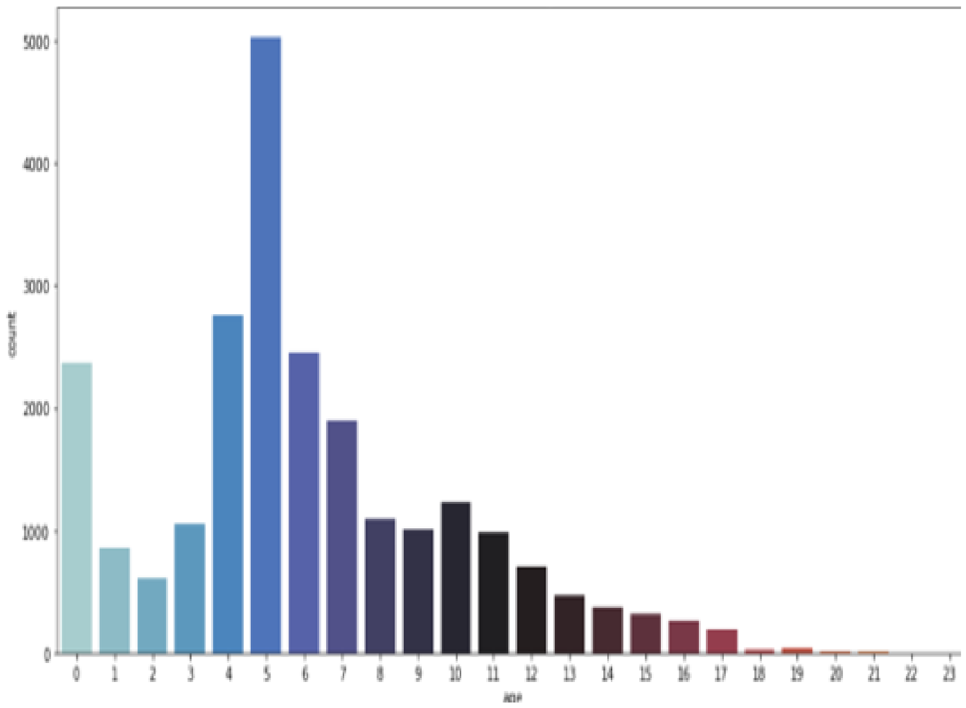


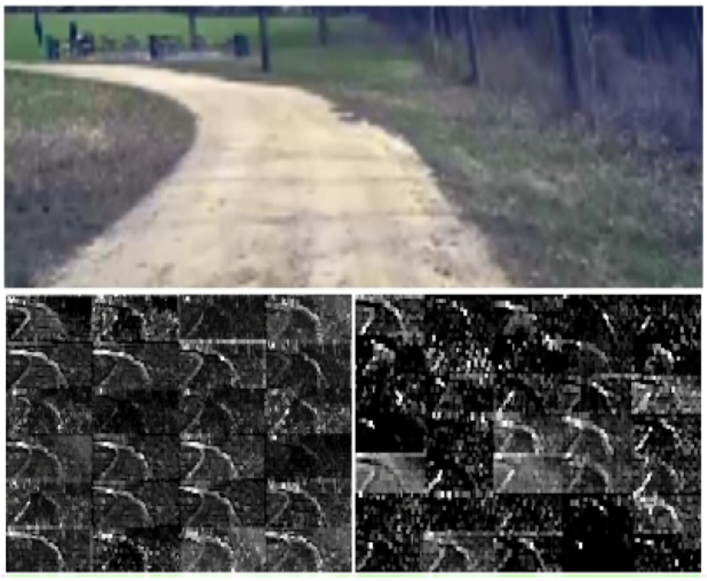
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 32)        896

 batch_normalization (BatchN  (None, 48, 48, 32)       128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 24, 24, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 24, 24, 64)        18496

 batch_normalization_1 (Batc  (None, 24, 24, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 12, 12, 64)        36928

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 256)               590080

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 24)                6168
```

5) Evaluate Model-



```
[ ] Model.evaluate(X_test,Y_test_2)

186/186 [==============================] - 2s 8ms/step - loss: 7.3226
[7.322624683380127,
 0.2673497498035431,
 6.476017951965332,
 0.8808841109275818,
 0.045891683548688889]
```


count
age



# CHAPTER-05 : CONCLUSIONS

## 5.1 Conclusions

Self-driving cars will revolutionize road trips by making road trips safer and more efficient. In this article, we have discussed the key components such as LiDAR, RADAR, cameras and, most importantly, the algorithms that enable autonomous vehicles.

Promising but still has a lot of room for improvement. For example, modern self-driving cars are at Level 2 of Level 5 of development, meaning that they still need someone ready to intervene if needed.

Few things need to be taken care of:

1. The algorithms used are not yet optimal enough to perceive roads and lanes because some roads lack markings and other signs.
2. The optimal sensing modality for localization, mapping, and perception still lack accuracy and efficiency.
3. Vehicle-to-vehicle communication is still a dream, but work is being done in this area as well.
4. The field of human-machine interaction is not explored enough, with many open, unsolved problems.

Still, the technology we've developed so far is amazing. And with orchestrated efforts, we can ensure that self-driving systems will be safe, robust, and revolutionary.

We have empirically demonstrated that CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning, and control. A small amount of training data from less than a hundred hours of driving was sufficient to train the car to operate in diverse conditions, on highways, local and residential roads in sunny, cloudy, and rainy conditions. The CNN is able to learn meaningful road features from a very sparse training signal (steering alone).

The system learns for example to detect the outline of a road without the need of explicit labels during training. More work is needed to improve the robustness of the network, to find methods to verify the robustness, and to improve visualization of the network internal processing steps. We have empirically demonstrated that CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning, and control. A small amount of training data from less than a hundred hours of driving was sufficient to train the car to operate in diverse conditions, on highways, local and residential roads in sunny, cloudy, and rainy conditions. The CNN is able to learn meaningful road features from a very sparse training signal (steering alone).

The system learns for example to detect the outline of a road without the need of explicit labels during training. More work is needed to improve the robustness of the network, to find methods to verify the robustness, and to improve visualization of the network internal processing steps.

To sum up, I can say that it was a really interesting and complex project at the same time. Deep learning is an exciting field and we live in an age of invention.

In 10 years most of us will be without cars. We will be subscribing to services like Uber. And you'll pay $149 a month and drive to work every morning on your way to work.

## 5.2 Future Scope

There are several ways in which this technique might be enhanced in order to mitigate the weaknesses in autonomous vehicles. First, by using a huge and complicated model that was trained on more photographs for which many photographs of different tracks in different weather conditions are being used, then analyzing the estimate accuracy & performance speed while not consuming a significant number of resources, and then optimizing it to be useful in application forms. It's also realistic to create a training model through CNN using a large dataset which contains photos, then evaluate it using the testing dataset, given the low prediction accuracy in self driving, which are due to several issues we discussed previously. In addition, we want to build a 'Restful API' for the same, which will allow it to be utilized by anybody without difficulty. Next we will create a website using Django – a python framework and then we will deploy that website using Heroku. Again we are saying, this all project has been purely based on research and don’t try it to copy for training a real car.

## 5.3 Applications

*A) <u>ETHICAL CONSIDERATIONS</u>*

The ethics of autonomous vehicles can be summarized as the cart problem. In the classic version of the problem, the trolley is charging along the rails and the driver has to choose between staying on the track and switching to the track. If he stays on the track, 5 people will die. But if he changes them, one person dies. In 2014, researchers at the Massachusetts Institute of Technology created a "moral machine" to test people around the world's attitudes and approaches to the complex issues that will arise with autonomous vehicles. Essentially, the car had to choose what to collide with**.**For example, the car is going to collide with a pet, should it swerve to instead collide with a human? Such situations were set up for disabled, old, young, law abiding citizens, criminals, men, women, fit and sickly people. The decisions programmed into self-driving cars may have to depend on where in the world the car is located.

*B) <u>ENVIRONMENTAL IMPACT</u>*

*Most of the research assessing the environmental impact of autonomous vehicles has so far been inconclusive. This is because we do not know exactly what kind of changes the roll out of fully autonomous technologies will have on the behaviour and patterns of humans. Bear in mind that autonomous vehicles are different from electric vehicles, and it is not necessary that all autonomous vehicles will be electric. According to a study by University of California researchers, in a business as usual scenario where vehicles are automated but not electrified, emissions will go up by 50 percent, by 2050. However, if the vehicles are electrified and automated, greenhouse gas emissions are projected to reduce by 80 percent.* *Most of the research assessing the environmental impact of autonomous vehicles has so far been inconclusive. This is because we do not know exactly what kind of changes the roll out of fully autonomous technologies will have on the behaviour and patterns of humans. Bear in mind that autonomous vehicles are different from electric vehicles, and it is not necessary that all autonomous vehicles will be electric. According to a study by University of California researchers, in a business as usual scenario where vehicles are automated but not electrified, emissions will go up by 50 percent, by 2050. However, if the vehicles are electrified and automated, greenhouse gas emissions are projected to reduce by 80 percent.* *Electricity production itself can have a negative impact on the environment.*

Generally speaking, there are several factors that are expected to contribute to reducing emissions. Ride sharing is expected to become increasingly common, rather than one person per vehicle. The availability of on-demand shared mobility is expected to make a significant contribution to reducing emissions. Mobility on Demand is expected to reduce the number of vehicles owned by individuals. This will be facilitated by new ownership models such as car sharing and subscriptions. Platooning of vehicles and Coordination between the two is expected to contribute to reducing emissions by reducing congestion, avoiding frequent stops and reducing travel time. Due to the increased safety provided by autonomous vehicles, the weight of the vehicle can be reduced by eliminating safety systems.

## *C) CITIES OF THE FUTURE*

Closely related to the environmental impact is how the adoption of autonomous vehicles can change the urban landscape. Factors that affect a city are the environment. As more people choose the sharing option, the demand for parking space decreases. In an ideal scenario, most vehicles are on the road and busy, rather than waiting for their owners in a parking space. Cities can earn much less from parking spaces. Also, real estate prices may adjust as more space becomes available for other uses. Pick-up and drop-off points can be a potential new revenue stream and require proper design, especially in congested areas. Access to these points is payable in the city. for example.

When all vehicles become fully autonomous, road designs are likely to change as well. For example, instead of a dedicated lane for buses and trucks, you can simply program all vehicles to keep their lanes open only when buses pass. In fact, policymakers could use these technologies to ensure that autonomous vehicles ultimately deliver value. For example, single-seater vehicles may have limited or higher rates.

Historically, one of the biggest factors influencing the size of a city has been the time it takes to travel from the periphery to the city center. With the advent of automobiles, highways, and high-speed transportation systems, cities have also grown exponentially in size. The introduction of autonomy could increase the amount of time people are willing to spend commuting. So far, the average travel time around the world has been about 30 minutes, regardless of the means of transportation. This means you can reach the city center from the outskirts of town in about 30 minutes. If the passengers of future self-driving cars are more tolerant of long-distance commuting, it is possible that the area of large cities will expand. In short, cities are expected to get bigger.

## *D) CYBERSECURITY*

Because there are no Level 5 autonomous vehicles yet, researchers have little direct cybersecurity experience with autonomous vehicles. However, security researchers have outlined several potential scenarios to ensure that appropriate security measures are in place. Some of these threats aren't entirely new, but in the context of self-driving cars, they could be even more terrifying. Because computers, smartphones, and even cloud storage systems only need to protect the data they hold. It is entirely in cyberspace that makes them valuable in some way. But with self-driving cars, people's lives can be at risk.Let's take ransomware as an example. It infects the device and prevents users from accessing content until payment is complete. Now imagine that the door is closed and locked when you get into the car. You cannot leave or travel anywhere without making a cryptocurrency payment. Safety devices built into vehicles can also be counterproductive. In a fully autonomous vehicle, the driver may not have a backup to control in case of a problem, and even if there is a backup, no one in the vehicle may actually be able to drive the vehicle. Automakers expect to solve these problems by establishing safe driving destinations when problems arise (eg, they discover that they have been hacked). But what if this mechanism works in emergency situations, such as during a hospital visit?According to University of Michigan researchers, the safety risks faced by autonomous vehicles are not well understood. This is because autonomous vehicles are both physical and virtual systems. This means that the impact of the threat is doubled. They are vulnerable to traditional car thieves and thieves. They are also vulnerable to hackers, spoofers and DDoS attacks. Researchers have developed tools to identify different types of threats, attack vectors, and motives for their consequences. André Weimerskirch, chair of the University of Michigan Cyber Security Working Group, said, “Cybersecurity is an underestimated research area in autonomous vehicle development. Our tools not only mark important early steps in addressing these challenges, but also provide a blueprint for effectively identifying and analyzing cybersecurity threats and creating effective approaches to secure autonomous transportation systems.”

# REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, Winter 1989. URL: http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with
deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and
K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages
1097–1105. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[3] L. D. Jackel, D. Sharman, Stenard C. E., Strom B. I., , and D Zuckert. Optical character recognition for self-service banking. AT&T Technical Journal, 74(1):16–24, 1995.

[4] Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/challenges/LSVRC/.

[5] Net-Scale Technologies, Inc. Autonomous off-road vehicle control using end-to-end learning, July 2004. Final technical report. URL: http://net-scale.com/doc/net-scale-dave-report.pdf.http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf.

[6] Dean A. Pomerleau. ALVINN, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, 1989. URL: http://repository.cmu.edu/cgi/viewcontent. cgi?article=2874&context=compsci.

[7] Wikipedia.org. DARPA LAGR program. http://en.wikipedia.org/wiki/DARPA_LAGR_
Program.

[8] Danwei Wang and Feng Qi. Trajectory planning for a four-wheel-steering vehicle. In Proceedings of the 2001 IEEE International Conference on Robotics & Automation, May 21–26 2001. URL: http: //www.ntu.edu.sg/home/edwwang/confpapers/wdwicar01.pdf.

[9] DAVE 2 driving a lincoln. URL: https://drive.google.com/open?id= 0B9raQzOpizn1TkRIa241ZnBEcjQ.

[10] https://www.kaggle.com/roydatascience/training-car

## APPENDICES

Moreover, the automotive industry spends around €77 billion worldwide on R&D in order to nurture innovation and to stay competitive [4, 5].

The rapid development of communication technology and the need to cater to the aging population in developed countries has potentially made AVs a necessity and a vital business paradigm [6]. In light of looming new ideas and technologies such as social networks, smart phones, and AVs, some scholars have emphatically warned that the landscape of transportation is rapidly changing [7–9]. An example is Uber which is sweeping cities to the extent that taxi companies are struggling to retain business and to remain competitive. Manyika et al. New technologies in communications and robotics have a significant impact on our daily lives, and transportation is no exception. These technologies have given rise to the potential of autonomous vehicle (AV) technology, which aims to reduce accidents, energy consumption, pollution and congestion while improving transportation accessibility. The idea of autonomous vehicles has been around for decades, but huge costs have prevented large-scale production [1]. However, over the past decade, research and development efforts to realize the idea of AV have accelerated. For example, the advent of Google Car has sparked interest in driverless cars [2, 3]. Moreover, the automotive industry spends around €77 billion worldwide on R&D in order to nurture innovation and to stay competitive [4, 5].

The rapid development of communication technology and the need to cater to the aging population in developed countries has potentially made AVs a necessity and a vital

business paradigm [6]. In light of looming new ideas and technologies such as social networks, smart phones, and AVs, some scholars have emphatically warned that the landscape of transportation is rapidly changing [7–9]. An example is Uber which is sweeping cities to the extent that taxi companies are struggling to retain business and to remain competitive. Manyika et al. [10] lists automotive automation as one of the top ten disruptive technologies of the future. As a result of intense competition among automakers, 2020 has become the year on the horizon for delivering commercial AV to the entire market[1]. figs. 1 provides an overview of competition among major automakers [2]. Perhaps the middle of this century will be the maturity period for the AV market. Based on the deployment and adoption of existing smart vehicle technologies (such as automatic transmission and hybrid electric drive) [3], drones are expected to account for around 50% of vehicle sales, 30% of vehicles and 40% of total vehicles. . Journey to 2040

[14] depicted the AV in a figure with two additional components to ensure a successful working AV paradigm: “Connected” and “Big Data.” Accordingly, the terms “Connected” or “Connected Vehicle” refer to the technologies that ensure communication between all contributing agents or stakeholders including pedestrians, authorities and vehicles, as well as infrastructure.Footnote1 Figure 2 depicts a conceptual representation of a connected system. The connected component will require a massive amount of data from a variety of sources. As a result, “Big Data” is a term used to highlight the importance of handling such an unprecedented amount of information for which special provisions including software and hardware will be required. AV is associated with a range of positive societal outcomes, such as safer transportation systems, lower transportation costs, people with walking disabilities, and some mobility for low-income families. The direct social value to be created by 2025 is estimated to be between \$0.2 trillion and \$1.9 trillion per year. This positive impact is the driving force behind the emergence of AV technology and makes it a viable economic model for the foreseeable future and into the future.

Some people think that AV is an interdisciplinary technology that requires seeing through a wide-angle lens. Maddox et al. [14] depicted the AV in a figure with two additional components to ensure a successful working AV paradigm: “Connected” and “Big Data.” Accordingly, the terms “Connected” or “Connected Vehicle” refer to the technologies

that ensure communication between all contributing agents or stakeholders including pedestrians, authorities and vehicles, as well as infrastructure.Footnote1 Figure 2 depicts a conceptual representation of a connected system. The connected component will require a massive amount of data from a variety of sources. As a result, “Big Data” is a term used to highlight the importance of handling such an unprecedented amount of information for which special provisions including software and hardware will be required.

Each of these components is or has been the subject of extensive research in various fields. As such, AV technology could be considered at the crossroads of many disciplines such as transportation science, electrical engineering, information technology, software and hardware engineering, law, ethics, and philosophy. In this article, we look at the AV from a transportation point of view. We aim to shed light on the overarching implications of the AV for scholars, policy makers, planners, and practitioners involved in the transportation sector[Footnote2].

Interestingly, the existing literature does not discuss the methods by which AVs find and determine their routes in the road networks (vehicle routing). Perhaps, it is presumed that AVs are not different to other cars in vehicle routing. As noted before, connected vehicle technology is an indispensable part of a working AV scheme. Such (realtime) communication data may result in collaboration between the AVs directionality capabilities, leading to more efficient and intelligent pathfinding (or traffic flow). 4444 Realtime data (including travel time and incidents) can be processed and analyzed centrally in order to calculate and direct (or advise) AVs towards the best possible route. In particular, we elaborate on features directly pertaining to transport planning such as safety, fuel consumption, road pricing and parking requirements, land use, and demand forecasting. We also touch on other related issues such as cybersecurity, law/regulation, as well as ethical concerns. 4444 The aim is to highlight the opportunities and the challenges that may arise from the introduction and application of AVs. First we shall consider AV within the context of existing transportation systems and society at large, as well as define some related terminologies. We will then endeavor to show the impact of AVs for the short and long term future based on previous studies. In order to do this, we have reviewed over 118 references related to AV technology which have been published mainly in the past 5 years so as to provide a comprehensive and updated narrative. In particular, we elaborate on features directly pertaining to transport planning such as

safety, fuel consumption, road pricing and parking requirements, land use, and demand forecasting. We also touch on other related issues such as cybersecurity, law/regulation, as well as ethical concerns.
The aim is to highlight the opportunities and the challenges that may arise from the introduction and application of AVs. First we shall consider AV within the context of existing transportation systems and society at large, as well as define some related terminologies. We will then endeavor to show the impact of AVs for the short and long term future based on previous studies. In order to do this, we have reviewed over 118 references related to AV technology which have been published mainly in the past 5 years so as to provide a comprehensive and updated narrative.
Interestingly, the existing literature does not discuss the methods by which AVs find and determine their routes in the road networks (vehicle routing). Perhaps, it is presumed that AVs are not different to other cars in vehicle routing. As noted before, connected vehicle technology is an indispensable part of a working AV scheme. Such (realtime) communication data may result in collaboration between the AVs directionality capabilities, leading to more efficient and intelligent pathfinding (or traffic flow).
Realtime data (including travel time and incidents) can be processed and analyzed centrally in order to calculate and direct (or advise) AVs towards the best possible route. In this way, more sophisticated and robust vehicle routing models can be developed, for which this manuscript uses the term "vehicle navigation". We also propose a search model based on the concept of system optimality to find the optimal traffic pattern (minimizing the total travel time of the system).
In the remainder of this article, sect. 2 contains a brief history of AV. sect
3 defines the level of automation. secretary. 4 shows the principle of AB operation. secretary. 5 describes the sensor and monitoring technologies required for real-time data collection and communication. secretary. 6 describes the advantages and disadvantages of AB. and seconds. 7 is dedicated to aircraft navigation models. Conclusions are also provided in the section. 7.