I. J. Engineering and Manufacturing, **2020, 6, 11-19**
Published Online December 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2020.06.02

Image Recognition Based Autonomous Driving: A Deep Learning Approach

N. M. Tahir
Department of Mechatronics and System Engineering Abubakar Tafawa Balewa University Bauchi, Nigeria

U. I. Bature*, M. A. Baba
Department of Computer and Communications Engineering, Abubakar Tafawa Balewa University Bauchi, Nigeria
Email: biusman@atbu.edu.ng *

K. A. Abubakar
Department of Materials & Metallurgical Engineering Nigerian Army University, Biu Borno, Nigeria

S. M. Yarima
Department of Electrical Engineering, Abubakar Tafawa Balewa University Bauchi, Nigeria

Received: 28 July 2020; Accepted: 06 September 2020; Published: 08 December 2020

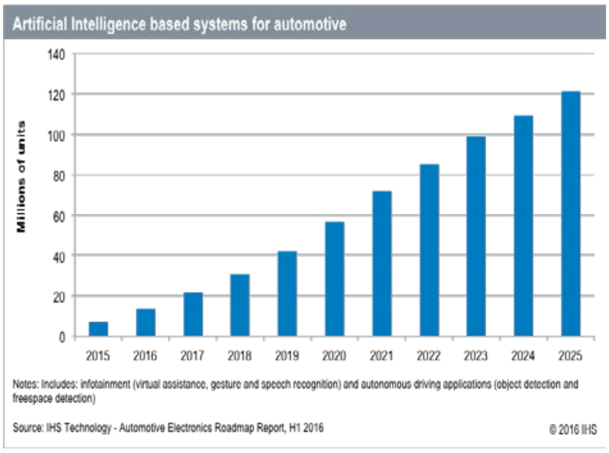
Abstract: Autonomous vehicle (AV) is a broad field in artificial intelligence which has seen monumental growth in the past decade and this had a significant impact in bridging the gap between the capability the intelligence of human and the efficiency of machines. With millions of people losing their lives, or have being a victim of road traffic accidents. There is a need to find a suitable algorithm for a navigation system in an autonomous vehicle with the purpose of help mitigate the traffic rule violation that most human drivers make that lead leads to traffic accidents. With both researchers and enthusiasts developing several algorithms for AVs, this field has been split into several modules which continually broaden the scope of AV's technology. In this paper, we focus on the lane navigation which has an important part of the AV movement on the road. Here lane decision making is optimized by using deep learning techniques in creating a Neural Network model that focuses on generating steering commands by taking an image the road mapped out with lane markings. The navigation aid is a front-facing camera mounted and images from the camera are used to compute steering commands. The end to end learning scheme was developed by Nvidia cooperation to train a model to compute steering command from a front-facing camera. The model does not focus on detecting the lane but only generating the appropriate command for steering AVs' on the road. This focus on one objective of the model helps in maximizing the potential of better accuracy in lane navigation of our AVs. The modeled car navigates through the designed lanes accurately with the level of intelligence the car shows in maneuvering through the lanes shows this method is more suitable in lane navigation.

Index Terms: Autonomous vehicle, lane navigation, deep learning method, lane navigation, steering command, autonomous navigation.

1. Introduction

The use of artificial neural networks (NNs) in the deep learning (DL) process involves the use of several machine learning techniques like language understanding, image segmentation, and process, speech process [1]. NN is a process of detecting or sensing the environment via several connected intelligent neurons after its activation by sensors that help in identifying the environs. These neurons were interconnected such that action can continuously be pass from one neuron to another via a real-valued activation process, the neuron can trigger an action to the next neuron through weighted connections, this can be termed as learning that makes the NNs shows some intelligent actions like guiding an object to follow the desired direction. Therefore, in machine learning and pattern recognition contests artificial DL has won several awards recently [2], the work summarizes several NNs works and reveals the significance and intelligence behind the DL processes.

A moderate distance detection can be achieved through computer vision, it can help in detecting lane as well as redundant objects, and similarly, radar detects objects like vehicles or metals. However radar cannot vividly distinguish between the objects and also has a higher variance, thus, it can cause errors when detecting sharp bends. Nowadays, the demand for systems that support artificial intelligence (AI) increases. This can be illustrated by the report from Information Handling Services (HIS) [3], it reveals that new vehicles with various installed AI system like the advanced driver assistance systems (ADAS), autonomous vehicles, and infotainment systems will emerge, rising from 8% in 2015 to 109% in 2025 as shown in Fig 1. It is worth noting that ADAS based systems are witnessing a tremendous increase these days [4-11].



Notes: Includes: infotainment (virtual assistance, gesture and speech recognition) and autonomous driving applications (object detection and freespace detection)

Source: IHS Technology - Automotive Electronics Roadmap Report, H1 2016

Fig. 1. Prospect Indicator of the AI-based system 2015 to 2025 [3]

Furthermore, the application of deep learning on several online and mobile platforms like Google Assistant, Microsoft Cortana, voice recognition, Facebook, Amazon's Alexa, and Webdnn, face recognition [12-14], etc. have shown the level of demand for AI-based system. Deep learning shows promise for self-driving vehicles, a real-time system requires CNNs for object and lane detection [6], this was illustrated in [15], the trained system using inputs from a video camera and laser rangefinder, tRNAscan-SE sometimes return false positive due to the growing scale of genomic datasets, the characterizing power of DL and CNNs may help in alleviating this effect [16]. Nowadays, computer vision has attracted huge attention in ADAS systems [6], mostly used in conjunction with other road models or sensors [17-20], they reveal various approaches on localizing the target object under some specified environmental sceneries.

This work uses a deep learning approach to design an autonomous system. We focus on the lane navigation that reveals the significance of Autonomous vehicle (AV) in today's activities, this approach will aid in providing an appropriate command for steering AVs on the road and better the accuracy in AV lane navigation. Consequently, this will aid in avoiding road accident related issues. Before the widespread adoption of CNNs into the field of artificial intelligence, most pattern recognition tasks in computer vision were performed using an initial stage of hand-crafted feature extraction followed by a classifier. The breakthrough of CNNs has made a significant impact in the world of pattern recognition. The CNN approach is widely adopted in image analysis but can also be used for data analysis and classification problems. CNN can be thought of as part of an artificial neural network that has some specialization of identifying patterns and able to interprets these patterns. And these pattern detections are what made it suitable for our deep learning model in this work.

The tasks involved in achieving autonomous driving can be divided into three modules: environment recognition, decision making, and vehicle control. With millions of people losing their lives, or have being a victim of road traffic accidents. There urgent need to find a suitable algorithm for a navigation system in an autonomous vehicle with the purpose of help mitigate the traffic rule violation that most human drivers make that lead leads to traffic accidents

The rest of the paper is structured as follows; section 2 is the methodology that provides the CNN architecture and Parameters of the Structure of Network Architecture used, section 3 gives the discussion on the results obtained like the lane detection procedures, Lane Navigation Using Deep Learning and validation and evaluation of the proposed method. Finally, section 4 provides the conclusion and future work.

2. Methodology

Inspired by the work of NVidia's team in developing self-driving cars using end to end learning methods, we thought of creating a model specifically to take on the task of handling lane navigation of Autonomous vehicles. Since it is easier to maximize a model by specifying a single task to perform. In a neural network model, it learns from the training data and its target is son specific output it can learn specifically the features that will determine the output in our case it's the steering command for navigation [4].

Images are fed into a CNN that then computes a proposed steering command. The proposed command is compared to the desired command for that image, and the weights of the CNN are adjusted to bring the CNN output closer to the desired output. The +weight adjustment is accomplished using back-propagation. The training model for the neural network is shown in Fig 2.

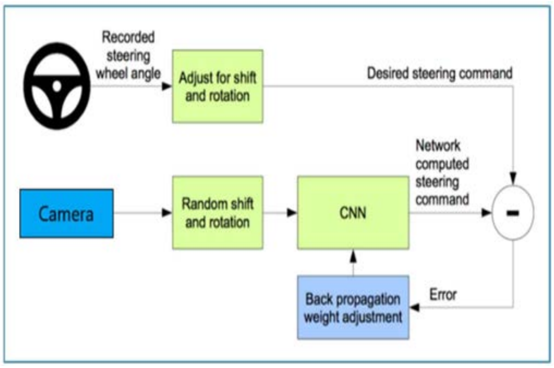

Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Network computed steering command
Camera
Random shift and rotation
CNN
-
Back propagation weight adjustment
Error

Fig. 2. Training the neural network.

Machine learning is a branch of artificial intelligence that focuses on using an algorithm to analyze data and learn from the data and able to make a new prediction on new data. Deep learning is a subfield of machine learning that uses algorithm inspired by the structure and function of the human brain neural pathways, to analyze and learn from data and identify a structure to be able to make an accurate prediction on similar data [2].

The deep learning model is inspired by the NVidia model for their self-driving car. That can map raw pixels from a front-facing camera into the steering angle. The model is made of convolutional layers and some fully connected layers and is a train on a data set that contains a lane pathway from a camera frame labeled with the correct lane navigation angles. The model is trained until can predict accurately the navigation angles for new lane pathways. Though the process state for achieving this looks straight forward there is a certain process to achieve it. The image pixel value to a range of 0 to 1, 5 convolutional layers, and 3 fully connected layers as shown in Fig 3. The first layer of the network which is the normalization layer is hard-coded and was not included in the learning process. The convolutional layers were designed to perform feature extraction. The five convolutional layers are followed by three fully connected layers, leading to a final output control value which is the inverse-turning-radius. The Nvidia model architecture was faithfully implemented, except for the normalization layers which was removed from inside of the model (as it was implemented outside the model), and some dropout layers were added, to make the model more robust [19]. Table 1 shows the parameter of the architecture.

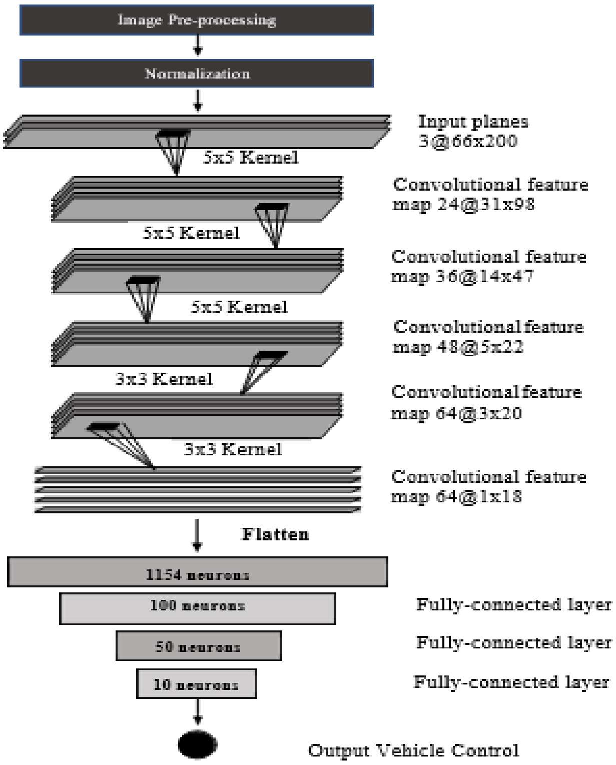


Fig. 3. CNN architecture.

Table 1. Parameters of Network Architecture

Structure of Network Architecture			
Design Parameter	Layer	Output Shape	Parameter
Convolutional layers	Conv2d	(None, 31, 98, 24)	1824
	Conv2d_1	(None, 14, 47 36)	21636
	Conv2d_2	(None5, 22, 48)	43248
	Conv2d_3	(None, 3, 20, 64)	27712
	Conv2d_4	(None, 1, 18, 64)	36928
Dropout part	Dropout	(None, 3, 20, 64)	0
	Dropout_1	(None, 1150)	0
Flatten	Flatten	(None, 1150)	0
Fully connected layers	Dense	(None, 100)	115300
	Dense_1	(None, 50)	5050
	Dense_2	(None, 10)	510
	Dense_3	(None, 1)	11

2.1 Hardware Requirements

The software runs on standard hardware as shown in fig. 4. It can also be run whether in the time-sharing network, mainframe, or minicomputers, thus the hardware requirements are:

IBM Intel or Microsoft compatible computers.

- Processor: Pentium 3.0 GHz or above
- Memory: 3 GB RAM or above
- Hard Disk: 100GB or above [at least 2GB free space required]

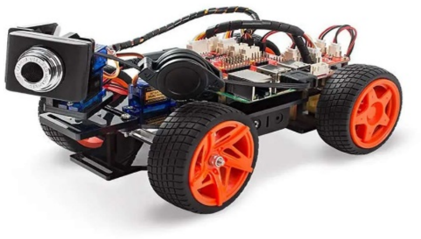


Fig. 4. The Pi Car

3. Result and Discussion

The autonomous lane navigation has been broken down into two parts Lane detection (using Hough Transform) and Lane Navigation. This section dives deep into the analysis of the data used in the process of achieving the results and why these methods are used.

3.1 Lane Detection

Lane detection is achieved by using the computer vision package Open CV, to make this to work on a camera from the car we made it to work on an image first. The image is passed through a series of filters to enhance the lane features, an example is converting the image to grayscale which helps creates an image with a pixel value of lane being of higher value than that of the background, this makes it easier for edge detection when the Canny edge detector as applied on the image shown in Fig 5.

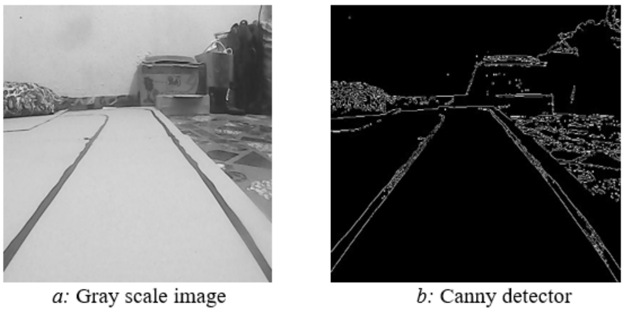


Fig. 5. Transformation by Canny Detector

The edge detector returns an image with all the edges. The canny image does this by getting places with a sharp change between pixel values between the pixel images. The pixel in which the detector determines that are edges is returned. The image has a redundant part which is not part of the lane. These parts are cropped out which leaves an image with only our lanes detected as edges.

This is when the Hough Transform comes into play. The Hough Transform takes a lot of parameter such as:

- The voting threshold is the number of votes needed to be considered a line segment. If a line has more votes, Hough Transform considers them to be more likely to have detected a line segment. (set to 10)
- Minimum Line Length: is the minimum length of the line segment in pixels. Hough Transform won’t return any line segments shorter than this minimum length. (Set to 6).
- Max Line Gap is the maximum in pixels that two-line segments that can be separated and still be considered a single line segment (set to 4).

This gives the lane lines for the lanes from the segmented image as shown in Fig 6.

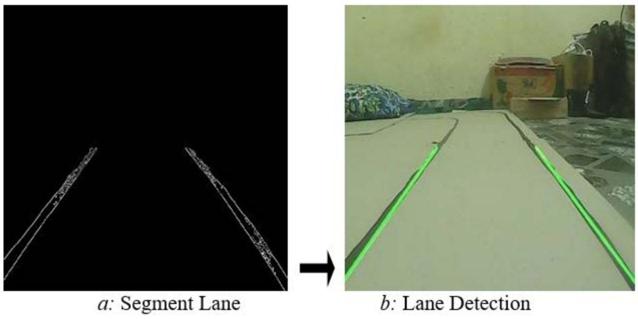


Fig. 6. Transformation by Hough Transform.

3.2 Lane Navigation Using Deep Learning

In this module, the data, results, techniques, and decision made would be focused on while training the model. The model training was done online through Google Collab using Keras and Tensor Flow as a backend to train the model, its architecture which is an imitation of the Nvidia model consists of 8 layers 5 convolution layers: first three convolutional layers with a 2×2 stride and a 5×5 kernel, and a non-stride. Convolution with a 3×3 kernel size in the final two convolutional layers, and 3 fully connected layers, which leads to a final output control value which is the inverse-turning-radius. The fully connected layers are designed to function as a controller for steering, but we noted that by training the system end-to-end, it is not possible to make a clean break between which parts of the network function primarily as feature extractor, and which serve as the controller.

3.3 Evaluation

Evaluation of the model was done in two phases: Simulation and Experiment. In simulation first, thing is to plot the loss function of both training and validation sets

1) Validation and training accuracy

Validation accuracy is used to measure the ability of the model to generalize on unseen data. From Fig 7, it can be seen that the validation performance has separated from the training performance a bit at the end of the cycle. This means that the model has started over-fitting, if the learning rate was changed and train further, the model would only memorize features from the training set and the validation set performance would rise.

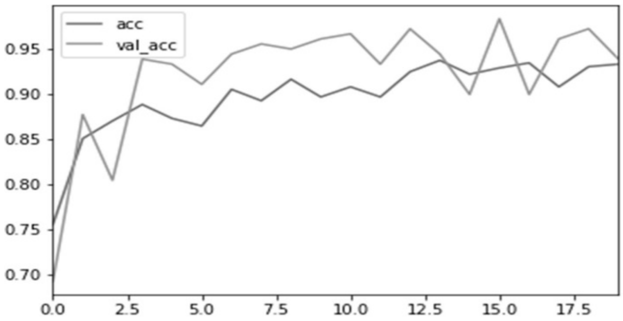


Fig. 7. The plot of validation accuracy and training accuracy

2) Validation and Training Loss

It is good to see that both training and validation losses declined rapidly together, and then stayed very low after epoch 5. There wasn't any over-fitting issue, as validation loss stayed low with training loss. After changing the learning rate, it can be observed from the plotted losses in Fig 8 that there is a small rise after the initial drop but this is still better than the initial.

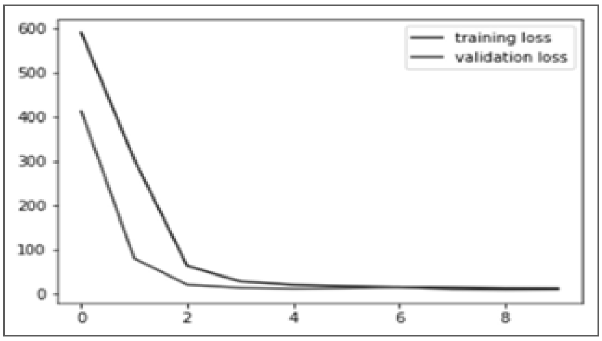


Fig. 8. The plot of validation loss and training loss

A. Testing the Model

The model is tested by feeding it a recorded video to get steering command from the model and see if our training is following the lane by the heading line plotted by the model on the frame. After testing on the video, it was found that the model follows the lane in the video but there are situations where the car skids off the lane marking this is because of the limited amount of training data and feeding more data to train the model will overcome this constraint. The way the model works during the testing is shown in Fig 9.

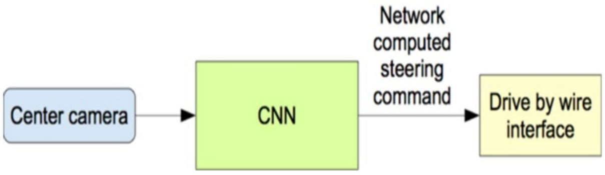


Fig. 9. The trained network is used to generate steering commands from a single front-facing center camera.

B. Image from the Front-facing Camera

When interfacing the hardware, the deep learning method for lane navigation is applied. The car is made to navigate through the lane using this method about 5 times and how well it follows the lane was recorded and where the algorithm was lagging was also recorded. The video from the front-facing camera, while it was following through the lane, was also recorded and the best from the five times of navigating through the lane was selected as shown in Fig 10.

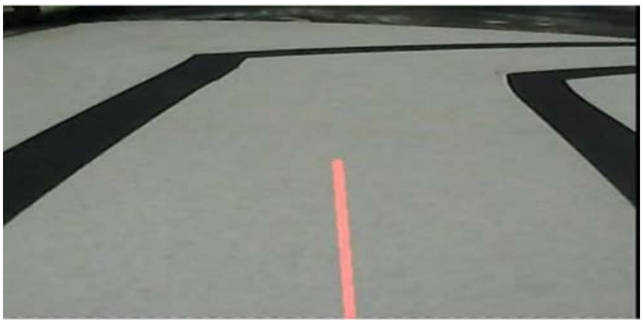


Fig. 10. The end to end follower

The end to end model is used to drive the car through the lanes five times and the best of the best video from the camera while navigating through the lanes is also selected and the places where the model fails to obtain the correct steering angle or deviate too far from the lanes are also recorded.

4. Conclusion and Future Work

The deep learning model generally performs well in a new environment and can take care of scenarios that slightly differ but have the same feature with what it’s being trained on. For example, if the color sequence or the camera is changing, other methods might find it difficult to navigate through the lane whereas a good model can navigate with a little hitch if any. A good point recognized is that a model can continually be improved by just giving it more data to train on and it can continually be improved easily to fit in any environment. Though the raspberry pi is having a good processing power it is not meant to run Tensor Flow models, this can be solved by using hardware specialized for running Tensor Flow models such as Google-Edge TPU which can be used with the raspberry pi. Secondly, there is a need to use a good camera with a wide-angle for better pixel density that could provide better processes for detection and steering command generation by the model. As a recommendation, an additional label for the training such as the speed of the car will serve to improve the Navigation of the car through the lane tracks and real-time experimentation on a different type of car.

Acknowledgement

The authors are indebted to the Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University Bauchi for the provision of the necessary facilities and guidance.

References

[1] Jyoti Shokeen, Chhavi Rana, "An Application-oriented Review of Deep Learning in Recommender Systems", International Journal of Intelligent Systems and Applications(IJISA), Vol.11, No.5, pp.46-54, 2019. DOI: 10.5815/ijisa.2019.05.06

[2] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks, Vol. 61, pp. 85-117, 2015.

[3] L. De Ambroggi, “Artificial Intelligence Systems for Autonomous Driving on the Rise, IHS Says”, IHS, June 2016, [online] Available: https://technology.informa.com/579746/artificial-intelligence-systems-for-autonomous-driving-on-the-rise-ihs-says. Retrieved 26 July 2020.

[4] Falcini, Fabio, Giuseppe Lami, and Alessandra Mitidieri Costanza. "Deep learning in automotive software." IEEE Software, vol. 34, no. 3, pp. 56-63, 2017.

[5] Fujiyoshi, Hironobu, Tsubasa Hirakawa, and Takayoshi Yamashita. "Deep learning-based image recognition for autonomous driving." IATSS Research, 2019.

[6] Huval, Brody, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka et al. "An empirical evaluation of deep learning on highway driving." arXiv preprint arXiv:1504.01716, 2015.

[7] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks, vol. 61, pp. 85-117, 2015.

[8] Luckow, Andre, Matthew Cook, Nathan Ashcraft, Edwin Weill, Emil Djerekarov, and Bennie Vorster. "Deep learning in the automotive industry: Applications and tools." IEEE International Conference on Big Data (Big Data), pp. 3759-3768, 2016.

[9] Bojarski, Mariusz, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel et al. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316, 2016.

[10] Lee, Unghui, Jiwon Jung, Seokwoo Jung, and David Hyunchul Shim. "Development of a self-driving car that can handle the adverse weather." International journal of automotive technology 19(1) pp. 191-197, 2018.

[11] Kim, Jinkyu, and John Canny. "Interpretable learning for self-driving cars by visualizing causal attention." Proceedings of the IEEE international conference on computer vision, pp. 2942-2950. 2017.

[12] Hidaka, Masatoshi, Yuichiro Kikura, Yoshitaka Ushiku, and Tatsuya Harada. "Webdnn: Fastest dnn execution framework on web browser." Proceedings of the 25th ACM international conference on Multimedia, pp. 1213-1216. 2017.

[13] Muhammad Aamir, Ziaur Rahman, Waheed Ahmed Abro, Muhammad Tahir, Syed Mustajar Ahmed, "An Optimized Architecture of Image Classification Using Convolutional Neural Network", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.10, pp. 30-39, 2019.DOI: 10.5815/ijigsp.2019.10.05

[14] Priya Gupta, Nidhi Saxena, Meetika Sharma, Jagriti Tripathi,"Deep Neural Network for Human Face Recognition", International Journal of Engineering and Manufacturing(IJEM), Vol.8, No.1, pp.63-71, 2018.DOI: 10.5815/ijem.2018.01.06

[15] Pomerleau, Dean. "Rapidly adapting artificial neural networks for autonomous navigation." Advances in neural information processing systems, pp. 429-435. 1991.

[16] Gao, Xin, Zhi Wei, and Hakon Hakonarson. "tRNA-DL: a deep learning approach to improve tRNAscan-SE prediction results." Human heredity vol. 83, no. 3, pp. 163-172, 2018.

[17] Caraffi, Claudio, Tomáš Vojíř, Jiří Trefný, Jan Šochman, and Jiří Matas. "A system for real-time detection and tracking of vehicles from a single car-mounted camera." International IEEE conference on intelligent transportation systems, no. 15, pp. 975-982, 2012.

[18] Schneiderman, Henry, and Takeo Kanade. "A statistical method for 3D object detection applied to faces and cars." Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), vol. 1, pp. 746-751, 2000.

[19] Jazayeri, Amirali, Hongyuan Cai, Jiang Yu Zheng, and Mihran Tuceryan. "Vehicle detection and tracking in-car video based on motion model." IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 583-595, 2011.

[20] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

Authors' Profiles

N. M. Tahir received the B.Eng. degree in Electrical Engineering from Bayero University, Kano in 2012, and M.Eng. Degree in Mechatronics and Automatic Control from the Universiti Teknologi Malaysia in 2016. He has been involved in research at Mechatronics systems and Robotics Laboratory in Universiti Teknologi Malaysia on vibration control, robotics, and Mechatronics system design. He has been a lecturer at the Department of mechatronic and system Engineering, Abubakar Tafawa Balewa University, Bauchi Nigeria since 2014. His main research interests include Vibration Control, Mechatronics Systems Design, Control System Design, and Robotics

U. I. Bature is currently a Ph.D. Candidate in the Department of Electrical Engineering, Universiti of Teknologi Petronas (UTP) Malaysia. He received a B.Eng. Computer Engineering from Bayero University Kano (BUK), Kano city, Nigeria, and degree of Master of Engineering (Electrical - Computer and Microelectronic system) from Universiti Teknologi Malaysia (UTM), Skudai, Johor Bahru, Malaysia. He is currently a lecturer in the Department of Computer and Communications Engineering, Abubakar Tafawa Balewa University Bauchi (ATBU), Nigeria. His research interest includes Image processing, Nano-material design, embedded systems, and Biomedical Systems.

K. A. Abubakar received his B.Eng. degree in Electrical Engineering from Bayero University, Kano in 2012, and M.Eng. Degree in Mechatronics and Automatic Control Engineering from the Universiti Teknologi Malaysia in 2016. He has been involved in research works at Mechatronics Systems Laboratory and Centre for Artificial Intelligence and Robotics (CAIRO) in Universiti Teknologi Malaysia on Robotics, Control, Artificial Intelligence, Neural Network, and Mechatronics system design. He has been a lecturer at the Department of Materials and Metallurgical Engineering, Nigerian Army University Biu, Borno, Nigeria, since February 2020. His main research interests include Vibration Control, Mechatronics Systems Design, Control System Design, Robotics, and Software development.

M. A. Baba obtained his first degree in 2010 from Abubakar Tafawa Balewa University Bauchi, Nigeria (ATBU) in Electrical/Electronic Engineering. He further received his MSc. Degree in Information Technology from the University of Wolverhampton, U.K, in 2013. Obtained his Ph.D. from the University Teknologi Malaysia UTM, Malaysia in 2020. His areas of research interest include a meta-material antenna for MIMO applications and have been with ATBU Bauchi since 2014 as a lecturer

S. M. Yarima is currently a Ph.D. student with the Department of Electronics and Computer Engineering, University Teknologi Malaysia, UTM, since 2017. He obtained both his B.Eng and M.Eng degrees from the Abubakar Tafawa Balewa University (ATBU) Bauchi in 2008 and 2014 respectively. Before the start of a full academic carrier as a University lecturer in the Department of Electrical and Electronics Engineering of the Abubakar Tafawa Balewa University (ATBU) Bauchi in 2009, he had had both constant and intermittent touch with the Television and the Power companies in Nigeria. His major duties include teaching, research, and supervision of undergraduate students. He has a great interest in the research areas of digital signal processing (DSP), internet-of-things (IoT), and communications.

How to cite this paper: N. M. Tahir, U. I. Batureb, K. A. Abubakar, M. A. Baba, S. M. Yarima, " Image Recognition Based Autonomous Driving: A Deep Learning Approach ", International Journal of Engineering and Manufacturing (IJEM), Vol.10, No.6, pp.11-19, 2020. DOI: 10.5815/ijem.2020.06.02