IEM
श्रद्धावान लभते ज्ञानम्
INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

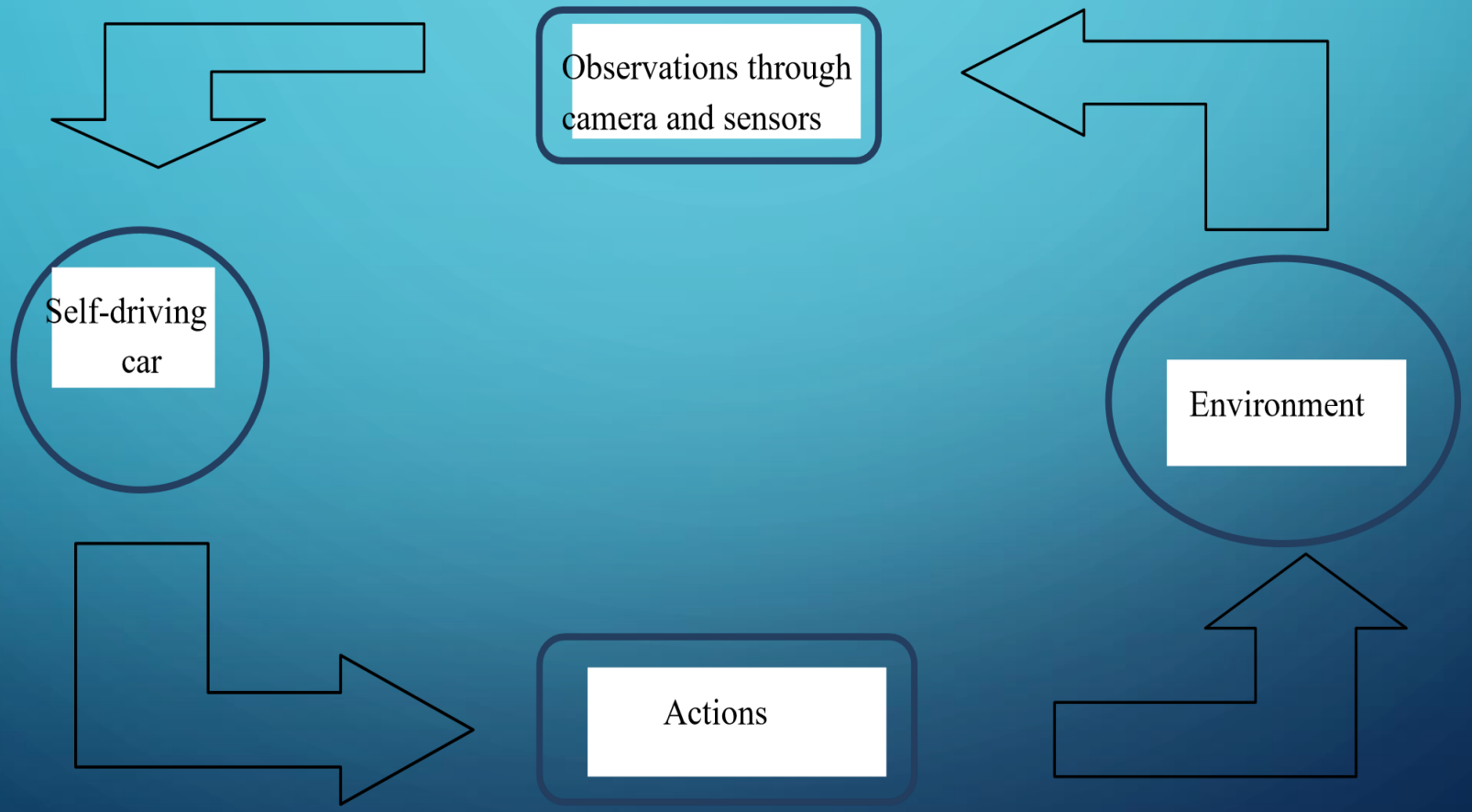
AUTONOMOUS TOY CAR USING COMPUTER VISION AND DEEP LEARNING TECHNIQUES

FINAL YEAR PROJECT

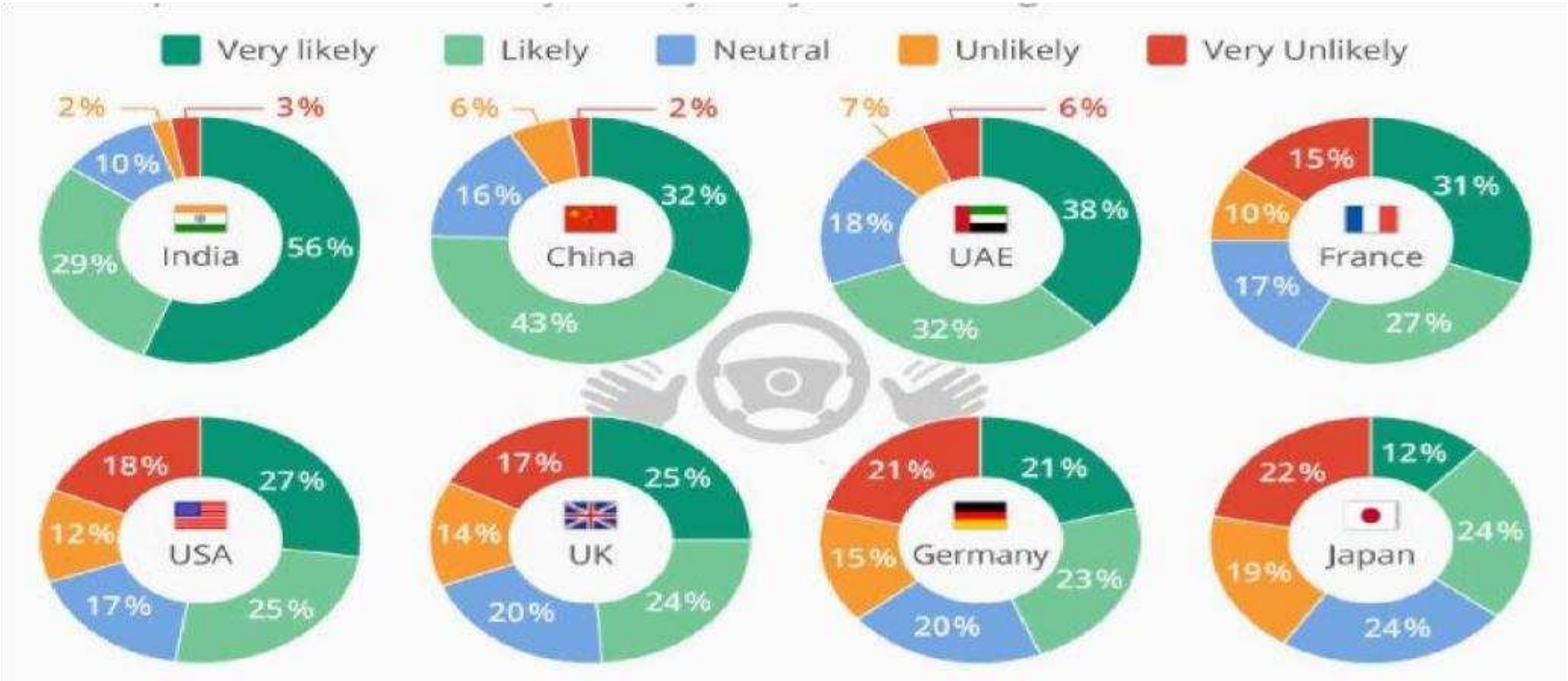
NAME – SOUJANYA SYAMAL (ROLL- *10400318065*)

CONTENT:

Introduction:-



The likeness of people around the world towards self-driving vehicles



PROJECT OVERVIEW:-

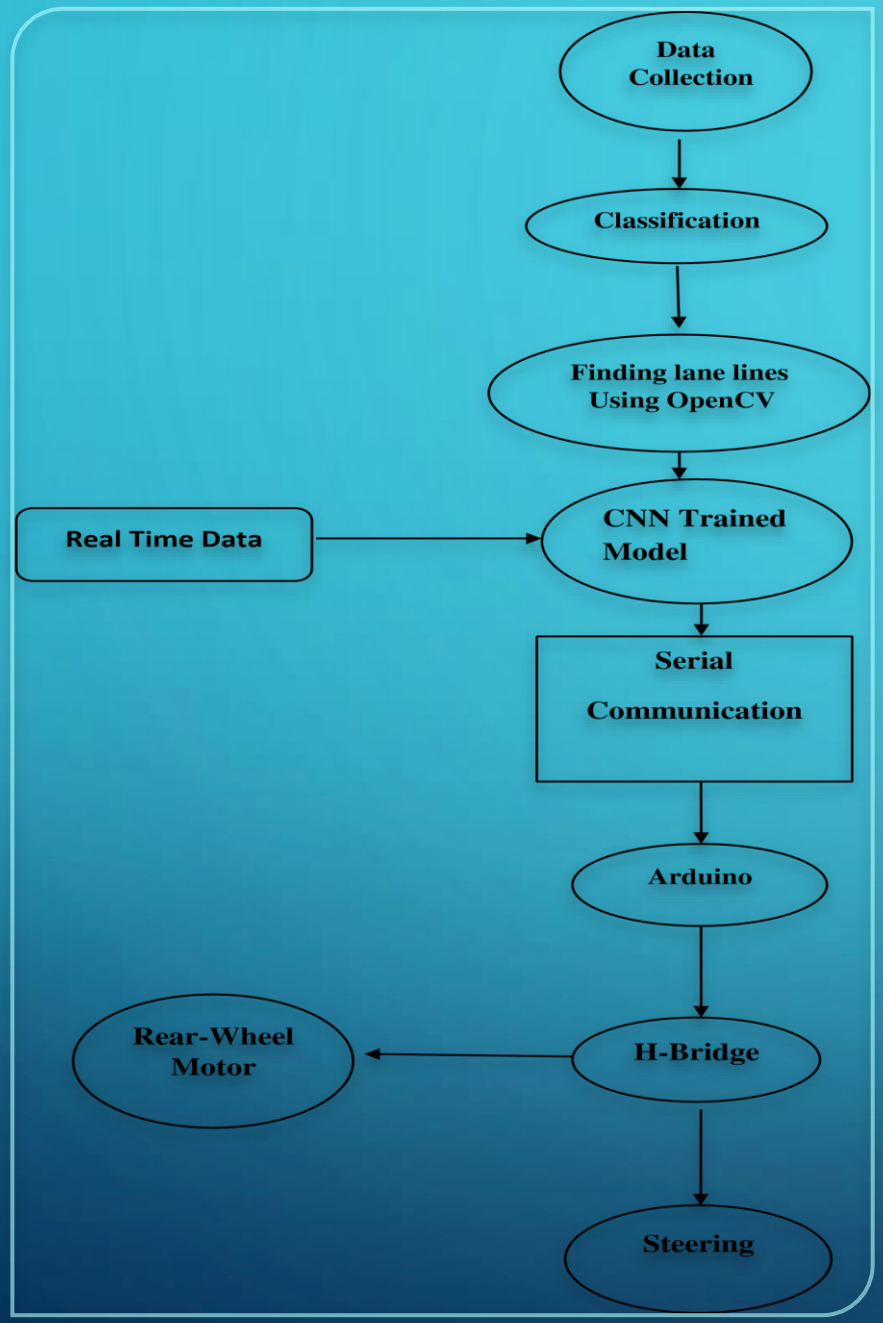
In this project we tried to covert a children's toy electric car into a self-driving car, using the following techniques:

- Computer Vision
- Deep learning methods ie- CNN

Self-driving car made in this project is able to navigate the track by making prediction using the trained data set with the help of CNN model. Before feeding the data to CNN model for training, it is pre-processed using computer vision techniques such as **Gray Scale, Gaussian blur, canny-edge detection, bitwise AND operator and Hough transform**. The preprocessing is done to identify the lane line on track on which car has to move. Initially, tracks are deployed on the ground in order to gather the data in a form of videos using OpenCV with webcam interface. From these videos, images are extracted for classification of the data into four different classes i.e. right, left, forward or stop. Before feeding the data to neural network model, Hough transform is applied using OpenCV for finding the lane line. This data is trained using Convolution Neural Network (CNN) model and a classifier is set which is able to predict in real time whether to move the steering of the car left, right, forward or stop accordingly.

Project Objectives:-

- Development of working prototype of a self-driving using a toy car that will mainly navigate using computer vision and deep learning techniques
- Usage of Convolutional Neural Network to identify a stop sign



Workflow & Methodology:-

The block diagram of project is shown in figure 1.5. Initially, tracks are deployed on surface in order to gather data through video streaming through a webcam using OpenCV which is an open source computer vision library. After the collection of data, the video was segmented into frames and classified into four classes i.e. right, left, forward and stop. This classified data is converted into required format using computer vision algorithms such that the data consists of only bright Hough lines on a black image.

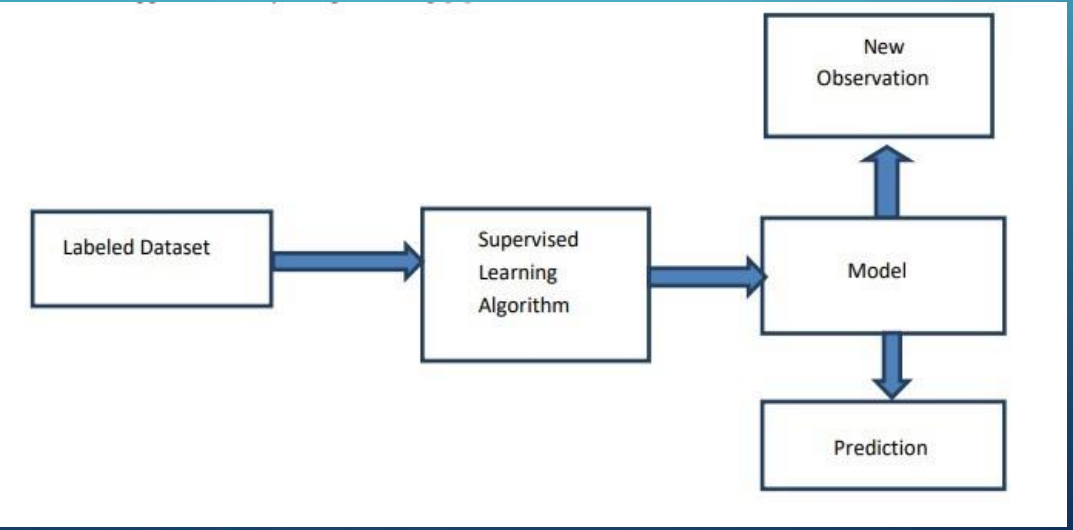
After pre-processing, the data is feed to a CNN model for training. The training and inference is done using core i5 laptop and 1070 core i7 system. Training data used in our project is about 70 percent of the complete data. Supervised learning is used for training of the data. This data is classified and labeled as Right, left and forward. This data is trained using CNN sequential model. CNN model used for the training of the data contains 15 hidden layers. These layers include dense layer, convolutional-2D layer, maxpooling-2D layer, flatten layer and fully connected layers. CNN is used for extracting the features from the images and learn through these features by updating 15 the bias and weights of the perceptron. Categorical cross entropy with Adam optimizer and a learning rate of 0.001 is used in this model. The trained model then takes the input images from live camera and predicts which direction to choose or stop. The trained model after prediction generates a string and through serial communication the string is sent to Arduino. Finally, the Arduino processes the wrappers embedded in its code according to the string received from the trained model and sends control signals to the H-bridge to drive motors of the car to move or stop according to the prediction.

Project Preliminaries:-

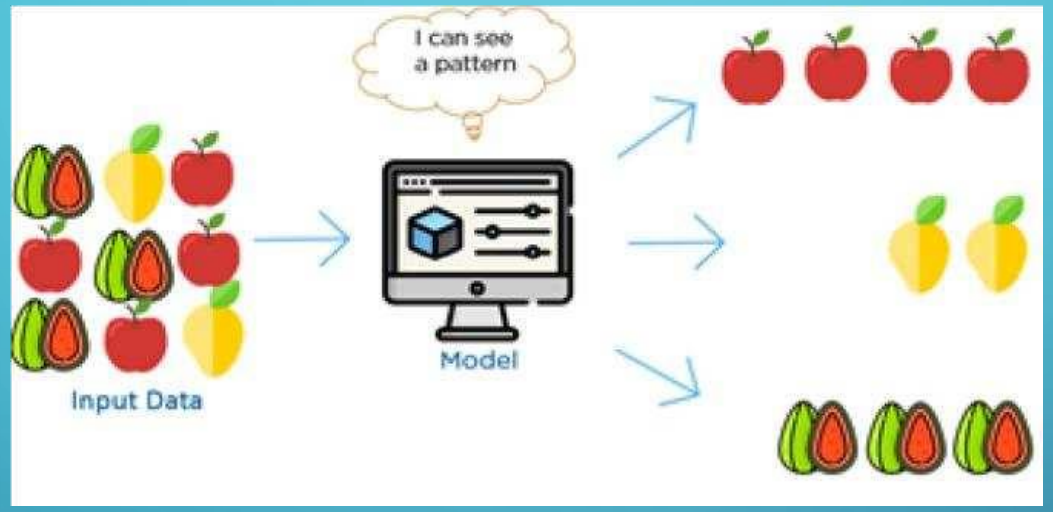
1. **Machine Learning-** Machine learning uses algorithms to describe and analyze data, learn from it, improve and predict useful outcomes. Mostly, it is not possible to directly program a computer to perform specific tasks such as driving car, speech recognition and object detection are way too complex to just program whereas, machine learning algorithm can learn and improve, based on experience. It also interacts with environment to learn, to detect and predict meaningful patterns to achieve desire results.

There are 2 types of Ml algorithms ie- Supervised learning, Unsupervised learning

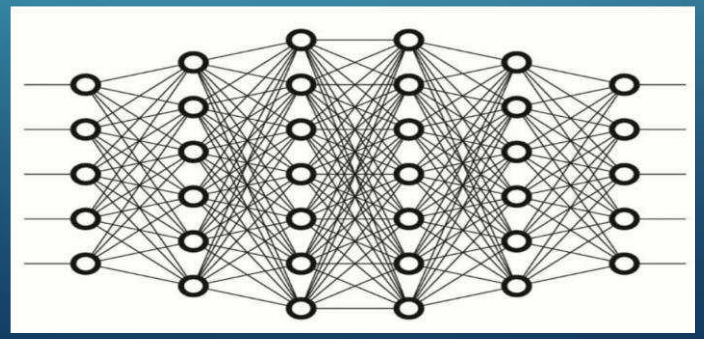
Supervised Learning- Supervised learning is the most popular machine learning technique that uses date to teach the algorithm what conclusion it should come up with. It typically begins with a dataset associated with labelled features that define the meaning of the data and find pattern 25 which can be applied to analytical processing



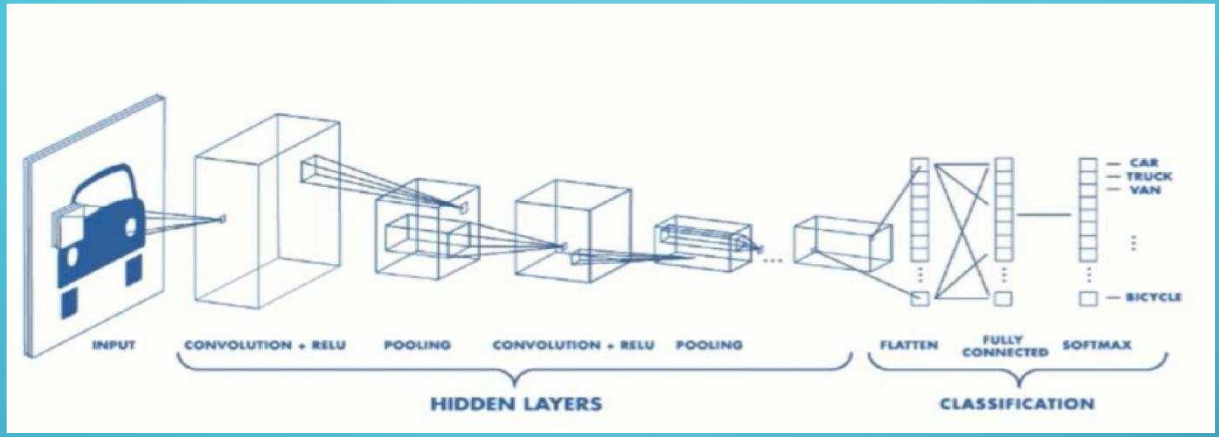
Unsupervised learning- Unsupervised learning is another machine learning approach that occurs between the learner and its environment. In this process, a learner gets a large data set with no labels and the learner task is to process that data, find similarity and differences in the information provided and act on that information without prior training.



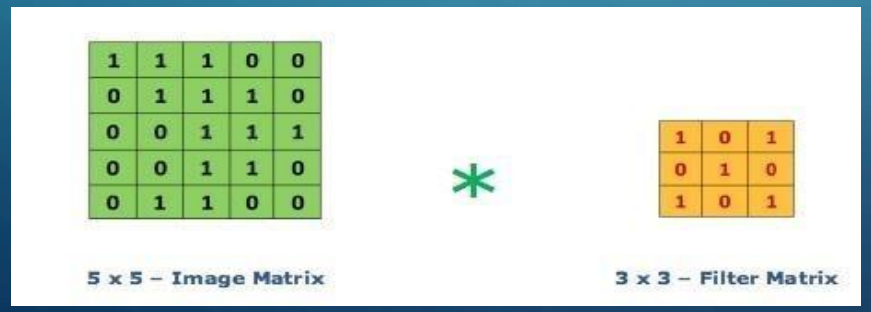
Deep Neural Network- In dealing with complex data, we require a deep neural with much higher capacity to learn. A deep neural network consists of several hidden layers interconnected to each other.

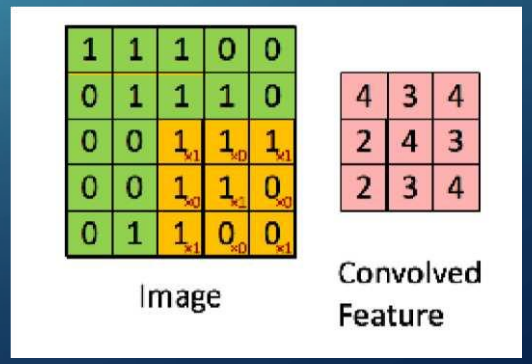


Convolutional Neural Network- This is the most popular and most successful neural network architecture for deep learning. CNN has the exceptional ability to extract important and distinctive feature from images for each class.



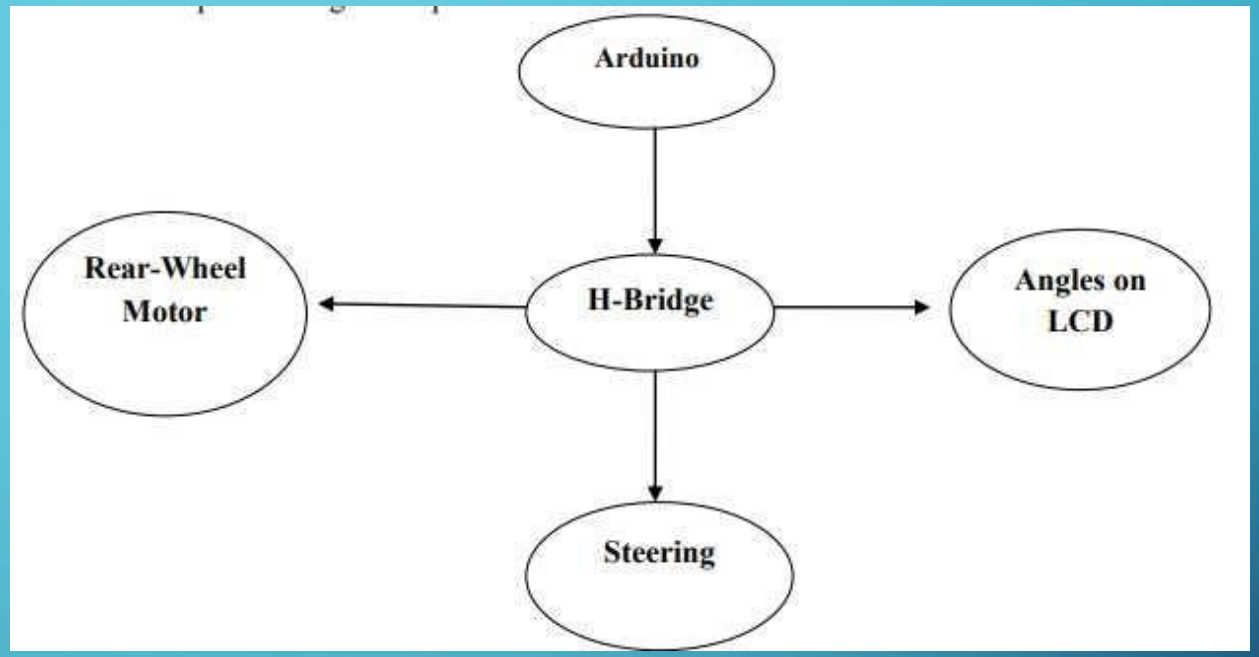
Convolutional layer is the first layer to take out features from an input image. Convolution conserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a kernel.
Consider a 5 x 5 image matrix whose pixel values are 0 &1 and kernel of 3 x 3, Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 kernel, which is called "Feature Map".





HARDWARE-

- Toy Car
- Rechargeable Battery
- Laptop as Main Processor
- H-bridge motor driver
- Arduino Mega 2560
- LCD
- Webcam
- Training tracks

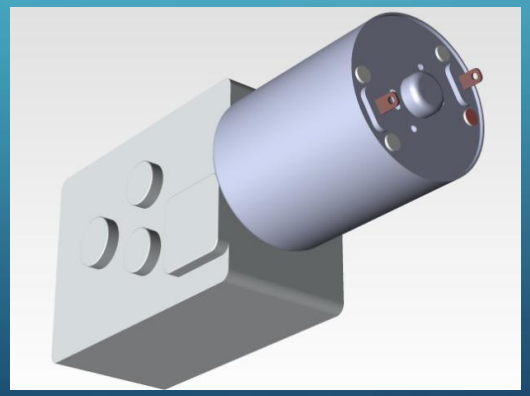


The trained CNN model after prediction generates a string and through serial communication the string is sent to Arduino. Finally, the Arduino processes the wrappers embedded in its code according to the string received from the trained model and sends control signals to the motors drivers, which further controls the motors of the toy car to move or stop according to prediction of trained model

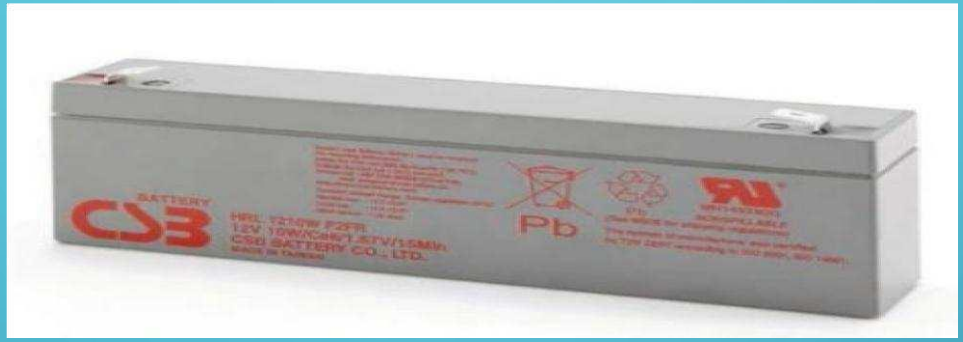
TOY CAR:-

We have used a toy car in our project since as a prototype self-driving car. It has two main motors, one is for steering control and the second motor is rear wheel motor. It has a rechargeable battery for its power source. All the other instruments that the toy car came up with, are removed and 31 assembled with Arduino, motor driver and variable resistor on the steering which is further connected with Arduino and LCD display for measuring and display of steering angles.

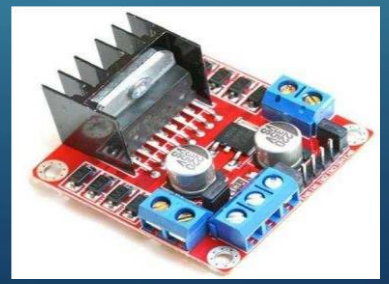
Motor:- Toy Car DC motor is used for controlling steering and rear wheel with assembly. It takes 12V DC voltage from battery and current of 1.6A and it has a resistance of 3 Ohms. Since, the values of current and voltage are known so we can calculate its power that is approximately 20 Watts. Second motor which is used to move steering is also a DC motor. It takes 12V and a current of 1.2 Ampere and has a resistance of 5 Ohms. It gives power of approximately 14 watts. For driving these motors, 2 H-bridge motor driver has been used



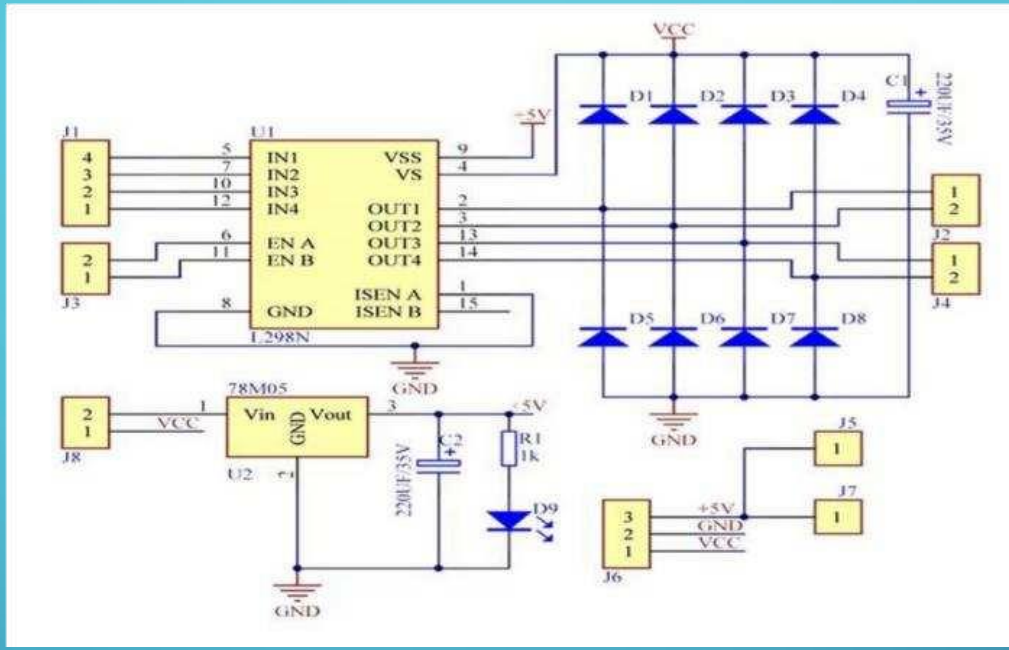
Rechargeable Battery:- The car consists of a rechargeable battery of 12 volts which has a rating of 2.3 AH. It has 12 volts output and can supply a maximum of 4A current with initial current of 0.69A. To recharge the battery we are using a smart charger. Use of these batteries are efficient, saves time and good for environment as Rechargeable batteries produce less waste because they can be recharged with a simple battery charger and reused hundreds of times.



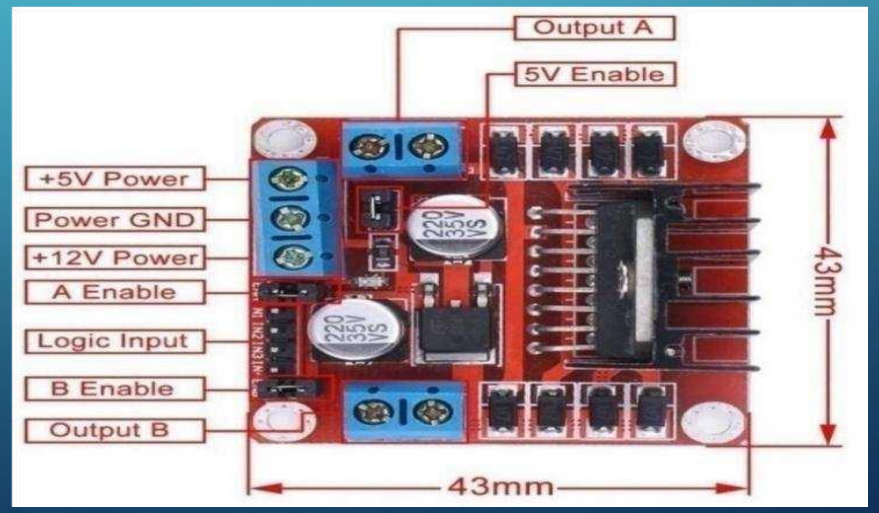
H-Bridge Motor driver:- An H-bridge is an electronic circuit that reverse the polarity of an applied voltage to a load [15]. H-bridges are usually available in the form of integrated circuits. An H-bridge motor driver can be built with four switches (solid state or mechanical) [15]. Only two switches are closed at a time and two switches are in open state. The positive voltage as well as the negative voltage can be applied to allow motors to operate in forward or backward direction. A bipolar stepper motor is invariably driven by a motor controller containing two H-bridges.



H-Bridge L298N Schematic:-



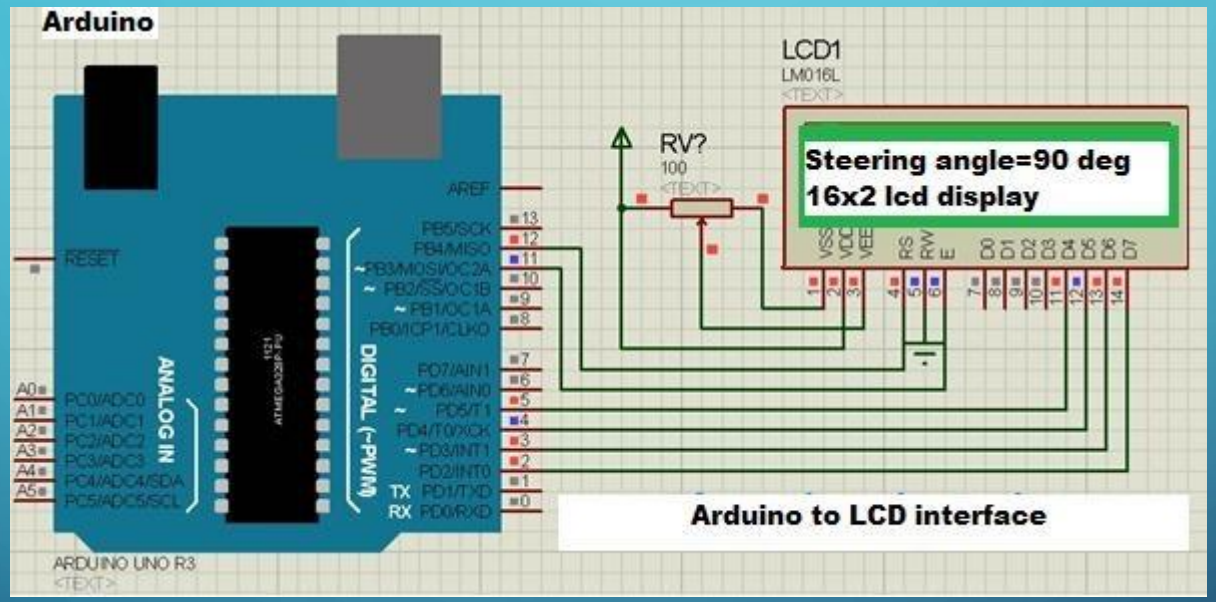
H-Bridge Pin Configuration-



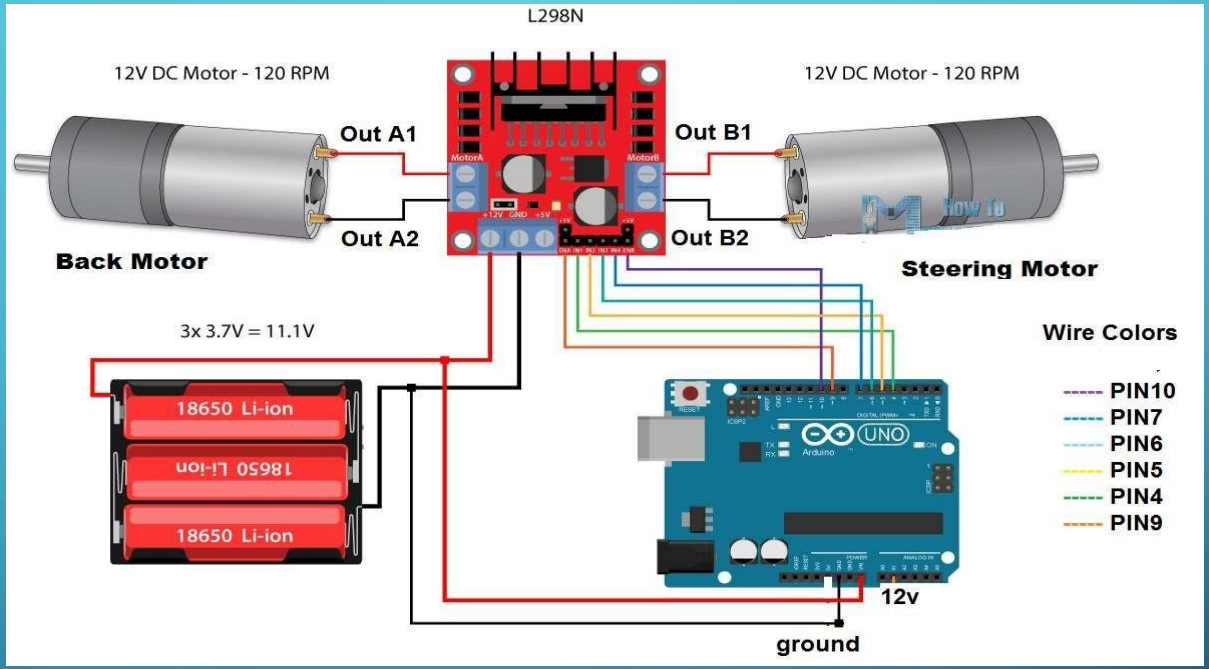
Arduino Mega 2560:- The Arduino Mega 2560 is a micro-controller board. It has 54 digital input and output pins. The 15 pins can be used as PWM output, 16 analog input pins, 4 UARTs (hardware serial ports pins), a 16 MHz crystal oscillator, a USB connection, a power jack, an ISCP header and a reset button. We give a supply of 5V to “Arduino” by using the Arduino power cable.

We have used Arduino to calculate the steering angle, to control the DC motors with H-bridge motor driver and for serial communication with python code.

Steering Angle:- Steering angle is calculated using an Arduino board with the help of variable resistor. Direct connection is established between potentiometer and steering motor such that when the steering motor operates, it causes a change in potential in the variable resistance. This change is calculated in terms of steering angle by making a program for its conversion from voltage to angle. After calculating angle it can be easily displayed on an LCD



Motors Controlling:- Speed of motor can be controlled by using Arduino Mega 2560. For controlling the speed, the input voltage is varied using a PWM signal which is given to the H-bridge motor driver. When switches 1 and 4 are closed and 2 and 3 are open then the voltage flows from the supply to the motor through motor driver module and the motor will rotate in positive direction and vice versa.



Serial Communication:- For interfacing with the image processing python code running on a laptop, we have used “Serial Communication”. By this we can interface between python code and Arduino code. In our project, we are using a signal from python code to Arduino; after receiving signal, the arduino decides to perform an action accordingly. For example, in our case, the python code has to decide the line path using a webcam , it will predict depending upon the track and then that signal will pass to Arduino through serial communication

LCD:- Liquid crystal display is a screen with electronic display module and a wide range of applications which can display 16 characters per row and there are 2 columns i.e. 16 x 2 [16]. In LCD each character is displayed in size of 5x7 pixel matrix. It consists of mainly 2 registers command and data

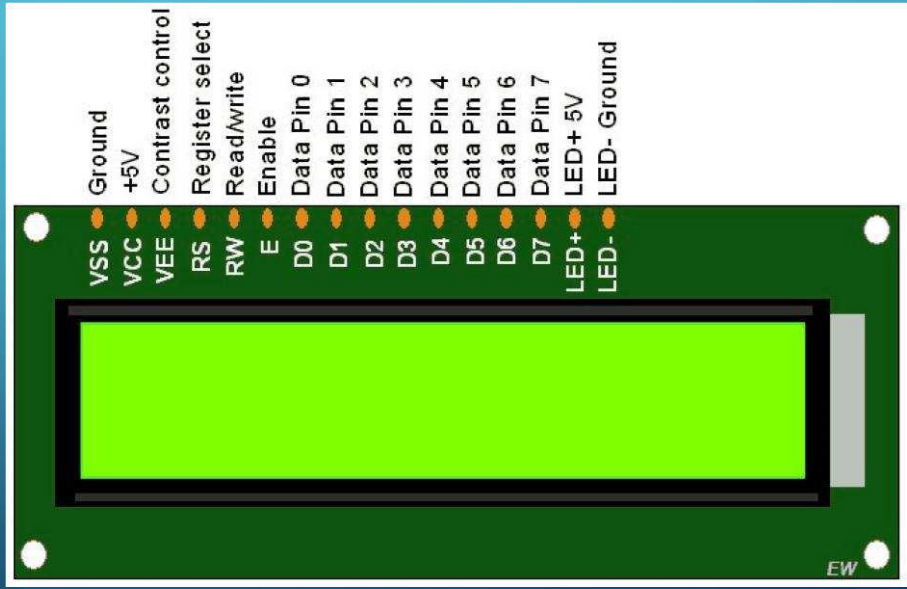


Figure 4.10: Liquid Crystal Display [4]

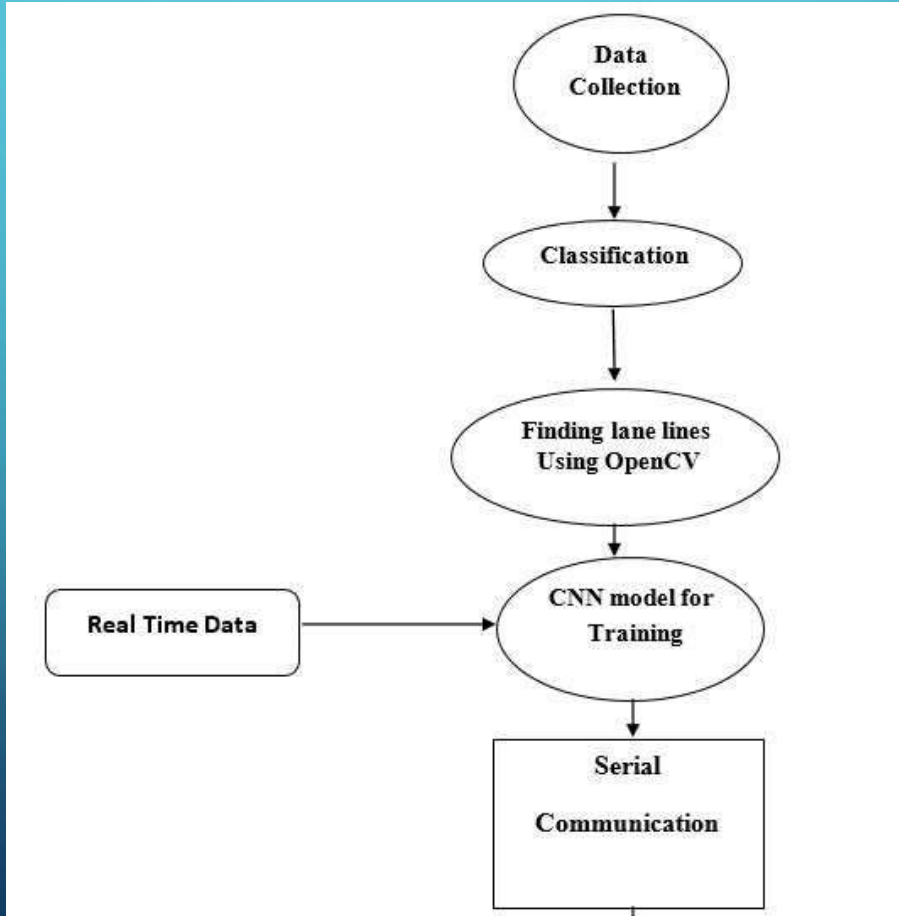
Laptop:- We are using a laptop as main processor instead of Raspberry Pi due to its limited processing speed.

Webcam:- We have used a webcam of Logitech C930e as shown in figure 4.11. It supports H 264 with scalable video coding and UVC 1.5 encoding to minimize its dependence on computer and network : Figure 4.10: Liquid Crystal Display [4] 36 resources. It captures video at 1080p HD quality. Webcam is used for making data set for its training. It is used to capture pictures of track and also used for capturing video in real time.

Training Tracks- We have used different tracks for data sets and training models. With the help of webcam, images of track from prototype of self-driving car are captured to be trained using convolutional neural networks.

SOFTWARES-

Development of prototype of a self-driving car is a big project. This project requires a lot computation power and techniques such as deep learning, computer vision, etc. Different OpenCV algorithms for preprocessing of data i.e. data collection (videos), video to frames extraction, plotting of frames for region of interest, classifying data, frames to video conversion are used. After the pre-processing of data, model is trained up to certain epochs for achieving higher accuracy at very low learning rate



COMPUTER VISION:

Computer vision is a discipline, which is concerned in building artificial intelligence systems in order to obtain information from images or multidimensional data. With this new emerging technology machines are enabled to see things as humans can in real world and it makes machine capable of making decision on its own. Computer vision strong algorithms and its modules are being used by millions of people worldwide for the sake of building new application which could make life of people easier and safer. Computer vision has many domains and people can used it any field of interest, like self-driving cars, virtual reality, cyber security and business, etc. As this is about prototype of self-driving car, computer vision has a lot of significance in this project. Computer vision robust algorithms are used in this project as follows:

- Gray scale conversion
- Gaussian blurring
- Canny edge detection
- Hough line transform
- Optimization

Convolutional Neural Networks

Convolutional neural networks is a class of deep neural networks and is used to analyze imagery. CNN contains multilayer perceptron referring to fully connected layers. These layers contains neurons and every neuron is connected to all neuron in next layer. Convolutional neural networks works in hierarchical pattern and it has capability to understand more complex patterns and solve these complex patterns using simpler patterns. It is so robust and peerless technique, with which any kind of complex data can be classified. CNN models trains itself up to certain epochs, by adjusting weights and biases using back propagation technique for predicting with higher accuracy. If model over-fits, there are certain simpler techniques with which overfitting can be avoided i.e. reducing number of perceptron in respective layers, reducing learning rates and reducing number of epochs for training in order to adjust weights and biases etc

The following CNN developments are as follow:

- **Importing Necessary Packages**

Such as, SkLEARN, Keras, matplotlib, numpy, cv2, os etc

- **Initialization and Sorting of Data**

data and labels are initialized with an empty array for placing the data accordingly and labelling the data. Data and labels are sorted in such a way that images path is grabbed and shuffled randomly. These paths are listed and sorted for reading images in a sequential manner

- **Preprocessing of Data and Labels**

In pre-processing of data, frames are obtained from their respective paths, resized and appended to data. These images are resized to reduce computational power of the system. If resizing is not done, then processing of larger sized images could take longer time effecting performance of system during prediction

- **Scaling of Raw Image Pixel Intensities**

Scaling of raw images pixel intensities is the procedure of normalizing pixel intensity values. Scaling is done in order to create an array of images with pixel intensities between 0 and 255. Afterwards, labels are managed in a single Numpy array vis a vis data. As data and labels are in one dimensional array so data points are recognized by labels in array at same level in correspondence to each other

- **Splitting Training and Testing Data**

Testing/Training splitting of data is be done using sklearn keras library. Testing data is usually 20% of total data and training data is about 80%of total data. Training model is validated using testing data in order to measure how accurately model is trained. But in this project 25% of total data is used as testing data and training data is about 75% of total data

- **Binarizing Label for Training and Testing Data**

Labels are binarized for testing and training. Training labels are binarized to fit model. And transforming testing labels is to normalize label between 0 and 1. Normalization transforms nonnumerical labels to numerical labels

- **CNN Model for Training of Data**

Convolutional neural networks is used to train model. This model contain 15 layers and a sequential model training pattern is followed. In this model five convolutional layers are imported in which a number of parameters are required. Convolutional layers requires number of perceptron, kernel size, subsample, input image dimensions and activation function etc. This model contains dense and dropout layers. Dense layers are the fully connected layers used in model for processing and connecting model to output layer of model. Dense layers requires number of perceptron and activation function etc. For compilation of model, Adam optimizer is applied at low learning rate of 0.001. Mean squared error loss function and matrices for accuracy are also used for compilation of model.

- **Model Summary & Model Evaluation:**

Model is summarized by validation against loss and accuracy level of training and testing data. Model summarization requires fitting of model specifying number of parameters. Model fitting requires training and testing data, steps per epoch, number of epochs, verbose and shuffling of data accordingly.

Model evaluation is also necessary procedure and very easily be done. An integral part of model deployment is model evaluation. Model evaluation helps us to find best model which could represent our data and way it performs in future

- **Plot for Training Accuracy and Loss**

Plots are created and validated for loss and accuracy against validation data. Validation plots are displayed for training and testing data accuracy/loss functions in order to ensure any changes required in model. Model changes with respect model overfitting/under-fitting and to increase trained model accuracy by varying a number of parameters

- **Model Saving and Label Binarization to Disk**

Trained model is saved to be imported and to be used in program at run time. Trained model is saved along with its binaries labels. Such that when they are called in another python program so they can perform accordingly

Serial Communication between Python and Arduino

The steps are following-

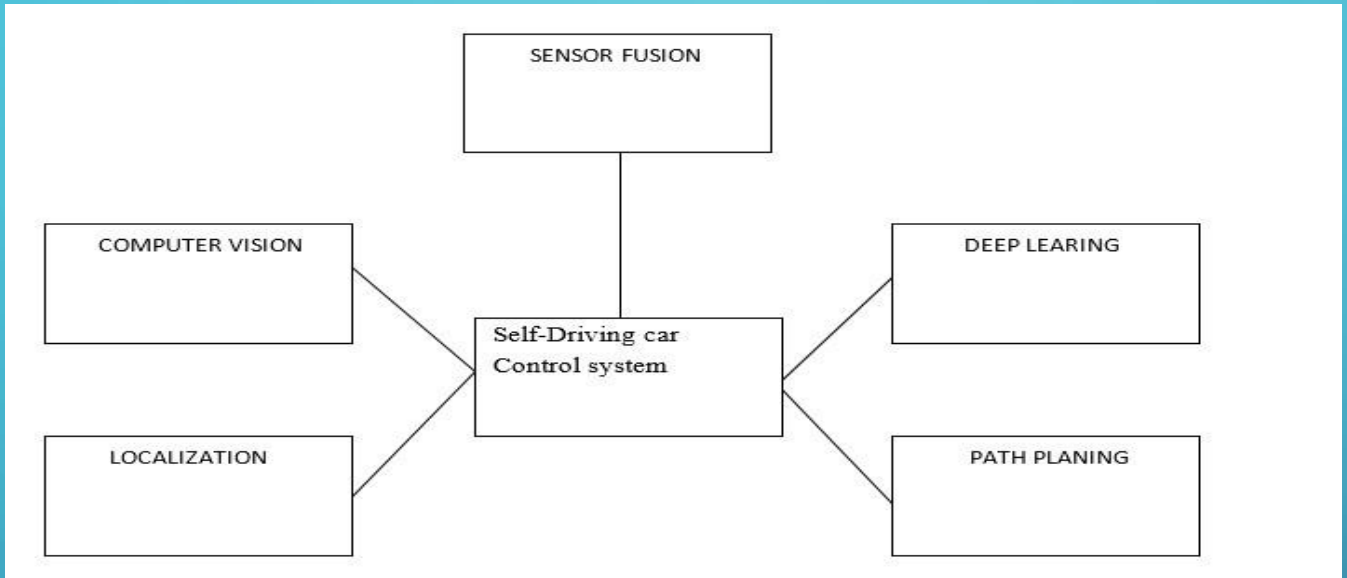
- Importing Libraries
- Arduino Port Detection
- Loading Trained Model for Pickling
- Prediction from Different Track Images
- Finding Labels According to Prediction
- Labels for Multiple Directions
- Writing Text Data and Labels on Frames

The Last and most important step is-

Arduino Controller Working According to Prediction

Arduino controlling and working section is so efficient and serene. Arduino operations are controlled by trained model prediction. As trained model predicts, so Arduino has to perform accordingly. Same string codes are set in both region side by side, firstly on Arduino functions and secondly string type labels encoded for different classes of images. Arduino functionality includes calculating steering angle, displaying steering angles on the LCD display and functions defined for right, left and straight to whom has to work depending on prediction of trained model. Trained model generates a string according to prediction of data fed into trained model as input. That particular string as a command is sent to Arduino using Pyserial communication.

Image Processing and Deep Learning Techniques used in the Project



self-driving car control system is the main laptop processor. This processor enables processing of all actions of self-driving car. All steps involved in maneuvering of self-driving car are directly linked with control system. And all of these sections are linked with each other through control system. Control system is the main core of complete system which controls all actions performed using computervision, deep-learning, path-planning and localization

DATA COLLECTION

Data for our project are the images of the track that is deployed on the surface on which the selfdriving car has to run. Tracks are established using small pieces of PVC pipes as a boundary and of white chart paper in middle of the track to segment it into two lanes. After that we placed our toy car on the start of the track with laptop placed on it and webcam mounted on the front bonnet of the toy car. For data collection we move the toy car on the track using Bluetooth control HC05, which is a Bluetooth module controlled with the help of Arduino and commands of right, left and forward was given using an android app. To operate the webcam, we have used OpenCV running on a laptop to capture the images of the track

- Classification-

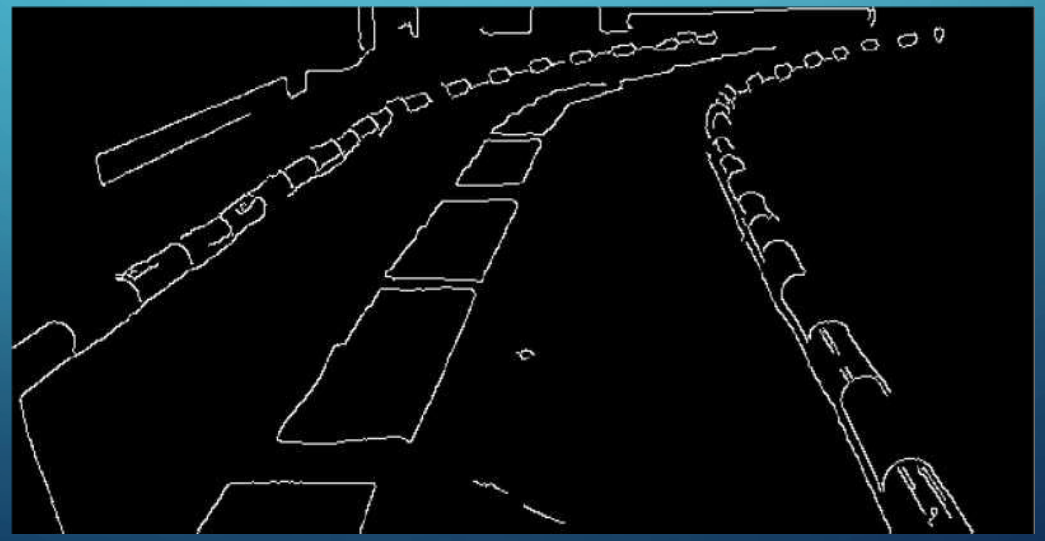
After the collection of video, the videos are converted into the frames using OpenCV. After conversion of videos to frames, the collected data in form of frames are classified using supervised learning before processing of data. Collected data is classified into three classes of forward, left and right. Data collection is vital in order to train the model as accurate as possible. All classes forward, right and left must have equal data in number and normalized in magnitude.

OPERATIONS FOR FINDING LANE LINES

- Before feeding the data to neural network model it is pre-processed using OpenCV to find lane lines for self-driving car. There are number of image processing operations required to tune for finding lane lines as follows:
- Gray scale conversion Gaussian Blurring
- Canny edge detection
- Region of interest
- Bitwise AND operation
- Hough Line Transform
- Display lane lines

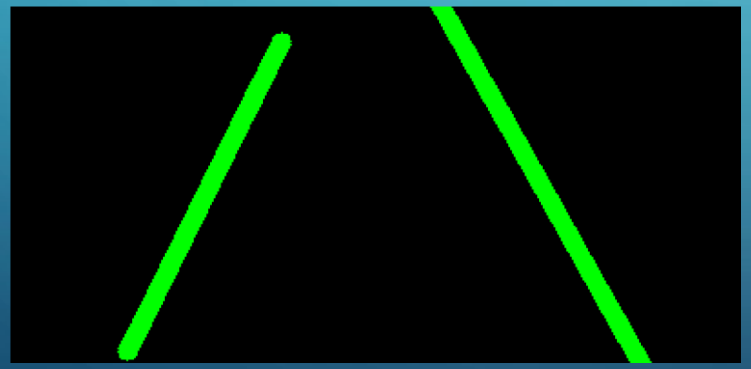
Gray scale conversion Gaussian Blurring

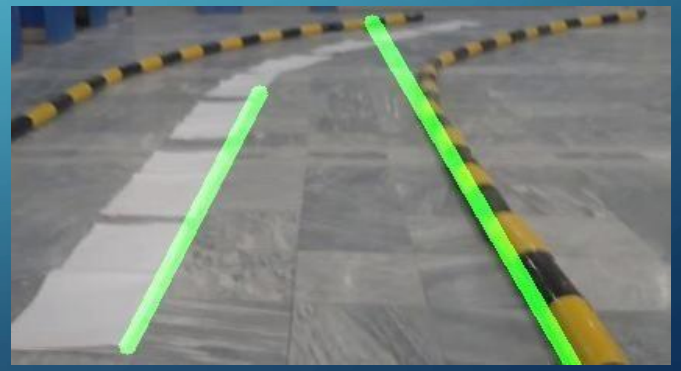
Canny Edge Detection-



Bitwise AND Operator- This technique is used to calculate the per-element bit-wise conjunction of two arrays or an array and a scalar. It is specifically used to only show the specific portion of the image and everything else masked

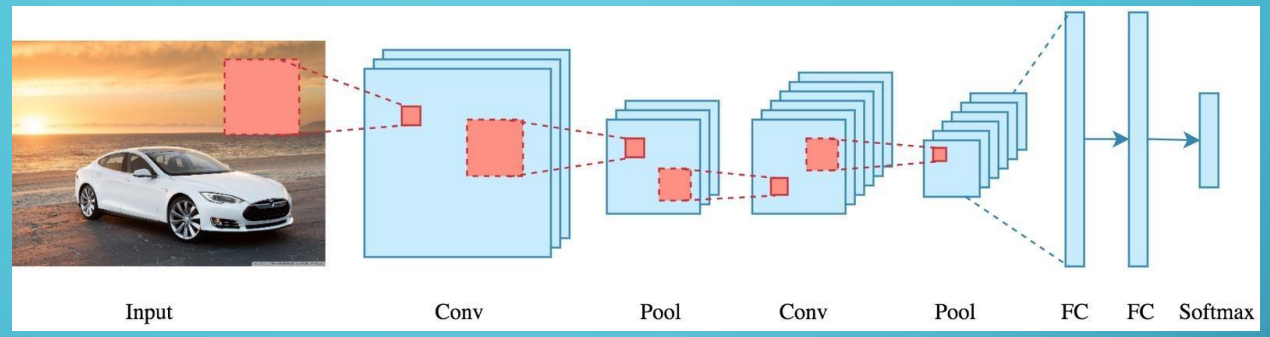
Hough Transform- Hough transform is a feature extraction technique used in image processing. The purpose of this technique is to find the points on the image with same threshold and create lines on them, This technique has ability to join the points having gaps on the image and same features. Lines are displayed on the images using OpenCV. Initially the Hough lines are drawn on a zero pixel image using bitwise AND operator inside the region of interest. Weights of Hough line transform image and the real image of the track are added. By adding weights of these images, the Hough lines are displayed. These displayed lines are then averaged according to their slopes and intercepts in order to display the lines in equal ratio.





CNN Architecture- Convolutional neural networks is a class of deep neural networks. Convolutional neural networks contains multilayer perceptron and are capable to understand complex patterns. A complete CNN architecture. It contains number of building blocks for creating neural networks model which are as follows:

- Data Pre-processing
- Input Layer
- Hidden Layers
- Output Layer
- Model Evaluation.



CNN Hyper Parameters- Convolutional neural networks techniques are embedded in prototype of self-driving car. Convolutional neural networks model used contains fifteen layers in it. The Hyper parameters which can be tuned in this model are as follows:

- Convolutional and Pooling Layer Parameters
- Add/Subtract and sequencing of convolutional and pooling layers
- Number of fully connected layers and their parameters
- Model compilation parameters
- Model evaluation parameters

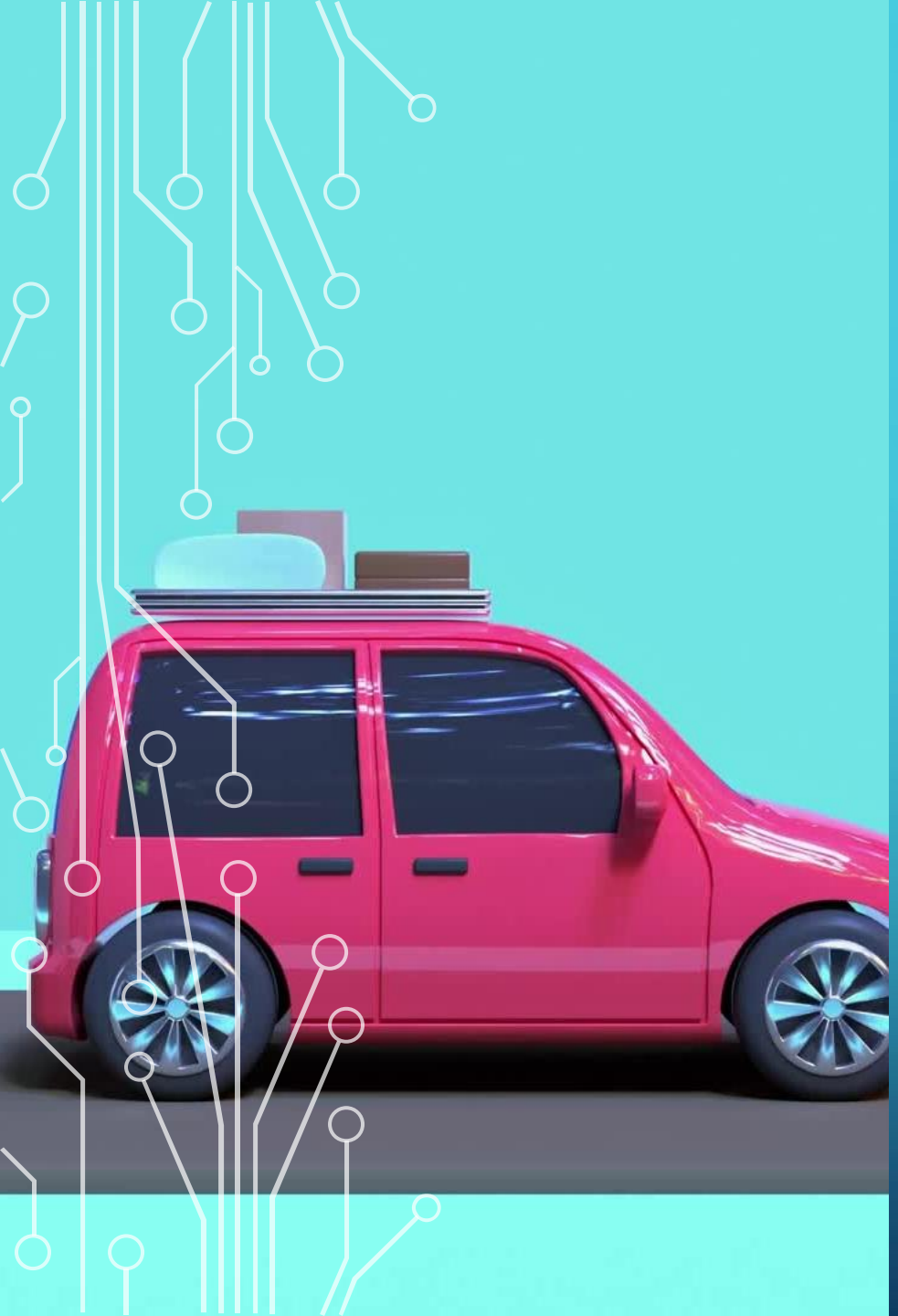
Training of data- Training data used in our project is about 70 percent of the complete data. Supervised learning is used for training of the data. This data is classified and labeled as Right, left and forward. This data is trained using CNN sequential model. CNN model used for the training of the data contains 15 hidden layers. These layers include dense layer, convolutional-2D layer, max pooling-2D layer, flatten layer and fully connected layers. CNN is used for extracting the features from the images and learn through these features by updating the bias and weights of the perceptron. Categorical cross entropy with Adam optimizer and a learning rate of 0.001 is used in this model.

Testing using trained model- A training data set consists of inputs each input corresponds to some label to train the model. The CNN learns from this data, update the bias and weights accordingly and come up with a trained model. Test images are the new data that the neural network has never seen before. The difference between test image and training data is that the training data is used to train the neural network which keeps on updating until the error is minimize. So when a neural network model is introduced to test image which it has never seen before, the model should be able to classify correctly.

Serial Communication- Interfacing of Arduino with python code running on laptop is done with the help of "Serial Communication". The trained CNN model after prediction generates a string and through serial communication the string is sent to Arduino. So, according to the prediction made by CNN model the Arduino coded function is called which moves the car according to the prediction of model. Arduino is further connected to motor drivers which moves accordingly. With the movement of steering motor the attached variable resistor also moves which is further connected to Arduino and LCD to display the steering angle.

CONCLUSION

In this project, a toy car has been used for making a prototype self-driving car. The basic concept behind a self-driving car is to sense its environment and take actions accordingly. The steering angle of the car was measured using a potentiometer mounted on the steering wheel. Movement in the steering wheel caused change in the potentiometer voltage which was then digitized for subsequent processing. An LCD has been used to show the value of the steering angle. The motors used to drive the car were controlled using an H-bridge motor driver. The signal to the motor driver card was given through the Arduino board using PWM. The Arduino board gets the signal from a laptop which runs the computer vision algorithm to navigate the car. Convolutional Neural Networks (CNN) has been used to train data to manoeuvre the car on pre-defined paths. CNN model is deployed along with the powerful computer vision techniques to enable the car to navigate autonomously.



FUTURE PROSPECT OF PROJECT

- Autonomous vehicle is the future
- We can develop a 1:1 prototype and can understand Indian road condition
- We can also use the computer vision technology to decrease the accident on Indian Road.
- Connected car can be developed further from this project

ADVANTAGES

- More efficient driving
- Less accident on road
- Connected car features which lead to better relative driving experience
- Better traffic management

DISADVANTAGES

- Costing
- Indian road condition
- Bad traffic and pedestrian management

ANY QUESTIONS?

THANK YOU