Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

Convolutional Neural Network based Working Model of Self Driving Car - A Study

Chaitra P G
Information Science And Engineering
Vidyavardhaka College Of Engineering
Mysore, India
chaitrasharma3@gmail.com

Deepthi V
Information Science And Engineering
Vidyavardhaka College Of Engineering
Mysore, India
deepthiv029@gmail.com

Gautami S
Information Science And Engineering
Vidyavardhaka College Of Engineering
Mysore, India
harinianan thc@gmail.com

Suraj H M
Information Science and Engineering
Vidyavardhaka College Of Engineering
Mysore, India
suraj.hm71@gmail.com

Prof. Naveen Kumar
Information Science And Engineering
Vidyavardhaka College Of Engineering
Mysore, India
naveen.is@vvce.ac.in

***Abstract*— A self-driving car is a vehicle that senses its environment and navigates without human intervention and is a high research topic in computer vision that involves various sub-topics and need to be deeply reviewed. To accomplish this, our paper discusses hardware and software components of a self driving car that includes usage of technologies such as Deep learning techniques namely Convolution Neural Networks, YOLO algorithm, Hough Transform Algorithms, Transfer Learning, Canny Edge Detection algorithm. Software components such as Arduino IDE, Raspberry Pi Cam Interface, Open CV, Tensor Flow, Carla simulators and hardware components such as Raspberry Pi 3, Arduino UNO, Pi Camera, sensors like radar, lidar are used to build a prototype of a self-driving car. This paper directs some of the complications in the existing technology and provides a few solutions that can be taken to overcome.**

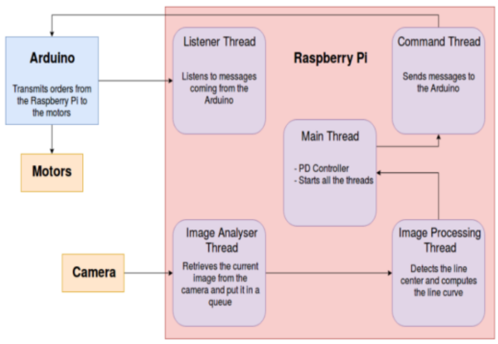
Keywords*— *Convolution neural networks, self-driving car, object detection, traffic sign detection, lane detection.

I. INTRODUCTION

Nowadays, car accidents are increasing worldwide and also driver negligence in following the traffic rules. Such incidents can be controlled by developing driverless cars. The motivation behind self-driving is to improve car safety and efficiency, to prevent traffic accidents, to free-up people's time, to reduce carbon emission, to lower the death rate of children under the age of 18, to save money and fuel consumption. It is a problem under Computer Vision and IoT. The self-driving car should own features such as lane detection, obstacle detection, traffic signal detection, signboard detection, self-parking, detection of humps, pothole detection, accident detection, driving on four-lane etc. Functionalities such as object detection, lane detection, traffic signal detection and signboard detection have been discussed in this paper. All these functionalities can be implemented using Convolution Neural Networks (CNN) in deep learning. Since the single-layer neural network cannot process the complex images while image processing efficiently, CNN is used for image classification.

II. HARDWARE AND SOFTWARE REQUIREMENTS

The hardware components Raspberry pi 3, pi camera and Arduino are used to build a prototype of the self-driving car. The pi camera captures the live images and sends to Raspberry pi for image processing, i.e, to the convolution neural networks, this device is attached at the top of a self-driving car. Arduino board is connected to a computer via USB cable and communicates with Arduino that sends the signals(move right, left..) to the motor as per the result of the neural network through a remote controller. As the signal is triggered the car moves according to it. Fig .1 shows an architecture overview of these hardware components.



Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.

Fig .1.Architecture Overview

1. *Raspberry pi 3*

Raspberry pi is a device that functions similar to a PC and it is a credit-card sized electronic board. It is a small, cheapest and the device does not contain any internal peripherals or storage space. Input, output and storage peripherals need to be attached to the device, an input to the device is given through keyboard or mouse, to display output a monitor is required, to store the data and to reserve an operating system an SD card is used, it also requires a power supply and other required cables. For additional storage hard drive like SD Flash memory can be used. The device supports wireless LAN and Bluetooth connectivity to connect to other devices. Also supports programming languages like Scratch and python[12]. The device contains a Quad-Core CPU of size 64bit and memory of 1GB RAM. The operating system Linux is compatible with this device. It has 4 USB 2 ports, video port, HDMI port and 4 Pole stereo output. Along from all this, for connecting a Raspberry Pi touch screen display a DSI display port, for connecting a Raspberry Pi camera it has a CSI camera port, for storing data and to load operating system a Micro SD port and for power supply it has a power source up to 2.5A Micro USB[16].

Fig .2.Raspberry Pi 3

2. *Arduino UNO*

Arduino is an open-source electronic board which is capable of reading analog or digital inputs from devices like sensors and produces the outputs like triggering the motors. It has a circuit board called microcontroller that can be programmed and software called Arduino IDE, where the code can be written and upload to the circuit board. Programs written on Arduino IDE should be saved as .ino extension, Instructions are sent to the microcontroller through this IDE to control the board functions and it uses a c programming language[12]. The Arduino microcontroller is based on ATmega329P.It consists of 14 digital I/O pins out of which 6 of them are PWM output pins. It has 6 Analog Input Pins and 6 PWM Digital I/O Pins. It has memory 2 KB of SRAM, 32 KB of Flash memory,1 KB of EEPROM and 16MHz clock speed[17].

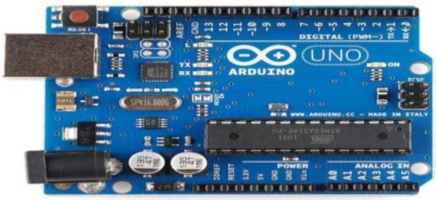


Fig .3.Arduino UNO

3. *Pi camera*

The pi camera is a lightweight portable module, a Raspberry pi is connected to this device through an elastic cord which has a serial interface. This Camera Module can take high-quality video and photographs. The pi camera is placed on top of the autonomous car, the images are captured by this module and are sent to Raspberry pi for preprocessing. It has 5MP color camera module without a microphone for Raspberry Pi, Resolution of 2592x1944 and also supports 1080p, 720p and 480p video quality.

Fig .4.Pi Camera

Software components like Arduino IDE, Raspberry pi cam Interface, OpenCV library, Tensorflow, Carla are used in this project.

Arduino IDE is open-source tools were programmed are written to control the Arduino board. To run Arduino circuit board programs written here are sent to the microcontroller through this IDE and it uses a basic C programming language.

Raspberry Pi Cam Interface is a software where the live images captured by Pi camera can be remotely viewed on a laptop. Live feed captured by Pi camera can be viewed on this interface. Images and videos can be recorded or downloaded in numerous resolutions with the various number of settings.

OpenCV is a cross-platform library where Read or write images, videos can be captured or saved, then images are processed i.e filters and transform the images, performs the feature detection on images, it also detects distinct objects in videos or images like faces, eyes, cars,..etc. Using these platform real-time computer vision applications are developed. It mainly concentrates on video capture and image processing. It supports programming languages like C++, C, Python and Java, it runs on different operating systems like Linux, Mac OS, Windows, Android and IOS.

Tensorflow is an open-source python library where it allows the users to build a deep learning model. Using this library one can Pre-process the data, Build the model, Train and estimate

Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.

the model. Once the models are build and developed they can be used by people for their projects.

CARLA is an open-source simulation environment for autonomous driving research. It supports the development of autonomous cars, to perform different tasks can be trained and also does the validation of autonomous driving systems. It has open digital resources like urban layouts, buildings, vehicles and can be used freely[18].

III. CONVOLUTION NEURAL NETWORK (CNN):

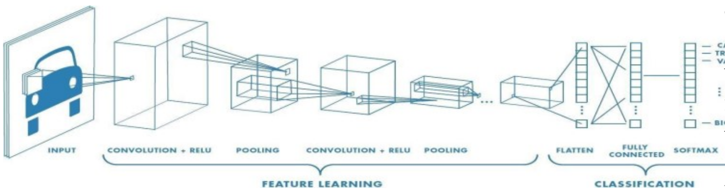


Fig .5.CNN architecture

.

Object detection is an essential component in a self driving car. Humans can sense and recognize the objects around them within a fraction of seconds. The self-driving car should detect and differentiate between objects such as humans, vehicles, traffic lights, roads etc. To accomplish such tasks, CNN is used as a tool for image classification. The idea of CNN took inspiration from the visual cortex of the human eye that is receptive to a precise region of the visual field

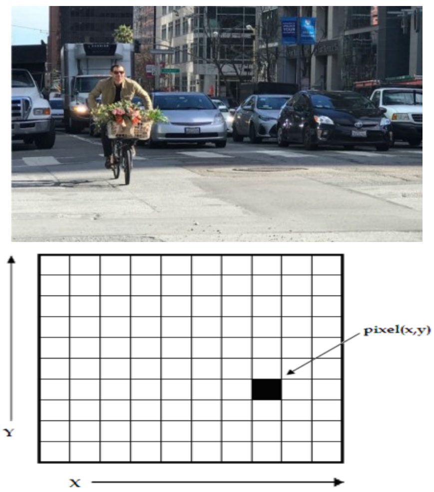


Fig .6.The input image and its interpretation in pixels

Image is shown in "Fig.6," is the input for CNN. The computer reads this image as an array of pixels. The output after classification is the probability that the input image classified as a certain object. By considering edges and curves as low-level features, the computer identifies and recognizes the object within the image.CNN has several layers such as convolution layer, non-linearity layer, pooling layer and fully connected layer.[7][8]

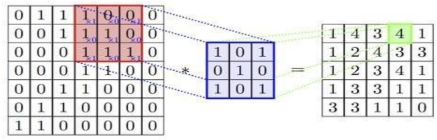


Fig .7.Input image converted into a feature map using a filter.

The first layer in CNN is the convolution layer. Consider the image in "Fig.7," which is of 32X32X3 array of pixels. The input image is fed into this layer. Convolution layer extracts features from the input image by performing the mathematical operation between two inputs such as input matrix and a filter or kernel. The filter is also a matrix containing weights and is unique to a particular class label. These filters are slid over the input matrix and corresponding cells are multiplied and added that results to produce a feature map shown in "Fig.7," [7][4].To reduce non-linearity in the neural network ReLU(Rectified Linear Unit) is used. Consider the feature in the "Fig.8," all the negative values are changed to zero using ReLU. [4][2]

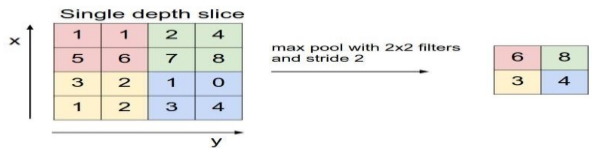


Fig .8.Example for Max pooling of feature map

The dimensionality of the feature map is reduced by pooling technique. The pooling layer extracts the largest element from the feature map by considering 2X2 filters shown in "Fig.9," thereby retain the original information.[2]

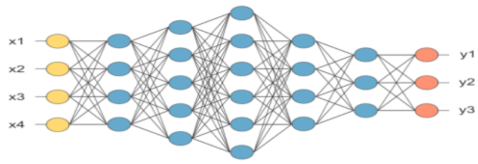
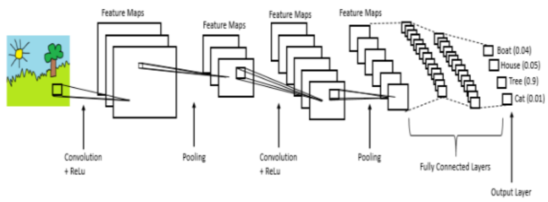


Fig.9.Fully Connected Layer of CNN

Feature map obtained after pooling is fed into the Fully Connected Layer "Fig.9," by converting it to a vector. This layer is responsible for classifying the image to a particular class. The FC Layer compares the vector with the various filters and multiplication, addition operation is performed between them. Different values will be obtained after performing the mathematical operation between the vector and the different filters. The maximum value among them and the filter used for multiplication is identified.[6][7][8].The image is classified to the class that the filter belongs as shown in "Fig. 10,".



Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.

Fig .10.Complete CNN architecture

Detection Of Objects by self-driving car using YOLO (You Only Look Once):

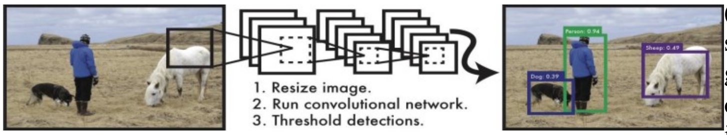


Fig .11.Detecting, classifying and localizing objects in an image by YOLO[1]

YOLO is one among the algorithms that use CNN, regression for object detection. It scans the entire image at once while image processing and hence the name YOLO. It is extremely fast and can process the image at the rate of 45 frames per second [1]. The neural network is applied to the entire image, calculates the bounding boxes and class probabilities. In “Fig. 12,” the entire image is fed as an input to the CNN, YOLO splits the entire image into 13 X 13 grids. Fixed number of bounding boxes, the class probability is predicted for each grid. Each grid cell predicts only one object.[1][9]

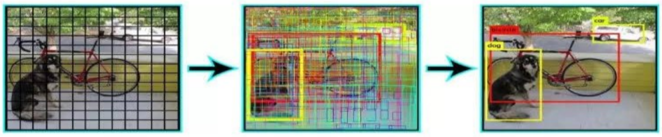


Fig .12.Working of YOLO

Each boundary box is associated with 5 aspects namely confidence score of the box, ‘x’ and ‘y’ representing offsets, ‘w’ and ‘h’ representing width and height of the bounding box respectively. The confidence score of the box determines the probability of the box containing the object. ‘w’ and ‘h’ is normalized by the width and height of the image respectively. Hence all the aspects ‘w’, ’h’, ’x’, ’y’ are between 0 and 1.There will be fixed number of conditional class probability for each cell determining the probability of the object belonging to the specific class. In “Fig.12,” the 13 X 13 grid, YOLO predicted 845 bounding boxes. The bounding box with its probability greater than the predefined threshold is considered for locating the object within the boundary box. In “Fig.12,” the bounding box detects dog, bicycle and car as objects. [6][7][8][9]

IV. LANE DETECTION:

Lane detection is an important substratum in the development of self-driving cars. The road lane detection consists of five major steps removing noise, discard color information, detection edge, a region of interest, Hough transform. The video frame is fixed on the car to detect road lane and to provide other information that can help in taking decisions.[10][11][19].

(a) *Removing noise*: First step in lane detection in the removal of noise using Gaussian Blur method. The captured image from the camera is cleaned up by applying Gaussian Blur method. Noise and small objects which are not pertinent are removed[19]. As shown in below “Fig.13c,”.

(b) *Discard color information:* After removing noise the next step is to convert the image into a grayscale image. The grayscale image is a type of black and white image. By conversion, the image is compressed. For example, if the image is of the size 500*500, the RGB representation is 500*500*3bytes. But when grayscale representation is considered got immediate scope of compression. The result of grayscale and blur will give a black and white image[19]. As shown in “Fig. 13d,”.

(c) *Detect edges:* For edge detection, the Canny edge detection algorithm can be used. To obtain structural information from other objects and effectually decreases the amount of data processed. The canny algorithm detects the edge on the image by gazing for swift changes in color between its neighbors and pixel. The main lane is identified with the help of grayscale and blur steps. Black and white image will be the resultant. The given below “Fig.13e,” is canny edge applied to a grayscale image.[12][19]

(d) *Region of interest:* It is a part of the image to process. It is a very important process. Region of interest is the lane which is obtained by considering the lower half the image since the upper half the image contains other objects (noise).[19] “Fig.13f,”.

(e) *Hough transform:* the basis of the Hough transform is to find aligned points in an image that creates lines. Suppose, for an input image Fig.13f and applied some edge detection algorithm and obtain an image which highlights the main edges of the image and the Hough transform is applied to detect points in the image that creates these lines[11][19][12] “Fig.13g,”.

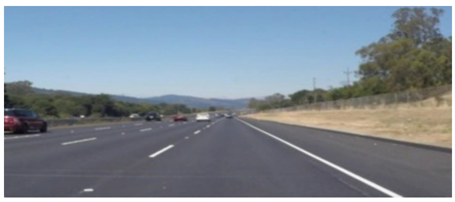
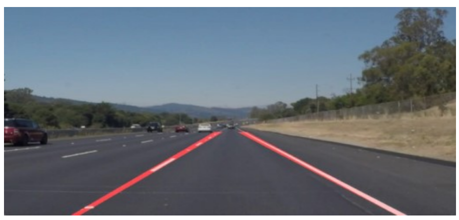


Fig .13a.Original image [19]



Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.

Fig .13b.After removal of noise [19]

Fig .13c.Blur applied to the image [19]

Fig .13d.Grayscale transformation applied to image[19]

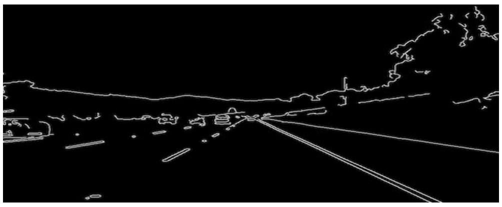


Fig .13e.Canny edge detection applied to grayscale image[19]

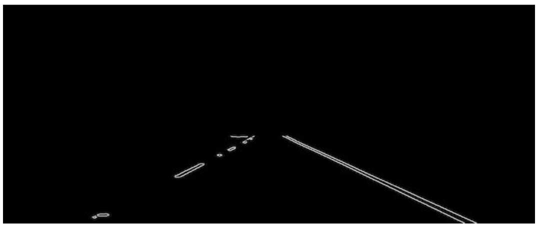


Fig .13f.Region of interest[19]

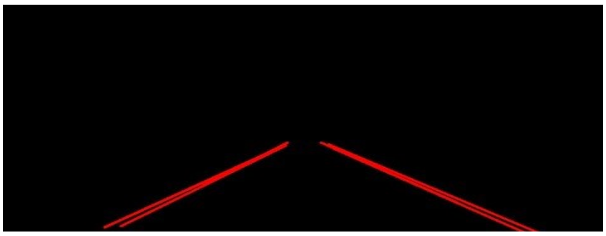


Fig .13g.Hough transformation returns a list of lines.[19]

V. TRAFFIC SIGNAL AND SIGNBOARD DETECTION

Classification and detection of traffic signal and signs are an important part of autonomous vehicles. Automatic Recognition of these traffic signs using intelligent systems can reduce the number of accidents worldwide. The government has established certain rules for safety in the flow of traffic. Traffic signs are displayed along the roadsides and it has to be followed to avoid accidents and these have to be identified and interpreted by an Autonomous car while driving. In this project, an artificial neural network is used for classifying the traffic signal and signs which is trained with real-time datasets. Recognition of traffic signs is carried out in two phases, detection followed by classification. Detection finds or detects the traffic signs in a bounding box of the specific category whereas classification classifies it by giving a class label to an image specifying which kind of sign is present[14]. The images of the road signs are converted to grayscale as shown in "Fig.14b," and these grayscaled images are filtered using simplified gabour wavelets. Region of interest is extracted using the maximally stable External regions algorithm and classified using support vector machines[15]. Finally, convolution neural networks are applied to classify the signs and to extract or crop the specific signs on boards a color based segmentation model is employed[14].

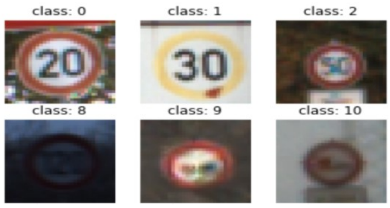


Fig .14a.Image before pre-processing

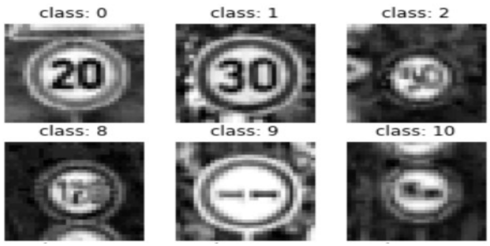


Fig .14b.Grey Scaled image

For stable recognition and detection of traffic lights, Faster R-CNN(Region-based convolution neural network) Inception-V2 model through transfer learning is used[13].
The transfer learning is an approach in machine learning where a learning model which is pre-trained and the knowledge of this pre-trained model is adapted to a different learning task but related one. For example, a model trained to predict an image of cars, this pre-trained model can be used during training to recognize the trucks.

Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.

Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

Fig .15.Traffic Light Images before classification

For training the CNN model, images of Indian traffic lights are collected and these images are pre-processed to generate the accurate dataset. After collecting the training datasets the CNN model is trained for a significant amount of time. After training the model with few iterations it is validated with testing datasets and the model finally strongly detects the traffic lights and classifies according to its type by giving a class label as shown in “Fig.16a,” and “Fig.16b,”[13].

Fig .16a.Image after classification

Fig .16b.Image classified as red light

CONCLUSION

In this paper, a prototype of a self-driving car is conferred. Deep learning algorithms like CNN (Convolution neural networks) is employed to make immediate decisions for the self-driving car. Detection of objects, lane detection and detection of traffic signs and signals have been studied in this paper. The algorithms like Faster R-CNN Inception-V2 model through transfer learning, YOLO and canny edge detection algorithms have been studied.

ACKNOWLEDGMENT

The authors thank and express gratitude towards the aid provided by our mentor and faculty members who guided us in all respects and helped us in achieving the desired results.

REFERENCES

[1] JosephRedmon, Santosh Divvalay, Ross Girshick and Ali Farhadiy,You Only Look Once:Unified, Real-Time Object Detection, University of Washington, Allen Institute for AIy, Facebook AI Research.
[2] Keiron OShea1 and Ryan Nash2,"An Introduction to Convolutional Neural Networks",Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB,School of Computing and Communications, Lancaster University, Lancashire, LA1
[3] Mochamad Vicky Ghani Aziz,"Implementation of Vehicle Detection Algorithm for Self-Driving Car on Toll Road Cipularang using Python Language",Control and Computer System Laboratory, Indonesia.
[4] A Thesis by GUANGRUI LIU,BS,"Real-time Object Detection For Autonomous Driving Based On Deep Learning", Beijing University of Technology, China
[5] BrilianTafjiraNugraha1 and Shun-Feng Su2,"Towards Self-driving Car Using Convolutional Neural Network and Road Lane Detector", Department of Electrical Engineering National Taiwan University of Science and Technology (NTUST) Taipei, Taiwan, ROC
[6] Milan Aryal,"Object Detection, Classification, and Tracking for Autonomous Vehicle", Grand Valley State University.
[7] Tareq Abed Mohammed,"Understanding of a Convolutional Neural Network",SaadAlbawi University of Diyala, Kirkuk University
[8] Rikiya Yamashita,Mizuho Nishio,Richard KinhGian Do and Kaori Togashi,Convolutional neural networks: an overview and application in radiology
[9] Juan Du1,"Understanding of Object Detection Based on CNN Family and YOLO", New Research and Development Center of Hisense, Qingdao 266071, China
[10] Aditya Singh Rathore,"Lane Detection for Autonomous Vehicles usingOpenCV Library",B.Tech , J.K. Lakshmipat University
[11] Fahmizal,"Towards Self-driving Car Using Convolutional Neural Network and Road Lane Detector", Electrical Engineering and Informatics Vocational College University of GadjahMada Yogyakarta, Indonesia
[12] AdityaKumar,"Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino", Jain Electronics and Communication Department Dharmsinh Desai University Gujarat
[13] RuturajKulkarni,ShruthiDavalikar and sonalbangar,"Traffic Light Detection and Recognition for Self Driving Cars using Deep Learning", Dept. of Electronics and Telecommunications
[14] Mohit Singh,"Traffic sign detection and recognition for autonomous vehicles",IMS Engineering College, Ghaziabad, Uttar Pradesh
[15] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6210476/
[16] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[17] https://store.arduino.cc/usa/arduino-uno-rev3
[18] https://github.com/carla-simulator/carla
[19] A lane detection approach for self driving vehicles by @ldesegur
[20] Aditya Singh Rathore,"Indiaane Detection for Autonomous Vehicles using OpenCV Library", B.Tech, J.K. Lakshmipat University

Authorized licensed use limited to: University of Wollongong. Downloaded on August 10,2020 at 06:06:37 UTC from IEEE Xplore. Restrictions apply.