Design & implementation of real time autonomous car by using image processing & IoT

Irfan Ahmad
Department of Electrical engineering
Indian Institute of technology
Jodhpur, India.
ganieirfan27@gmail.com

Karunakar Pothuganti
Department of Electronics and Communication Engineering
Sri Satya Sai University of Technology & Medical sciences
Sehore, India.
karna119@gmail.com

***Abstract*— Because of the inaccessibility of Vehicle-to-Infrastructure correspondence in the present delivering frameworks, (TLD), Traffic Sign Detection and path identification are as yet thought to be a significant task in self-governing vehicles and Driver Assistance Systems (DAS) or Self Driving Car. For progressively exact outcome, businesses are moving to profound Neural Network Models Like Convolutional Neural Network (CNN) as opposed to Traditional models like HOG and so forth. Profound neural Network can remove and take in increasingly unadulterated highlights from the Raw RGB picture got from nature. In any case, profound neural systems like CNN have a highly complex calculation. This paper proposes an Autonomous vehicle or robot that can identify the diverse article in condition and group them utilizing CNN model and through this information can take some continuous choice which can be utilized in the Self Driving vehicle or Autonomous Car or Driving Assistant System (DAS).**

Keywords*— *CNN, Deep Neural Network, Driving Assistant System (DAS), Self-Driving car.

I. INTRODUCTION

With the improvement of different sensor procedures, (for example, LIDAR, millimeter-wave radar, camera, and differential-GPS), more profound adaptive models have come to up investigate on autonomous- vehicles in the previous time. A most dependable answer for dispatch and board of Autonomous Car sooner than later perhaps Vehicle-to-Infrastructure correspondence [1]. In the V2I correspondence framework, oneself driving vehicle can associate with the general condition and with one another legitimately through the vehicular systems [2]. In any case, because of the absence of the V2I framework, finders for perceiving and following vision-based traffic lights, signs, paths people on foot despite everything assume significant jobs in present self-driving frameworks or self-ruling driving colleague frameworks.

Past, PC vision-based arrangements [3]-[6] are profoundly influenced by the situation of cameras, the effect of ecological light conditions, separation of articles [7], preparing capacity of the vehicle chip. In any case, a solitary arrangement of physically fixed parameters dependent on the heuristic strategy isn't adaptable for confused genuine conditions, and recognizable proof of reasonable parameters is additionally tedious. In this manner, investigate is focused on utilizing profound learning procedures [13]-[18] or AI [8]-[12] to prepare the model in an information-driven way. Profound adaptive based convolutional neural system (CNN) model accomplishes preferable execution or results [7],[19],[20] over the AI algorithms dependent on the Histogram of Oriented Gradients (HOG) highlights, (for example, SVM, and the Hidden Markov Model). This is a way that CNN can extricate and take in increasingly unadulterated highlights from the crude RGB channels than old calculations, for example, HOG. In any case, calculation multifaceted nature of CNN models is a lot greater than that of most AI calculations. Along these lines, in this paper, a profound neural system is proposed to utilize distinguish different segment in condition to settle on some significant choice which will be helpful in the field of a self-driving vehicle or Autonomous Vehicle or Driving Assistant System (DAS).

This paper organizes as follows section II describes the literature review, section III gives a block diagram and workflow of the proposed system. Section IV gives information about model for object detection. Section V computer vision algorithm for line detection followed simulation results and conclusion.

II. LITERATURE REVIEW

Autonomous car driving was analyzed and implemented by using CNN. Many methods were proposed and implemented in this section. In [21] authors discussed evaluating the current challenges in artificial intelligence and how these methodologies can be applied to cybersecurity in the USA, including traffic analysis schemes. In [22], this paper gives the advanced driving assistance system that will give past, present and future driving different variants and it is analyzed by using real-time embedded systems. Authors in [23], describes methods for naturalistic data collection and algorithm development based on pre-scan and driving simulator. Along with this in [24], the author describes various information technologies like IoT, AI, blockchain technology, human cloud and robotic techniques. Authors in [25], gave information about frequently used algorithms for computer vision in the field of medical and traffic analysis.
In [26], the authors described computer vision research in the university Waikato. Hyperspectral image formation and digital

line scan photography [26] these type of research areas are analyzed by using computer vision.

III. PROPOSED WORK

The proposed technology for the autonomous car by using CNN is shown in figure 1 and the experimental setup shown in figure 3. This total system has to be fixing on any motor vehicle so that camera has to be in the direction of the road. From camera, input gave to raspberry pi, in raspberry pi python language used to communicate between CNN network and different sensors in the network. Through raspberry pi, total data will be uploaded to the cloud platform. From the website, can be seen where the driving is. This total setup is made and shown in figure 3.

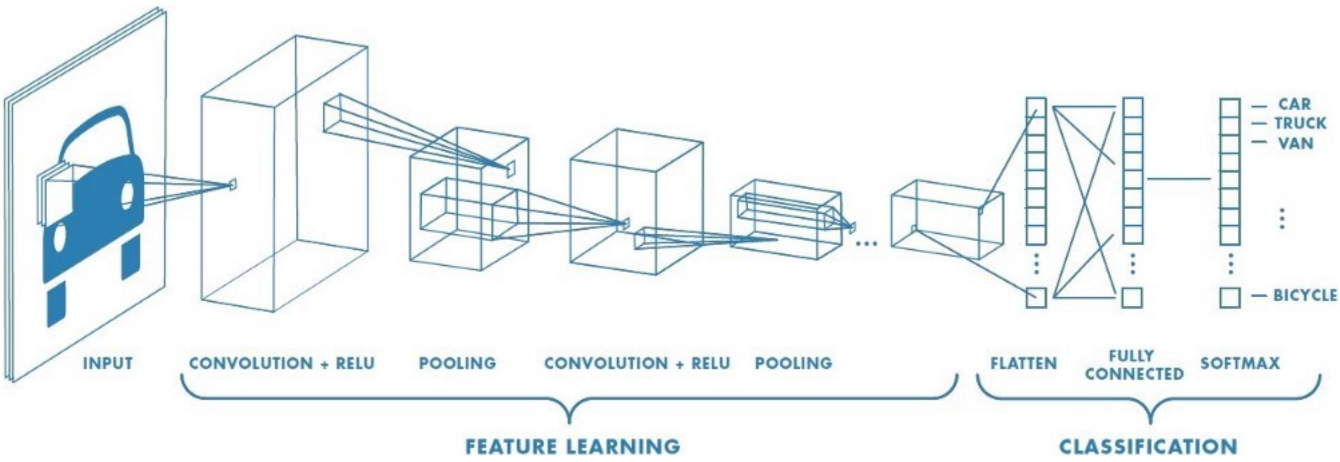


Fig 1. The architecture of Convolutional Neural Network

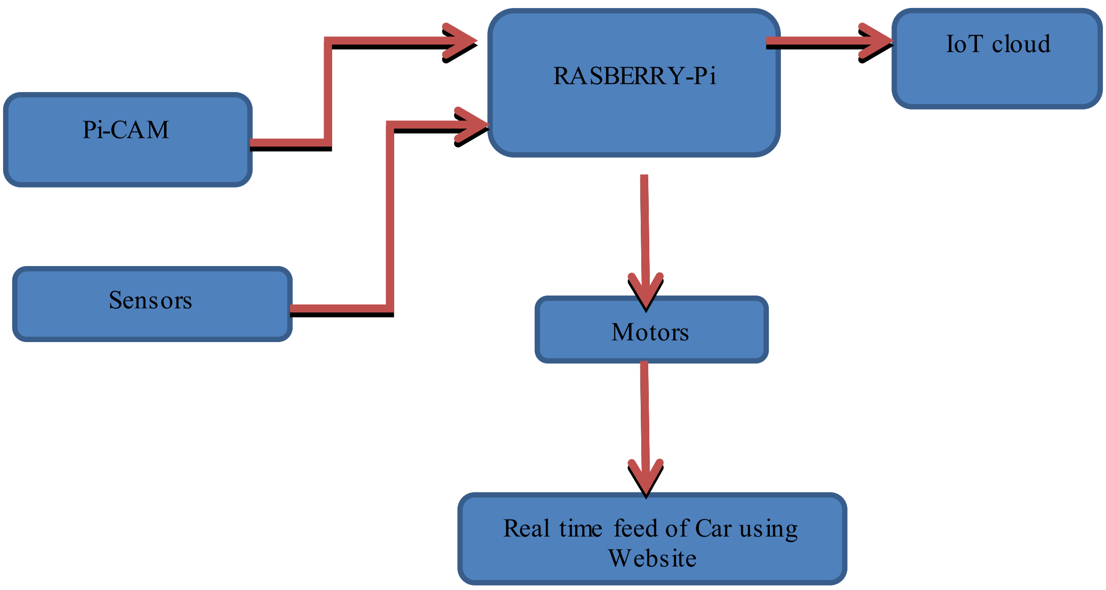
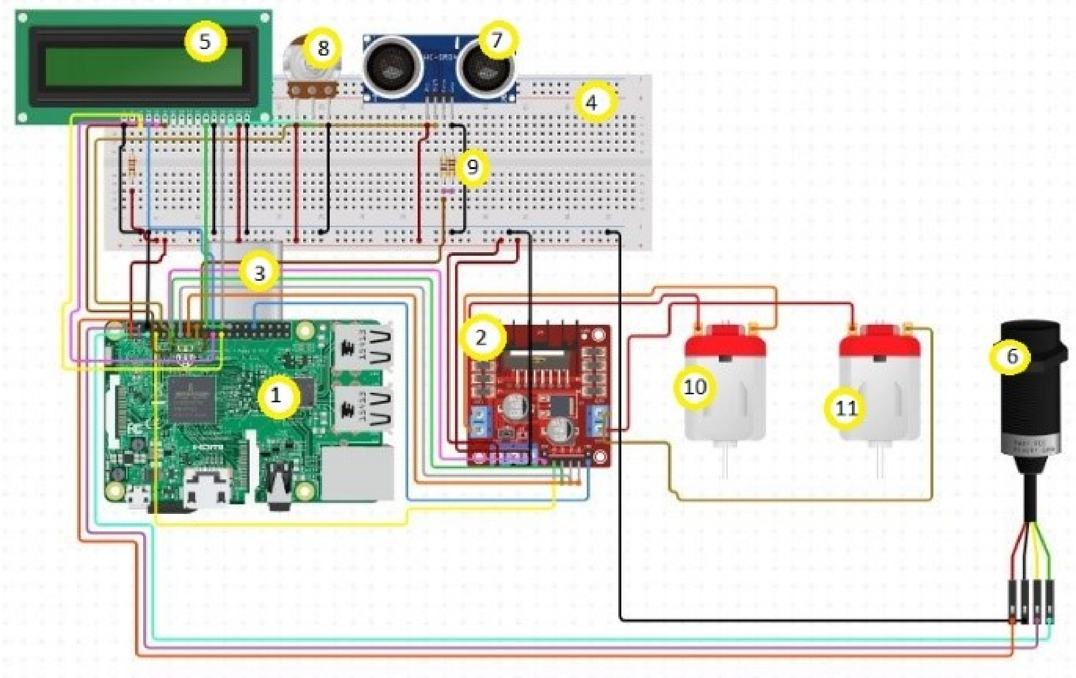


Fig 2: Block diagram of Autonomous car using CNN

Using a deep convolutional neural network and various image processing Techniques detecting real-time objects such as vehicles, Traffic Light, Traffic Signs etc. Detecting Lane using a neural network. Classifying the detected object using deep neural network.

Procedure for line detection:

- Giving insights regarding the detected object about their distance and location with regard to concern vehicle.
- Giving live feed of our vehicles on our web server using IoT.
- Live feed will have data regarding the environment seen by the robot using pi-Cam.
- Remote controlling of the vehicle on our Web Server using IoT
- Making of a small robot with various sensors like Ultrasonic sensor etc. to illustrate above-mentioned features.
- A semi-automated robot having the ability to make some real-time decision (Lane detection, stop & run, Traffic light condition etc.)



4

Fig 3: Experimental setup for lane detection using raspberry pi.

Components used in the experimental setup as follows.

1. Raspberry pi B+ model
2. **Motor Driver (L298N)**
   To drive DC and stepper Motors in the project, the L298N driver module is used. It is a high power driver module. Our module contains:
   - L298 motor driver IC
   - 78M05 5V regulator.

   This motor driver module can control two DC motors along with speed control and directional control.
3. Power Bank (for source)
4. Bread Board
5. LCD (liquid crystal display) (16x2)
6. **Pi CAM (for video feed)**
   The Raspberry Pi Camera Modules are official products from the Raspberry Pi. In our project, the 5-megapixel model Camera Module v2 is used.

TABLE 1: RASPBERRY PI CAM SPECIFICATIONS

Still resolution	5Mega pixel
Video Resolution	2592 × 1944 pixels
S/N ratio	36 dB
Sensitivity	680/lux-sec

7. **Ultrasonic sensor (HC-SR07)**
   The ultrasonic sensor is used in this project, to detect the object and to detect the distance between the object and source. This sensor contains a transmitter and receiver which used to transmit and receive inaudible sound waves, which is faster than audible sound. Based on the time to receive an echo from the object, the distance will be calculated.
8. Potentiometer(10k ohm)
9. Resistor(1K ohm)
10. Left Motor

11. Right motor

All the devices in the system are connected to the raspberry pi. From pi camera, input feed was given to the Rpi. Rpi analyzed by using the ultrasonic sensor to send the data to the Rpi. Rpi made the decision and sends it to left and right motor. The morion and status of the car will be known through LCD..

IV. MODEL FOR OBJECT DETECTION

There are several Neural Network architectures for detection:-

- R-CNN architectures
- Single Shot Detectors (SSD)
- YOLO — You Only Look Once

This paper is implemented on YOLOv3(Version 3 architecture) without going into many details as to how it works. YOLO models are not as accurate as R-CNN models but they are swift and are easily suitable for real-time applications. All these images are collected in a separate folder, named as training dataset and have labelled them by drawing bounding boxes around every car you found.

Yolo Object Detection Algorithm

It is a single neural system that predicts jumping boxes and groups the different item in one assessment. In the given model procedure, can process 45 edges for each second progressively and have executed this model as convolutional organize. This model partitions the picture taken as a contribution to a type of network. If the focal point of an article falls in a network cell, the cell is liable for recognizing the object. Each matrix cell predicts many jumping boxes with the assured scores which reflects how much the bounding box is sure that the article is available in the case. Each bounding box comprises of 5 expectations: x, y, w, h and certainty score.
Where: -

X = Center coordinate of bound box (x-axis)
Y = Center coordinate of bound box (y-axis)
H = Height of the bound box
W = Width of the bound box.

Fig 4: YOLO V3 Network Architecture.

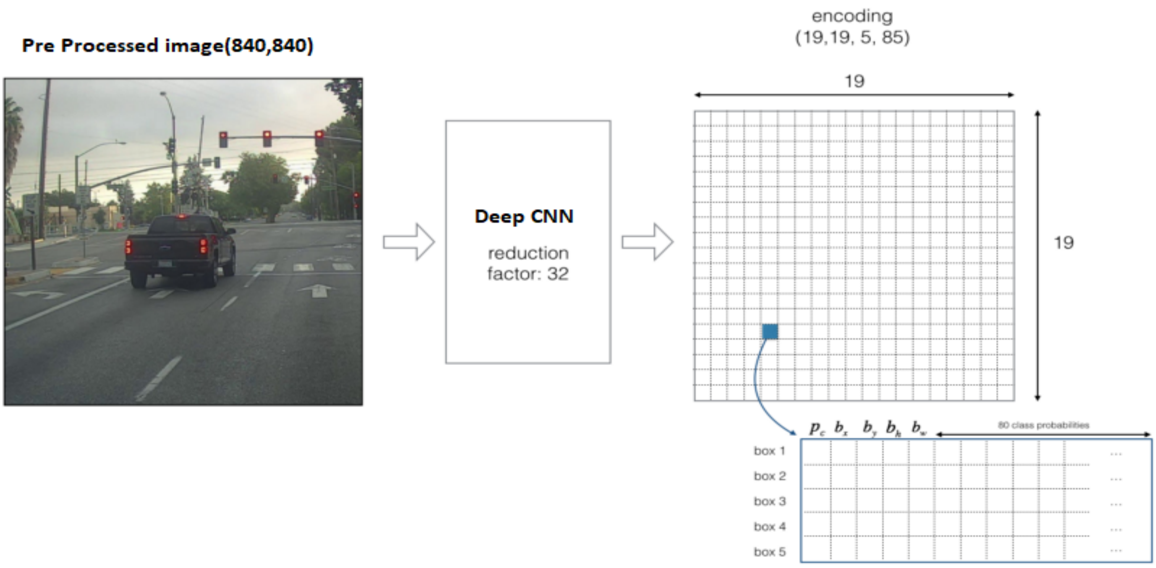


Fig5. Encoding architecture for YOLO V3

Design

Underlying layers of the convolutional neural system separate highlights while the completely associated layers yield directions and probabilities. Our model has 24 convolutional layers (CNN) in continuation with 2 completely associated layers.

In this, the primary purpose is to predict the lane in the image. This can be done by finding the change in gradient. Change in gradient to the color is the change in the intensity or difference in the intensity between adjacent pixels. So, first, have to convert our image to grayscale. The images are taken using a camera with 840X840.

A. *Inputs and outputs*

- The input is a batch of images, and each image has the shape (m, 840, 840, 3).
- The output is a list of bounding boxes along with the recognized classes. Each bounding box is represented by 6 numbers — pc, bx, by, bh, bw, c. 80-dimensional vector is used.

B. *Anchor Boxes*

- Anchor boxes are chosen by exploring the training data to choose reasonable height/width ratios that represent the different classes.
- The dimension for anchor boxes is the second to last dimension in the encoding: m, nH, nW, anchors, classes.
- The YOLO architecture is: IMAGE (m, 840, 840, 3) -> DEEP CNN -> ENCODING (m, 19, 19, 5, 85).

C. *Encoding*

In the encoding process, reduction factor 32 is used, Each box contains six features pc,bx,by,bh,bw,c as shown in figure 4. The object is identified with a probability score of 0.44 as shown in figure 5.

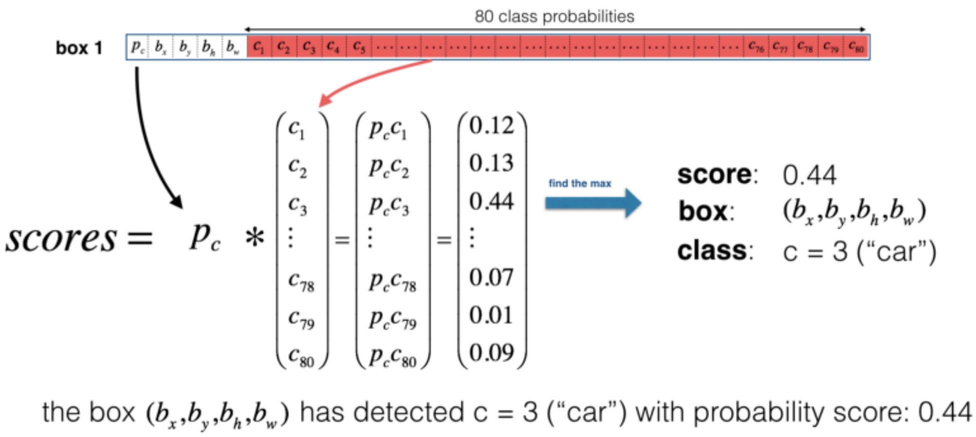


Fig 6: Class identification by each class box

(a) Original image

(b) Processed image

Fig 7: Shows the caany detection output from the input lane to edge detected lane

After this, to accurately detect as many edges possible in the image, noises to be reduced and smoothening of the image is required. The Gaussian filter has been applied for smoothening of images. Figure 7 shows how the image can be transferred into an edge detection image. The transformation of the original image to the processed image is shown in figure 7. Now for edge detection, the Canny edge detection method is applied to find the gradient of the image at each pixel. Now our image will be masked with the triangular mask which will be our area of interest and help us to find the lane.

V. RESULTS

The above-mentioned algorithm i.e. Yolo for object detection has been used and got all the bounding boxes whose threshold value is greater than the 0.44. As seen from the above image from Yolo algorithm, not able to just detect the traffic light but also the other things like (vehicles, people etc.). This gives the driver a good inference about his surroundings. This will help the driver to know his surrounding properly in case of loss of attention of the driver. This algorithm proves to be the fastest algorithm to detect the object in the surrounding and a valuable algorithm in the field of an autonomous car.
The experimental results of autonomous car detection were shown in figure 8 and 9. Figure 5 shows the object detection whereas figure 9 shows the lane detection in real-time.

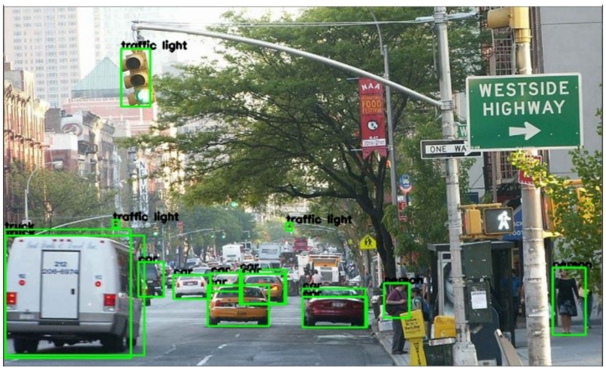


Fig 8: Object detection

Fig 9: Lane detection

From the lane detection algorithm, i.e. canny edge detection, it's a very simple algorithm to use to detect the edges in the images as shown in figure 8 and figure 9. This detection of edges has been utilized in the lane detection on the road. Apart from being a simple model it is can only be used for only simple edge detection problems because the basic concept is of this network is convoluting the image with the filter and it can detect edges up to some extent only. This algorithm only takes the input in Grayscale images. But in our case, this algorithm is holding up very well. It can detect the edges of the lane in which the driver is moving. This also giving the driver a proper insight if he is moving in the lane correctly or not. In Table 2, given the parameters of training time, error and accuracy for YOLO v3

TABLE 2: SIMULATION PARAMETERS OF THE MODEL

Parameters	**YOLOv3**
Time for training	9 hours
Accuracy on Test data	100%
Input image resolution	840X840
Loss	0.1708
Time for prediction	1.5 sec
Detection of object types	Small, medium & large

VI. CONCLUSION

From the above algorithms, can be concluded that both algorithms are working well in their respective case scenario. The Yolo proves to be the fast algorithm and work great in a real-time scenario for Object Detection. Instead of classifying a single class i.e. class like (traffic) lights, it can detect multiple classes like (people, vehicles, traffic, and lights, animal) giving a full and proper insight to the driver about his/her surrounding. It also uses CNN like other object detection but because of its speed and performance in a real-time scenario, it out-performs other algorithms. But a part of its huge advantages, it has some limitations like YOLO forces solid spatial requirements on jumping box forecasts since every lattice cell just predicts two boxes and can just have one class. But despite this, it is great for an autonomous car. For Lane detection also, the algorithm that is Canny Edge detection is good as it solves our purpose. But it is a very simple algorithm which fails in a complex situation and have used this algorithm to reduce the complexity of our model and reducing both power consumption and inference time of our application. Whenever making the application there are mainly three important constraints (time, Power and space). Use this simple algorithm reduces both and it is working in our scenario so this outcast other algorithm but irrespective to other algorithms its accuracy is less

REFERENCES

[1] O. Popescu and et al., "Automatic incident detection in intelligent transportation systems using aggregation of traffic parameters collected through v2i communications," IEEE Intelligent Transportation Systems Magazine, vol. 9(2), pp. 64–75, 2017.

[2] R. Atallah, M. Khabbaz, and C. Assi, “Multihop v2i communications: A feasibility study, modeling, and performance analysis,” IEEE Transactions on Vehicular Technology, vol. 18(2), pp. 416–430, 2017.

[3] G. Trehard and et al., “Tracking both pose and status of a traffic light via an interacting multiple model filter,” in 17th International Conference on Information Fusion (FUSION). IEEE, 2014, pp. 1–7.

[4] A. Almagambetov, S. Velipasalar, and A. Baitassova, “Mobile standards-based traffic light detection in assistive devices for individuals with color-vision deficiency,” in IEEE Transactions on Intelligent Transportation Systems, vol. 16(3). IEEE, 2015, pp. 1305– 1320.

[5] J. Al-Nabulsi, A. Mesleh, and A. Yunis, “Traffic light detection for colorblind individuals,” in 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). IEEE, 2017, pp. 1–6.

[6] X. Li and et al., “Traffic light recognition for complex scene with fusion detections,” in IEEE Transactions on Intelligent Transportation Systems, vol. 19(1). IEEE, 2018, pp. 199–208.

[7] M. B. Jensen and et al., “Vision for looking at traffic lights: Issues, survey, and perspectives,” in IEEE Transactions on Intelligent Transportation Systems, vol. 18(2). IEEE, 2016, pp. 1800–1815.

[8] Y. Ji and et al., “Integrating visual selective attention model with hog features for traffic light detection and recognition,” in 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015, pp. 280–285.

[9] Z. Chen, Q. Shi, and X. Huang, “Automatic detection of traffic lights using support vector machine,” in 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015, pp. 37–40.

[10] Z. Chen and X. Huang, “Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking,” in IEEE Intelligent Transportation Systems Magazine, vol. 8(4). IEEE, 2016, pp. 28–42.

[11] Z. Shi, Z. Zou, and C. Zhang, “Real-time traffic light detection with adaptive background suppression filter,” in IEEE Transactions on Intelligent Transportation Systems. IEEE, 2016, pp. 690–700.

[12] X. Du and et al., “Vision-based traffic light detection for intelligent vehicles,” in 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2017, pp. 1323–1326.

[13] V. John and et al., “Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching,” in IEEE Transactions on Computational Imaging, vol. 1(3). IEEE, 2015, pp. 159–173.

[14] S. Saini and et al., “An efficient vision-based traffic light detection and state recognition for autonomous vehicles,” in 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017, pp. 606–611.

[15] J. Campbell and et al., “Traffic light status detection using movement patterns of vehicles,” in 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 283–288.

[16] Z. Ouyang and et al., “A cgans-based scene reconstruction model using lidar point cloud,” in IEEE International Symposium on Parallel and Distributed Processing with Applications. IEEE, 2017.

[17] K. Behrendt, L. Novak, and R. Botros, “A deep learning approach to traffic lights: Detection, tracking, and classification,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 1370–1377.

[18] M. Weber, P. Wolf, and J. M. Zollner, “Deeptlr: A single deep convolutional network for detection and classification of traffic lights,” in 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016, pp. 342–348.

[19] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” in IEEE Transactions on Intelligent Transportation Systems, vol. 13(4). IEEE, 2012, pp. 1484 – 1497.

[20] M. Chen, U. Challita, and e. a. Walid Saad, “Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks,” vol. arXiv:1710.02913. arXiv.incident detection in intelligent

[21] Soni, Vishal Dineshkumar, Challenges and Solution for Artificial Intelligence in Cybersecurity of the USA (June 10, 2020). Available at SSRN: https://ssrn.com/abstract=3624487.

[22] A. Shaout, D. Colella and S. Awad, "Advanced Driver Assistance Systems - Past, present and future," 2011 Seventh International Computer Engineering Conference (ICENCO'2011), Giza, 2011, pp. 72-82, doi: 10.1109/ICENCO.2011.6153935.

[23] L. Li and X. Zhu, "Design Concept and Method of Advanced Driver Assistance Systems," 2013 Fifth International Conference on Measuring Technology and Mechatronics Automation, Hong Kong, 2013, pp. 434-437, doi: 10.1109/ICMTMA.2013.109.

[24] Soni, Vishal Dineshkumar, Information Technologies: Shaping the World under the Pandemic COVID-19 (June 23, 2020). Journal of Engineering Sciences, Vol 11, Issue 6,June/2020, ISSN NO:0377-9254; DOI:10.15433.JES.2020.V11I06.43P.112 , Available at SSRN: https://ssrn.com/abstract=3634361

[25] Q. Wu, Y. Liu, Q. Li, S. Jin and F. Li, "The application of deep learning in computer vision," 2017 Chinese Automation Congress (CAC), Jinan, 2017, pp. 6522-6527, doi: 10.1109/CAC.2017.8243952.

[26] M. J. Cree et al., "Computer vision and image processing at the University of Waikato," 2010 25th International Conference of Image and Vision Computing New Zealand, Queenstown, 2010, pp. 1-15, doi: 10.1109/IVCNZ.2010.6148863.

[27] ‘A Comparative Study of Real Time Operating Systems for Embedded Systems’ International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 1, June 2016, ISSN: 2320-9801, IMPACT FACTOR 6.7