Application of Deep Learning to Autonomous Robotic Car

Oluwagbemiga O. Shoewu, Samuel O. Adebayo
Dept of Electronic & Computer Engrg
Lagos State University
Epe, Lagos State

Oluwafemi J. Ayangbekun
Department of Computer Science
Redeemer's College of Tech& Mgmt. Mowe,
Ogun State, Nigeria

Lateef A. Akinyemi
Dept. of Elect/Computer Engr.
Lagos State University
Epe, Lagos State

ABSTRACT

Autonomous machines are becoming prevalent, even more so the advent of autonomous vehicles. While autonomous cars have been around for some time, the endless innovations in this domain have led to the removal of human-in-the loop, hence constantly seeking to remove human input while delivering optimal result. However, safety is a major concern, and users are wary of leaving safety level decisions to machines. There is a rise in road accident caused by autonomous cars, while some have blamed it on human's total trust in machines, and researchers have called for the development of human-level accurate algorithms to tackle decision making using state-of-the-art techniques. Therefore, this paper seeks to use computer vision leveraging on deep learning techniques to detect pedestrians, traffic signs, important objects, and lane lines to infer crucial driver decisions.Mask R-Convolutional Neural Network (CNN) was used for object classification with the aid of transfer learning saving the hassle of training and GPU times. A simple method for collecting data was applied using a wide-anglecamera and using Google TPU to perform real time object recognition without the need for a GPU enabled machine.

General Terms

Autonomous robotic car

Keywords

Deep Learning (DL), Robotic, Autonomous, Object detection, Object recognition, Motor

1. INTRODUCTION

Autonomous vehicle (AV) system is a technology system of vehicular construct with the ability to move and reach its destination without intended intervention by humans. Autonomous vehicles have been categorized into aerial vehicles, ground vehicles, and underwater vehicles. While the major focus of this paper is ground vehicles, other categories of AVs could also adopt technologies of ground AVs in its familiarizing and interaction with the immediate environment. AVs requires no intended human intervention and as such needs to be able to make decision and perform basic perception, such as human, vehicle, and traffic identification, location detection, measure proximity between vehicles, and make intelligent decisions [1].

For most AVs identifying obstacles usually require the use of expensive sensors, and the accuracy of the sensor of detection is left at the mercy of the sensors' range which are often limited by noises and environmental condition. The choice of sensor depends on its applicability. Examples of sensors in AVs are LIDAR, GPS, camera, 3D Radar, Ultrasonic sensors and LRF. While LIDAR allows for mapping the distance regardless of the environment and does not require a navigation system (GPS/INS). The operating costs is 10k/hour, and it is affected by sun's reflections. GPS on the other hand could also be used to infer the geolocation of the vehicle, however, it may be misleading due to bad atmospheric conditions. For LRF, it cannot detect object that are in proximity, and it consumes high energy. The use of high precision cameras could be used to bridge the gaps and lapses found in all the other sensors.

Recent developments in AVs have shown promising results, however, safety remains a major concern. In a recent survey, some of the highlighted concerns are vehicle breakdowns, hardware and software malfunctions [2]. While these are open challenges and are currently been addressed by teams of researchers and engineers. The issue of safety is still of major concern, in fact according to the survey 85% of respondents are not willing to be passengers in a driverless car [2], and majority of the respondents claimed they would feel unsafe as pedestrians when AVs become mainstay on major roads. These unsafe feelings became even more heightened since one of Uber's self-driving cars was involved in a crash [3].

According to the World Health Organization, about 1.35 million people die globally from road accidents annually [1]. In Nigeria, 39,000 people die via road accidents annually [2]. However, the cause of these road traffic accidents has been discovered to be multifaceted. Other than bad roads which is the major contributing factor, other factors such as drivers' behavior, visual and auditory judgement, decision making ability and over-speeding contribute to these accidents. Hence, safety should be a major factor when developing AVs, as the addition of these vehicles with no human sympathy or empathy to our roads could lead to an increasing number of road crash. The number of deaths and injuries could reduce if major failures of AVs such as inability to identify pedestrian, traffic identification, and vehicular safety distance are nipped in the bud at the early stage. As a result of addressing these issues, the application of deep learning for object recognition and classification is applied to autonomously drive a robotic car, avoid obstacle, identify pedestrian, and slow down at speed limit.

2. RELATED WORK

The advent of deep learning for computer vision has achieved beyond human accuracy for object recognition [4]. Moreso, the swift innovations in deep learning and advancement in hardware facilities, such as high-resolution cameras, improved memory capacities, and high computing processor enabled devices to have enhanced the applications of deep learning to nearly all aspects of image-based problems.

Driverless cars used to be science fiction out of cartoon series and animated movies, but today they are our reality. Carnegie Mellon University (CMU) started development of computer-controlled vehicle in 1984, and they completed the first project on autonomous vehicles in 1995. The team at CMU autonomously drove the vehicle from Pittsburgh, PA, United States to San Diego, Canada. The said vehicle was equipped with a computer, a perception, and a GPS. In 2005, a team of research engineers won the first place at the American Defense Advanced Research Projects Agency (DARPA) challenge. Autonomous car technology is already being developed by the likes of Lexus, BMW, and Mercedes, and Tesla is currently testing a lot of her driverless autopilot systems on UK roads. Across the Atlantic, Google is developing its automated technology in the wild, and Apple is rumored to be working with BMW on its own – probably automated – car.

2.1 Deep Learning for Computer Vision

Deep learning is a subset of machine learning that uses artificial neural networks (ANNs), a computational imitation of neurons in the human brain. These artificial neurons are responsible for simple operations, interconnectedly weighted for making decisions [5]. Traditional computer vision approach uses algorithms to separately compute edge detection, color segmentation, background subtraction, etc. to extract essential features from images. These extracted features are computed as bag-of-words and would later be used for tasks on new images. Therefore, in an object detection tasks, if these bag-of-words are found on new images, the images are classified, respectively. As against traditional approach to computer vision, DL does not need a separate process devoted to feature extraction, patterns and features are learned internally. With deep learning, a model is trained on a specific data peculiar to the task at hand (object detection, facial recognition, motion tracking, etc), hence eliminating the feature engineering stage. Deep learning has advanced the field of computer vision, eliminated unnecessary hard computation and manual engineering. It has performed exceedingly beyond known traditional feature engineering for computer vision tasks.

The introduction of CNN has had a wide impact on CV. CNNs were first implemented by Yann LeCun et al. to handwriting recognition [7]. CNN is made up of convolutional layers and artificial neural networks. The neural network is made up of input layer, convolutional layer, pooling layer, and fully connected (FC) layer. The convolutional layer is synonymous to a filtering layers found in traditional CV for extraction of edges, patterns, corners from an image. The convolution is made up of a kernel filter (basic 3 x 3) which strides over an image matrix to perform element wise multiplication. A non-linear activation (such as sigmoid, ReLU, LReLU, etc) function is applied to the output of the convolution The resulting values are summed and stacked into a matrix which would be used for further computation. The most popular pooling applied to CNN tasks are Max pooling, which takes the maximum value from the kernel computed matrix, and average pooling which takes the average value of the kernel computed matrix. The output of the previous layer is flattened into a 1D array of numbers and connected to a FC layer.Hence, for a classification task the final FC has the same number of output nodes as the number of classes. Each FC is followed by a non-linear activation function. Notable CNNs are **VGG models, ResNet, AlexNet, DenseNet, Stacked Hourglass**, etc.

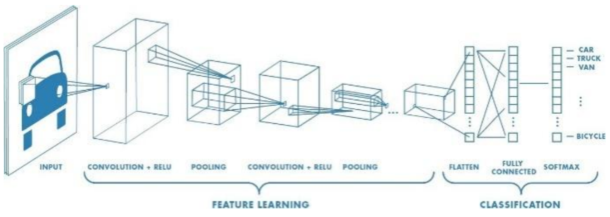


Fig 1: A CNN sequence to classify an image [8].

3. BASIC RESEARCH COMPONENT

3.1 Hardware Component

The following are the hardware components used in this project:

1. Raspberry Pi 4: this is a minicomputer on a chip. It is an atm sized computer with up to 8GB RAM, quad-core CPU, ethernet ports, USB 3.0 ports, wireless LAN, Bluetooth 5.0, and USB-C power [14]. Raspberry pi 4 was chosen due to its portable size, desktop-level performance, and capacity for other ad-on devices.
2. Memory Storage of 64GB.
3. Sunfounder PiCar-IV kit: The Chassis has the basic framework of a robot car, a space for a DV battery and enough space to accommodate the electronic panel and other mounted hardware. The Robot car chases is made of heat resistant plastic and could withstand a medium to high operating temperature. It is equipped with front and back wheel DC motor, and two unit of servo motor to serve as torque source.
4. Rechargeable batteries: 7V rechargeable DC batteries were used to power the robotic car.
5. Google Edge TPU: this is a add-on for Raspberry pi, since, raspberry pi is unable to meet the demand of performing deep learning inference at a faster rate. EdgeTPU was introduced in March 2019 by the Google Team, and it was designed to run deep learning models (written in tensorflow) on mobile and embedded devices [12].
6. Perception Camera: a 150^{o} wide angle was used to capture real-time images and record sequential frames for online processing and inference. The camera is plug and play and requires no calibration. It is mounted at the front middle of the robotic car for reason of lane keeping computation.
7. Wifi USB

3.2 Software Component

1. Python: it is an object-oriented high-level general-purpose programming language created by Guido van Rossum [11] and first released in 1991. It is an important programming language for Machine Learning as been fondly adopted by most Machine Learning Engineers.
2. OpenCV: OpenCV (Open source computer vision)is a python library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license [10]. OpenCV supports the deep learning frameworks like TensorFlow, Torch/PyTorch and Caffe. In this paper, OpenCV was deployed to read images into BGR (Blue/Green/Red) color space by default, instead of the more commonly used RGB (Red/Green/Blue)

color space. This contributes majorly to the computation oflane keeping module of the robotic Car.

3. Tensorflow – is one of the most used deep learning libraries due to its production ready features, flexible ecosystems, libraries and community support [9]. It was developed by Google Brain team to perform symbolic math operations based on dataflow and differentiable programming [9]. Tensorflow is open source.
4. Google Colab: this is an online platform powered by Google that allows to write and execute python codes on-the-go. It requires no configuration, it allows free access to GPU for training machine learning models, and allows easy sharing of models, codes, and documents [13]. The deep learning model training was done on Google Colab.

4. SYSTEM DESIGN

4.1 Robotic Car Hardware Assemblage

The raspberry pi was setup on a windows computer using a Linux built virtual machine. Then it was mounted on the robotic car therefore, connecting the corresponding slots on the robotic car with those on the car chassis. The perception camera was installed on the raspberry pi. Necessary software and software libraries (Python, OpenCV, and Tensorflow) were installed.

4.2 Deep Learning for Lane Detection and Navigation

Lane detection was computed manually using OpenCV. A make-believe environment like a road with lane lines was set up. Different tape colors (blue, yellow, and green) were used as the lane lines as against other approaches that used a single color [16], to demarcate the left, and right side of the road. Different colors were used in order to have varieties of colors in the data for training the deep learning model. In addition, all the three colors are unique color in the laboratory environment and as such color subtraction ad segmentation was easily computed.

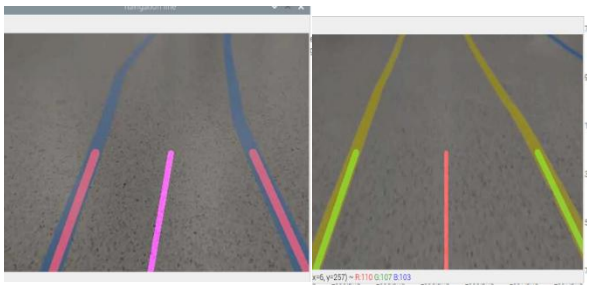
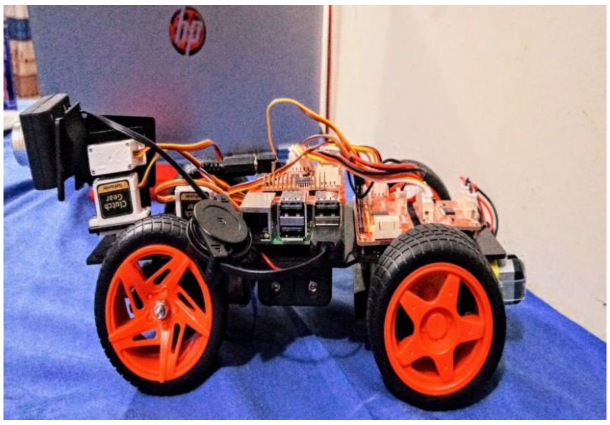
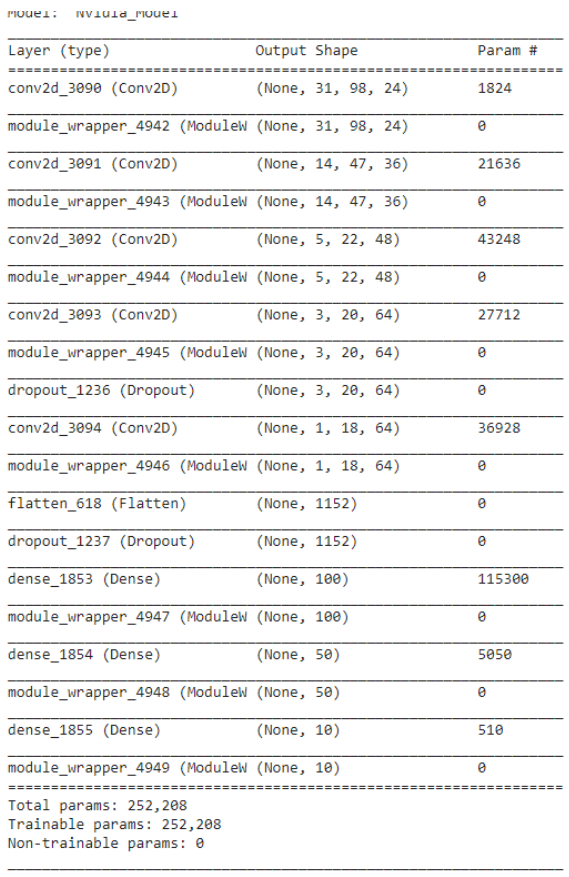


Figure 2: Detected Lane lines using conventional image processing algorithm.

To collect data for training, the robotic car was driven manually, while the perception camera captures the lane lines and the environment. A conventional computer vision algorithm of color thresholding was used to detect the lane lines. A color mask and edges were computed, corresponding to getting the region of interest in the picture frame. The centre of the lane line is equivalent to 90^0, therefore, placing the car in the middle of the road (centre), 0 -89^0 means steering the car to left, and 91 – 180^0 means steering the car to the right. The frames obtained from the video were heavily preprocessed via data augmentation techniques (zooming, blurring, cropping, and panning) in order to generate more data for training. A total of 1005 images was gathered comprising of blue, green, and yellow lane lines under varying light condition.

A modification of Nvidia model [15] was trained. The input to the CNN is a 66 x 200 x 3. The network originally consists of 9 layers made up of 5 convolutional layers, a normalization layer, and 3 fully connected layer. However, batch normalization was part of the data pre-processing during, hence eliminating the normalization layer in the network trained as seen in figure. Total parameters for training are 252,208, 11 parameters of the Nvidia model [15]. The network outputs steering angle given an input image.The model





```
Model: "Nvidia_Model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3090 (Conv2D)         (None, 31, 98, 24)        1824
_________________________________________________________________
module_wrapper_4942 (ModuleW (None, 31, 98, 24)        0
_________________________________________________________________
conv2d_3091 (Conv2D)         (None, 14, 47, 36)        21636
_________________________________________________________________
module_wrapper_4943 (ModuleW (None, 14, 47, 36)        0
_________________________________________________________________
conv2d_3092 (Conv2D)         (None, 5, 22, 48)         43248
_________________________________________________________________
module_wrapper_4944 (ModuleW (None, 5, 22, 48)         0
_________________________________________________________________
conv2d_3093 (Conv2D)         (None, 3, 20, 64)         27712
_________________________________________________________________
module_wrapper_4945 (ModuleW (None, 3, 20, 64)         0
_________________________________________________________________
dropout_1236 (Dropout)       (None, 3, 20, 64)         0
_________________________________________________________________
conv2d_3094 (Conv2D)         (None, 1, 18, 64)         36928
_________________________________________________________________
module_wrapper_4946 (ModuleW (None, 1, 18, 64)         0
_________________________________________________________________
flatten_618 (Flatten)        (None, 1152)              0
_________________________________________________________________
dropout_1237 (Dropout)       (None, 1152)              0
_________________________________________________________________
dense_1853 (Dense)           (None, 100)               115300
_________________________________________________________________
module_wrapper_4947 (ModuleW (None, 100)               0
_________________________________________________________________
dense_1854 (Dense)           (None, 50)                5050
_________________________________________________________________
module_wrapper_4948 (ModuleW (None, 50)                0
_________________________________________________________________
dense_1855 (Dense)           (None, 10)                510
_________________________________________________________________
module_wrapper_4949 (ModuleW (None, 10)                0
=================================================================
Total params: 252,208
Trainable params: 252,208
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: Training parameters of the adopted Nvidia Model

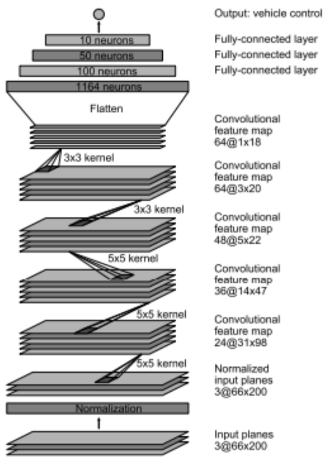


Figure 4: Nvidia CNN model [15]

4.3 Deep Learning for Traffic Sign Identification

Aside being able to navigate the road by performing inference on deep learning model for the computation of the steering direction, the robotic car should possess traffic and pedestrian intuition just as a normal human driver. Therefore, to achieve this, a deep learning-based object detection was applied. Most object detection problems are modelled as a classification problem, and as such the data requires accompanying label to point the images to their corresponding classes. Likewise, in focus, the problem of traffic signs, object recognition, and pedestrian for the robotic car has been modelled as a classification problem.

Potentially, to train an objection detection model requires a large quantity of data, consisting of several objects (signs in this case) to be detected. However, transfer learning enables for training a model with little quantity of data, provided there is a pre-trained network available for modification. In addition, with transfer learning, the first few layers of a deep neural network have already learned essential internal features of images, hence only the last 2 layers are to be modified for the task at hand.

Images of hand-drawn signs and pedestrian were collectedfor data, see figure 6 below for samples. Data augmentation as was performed for lane steering was also applied to the images for robustness. Anopen-source tool was used to draw bounding boxes around the images, in order to extract the region of interest's x and y coordinate. 205 images were collected altogether with their corresponding labels.Mask R-CNN model was used because of its robustness and generalization for object detection tasks [17]. The output model was able to generalize well, even for unseen data.

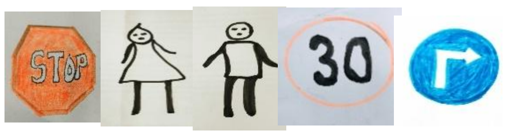


Figure 5: Traffic signs, pedestrians and speed limit

4.4 Algorithm for Pedestrian and Traffic signs Detection

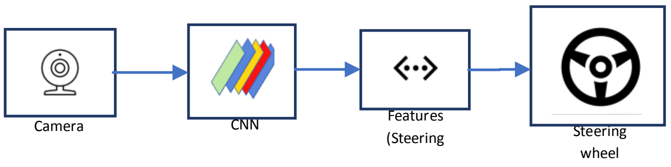


Figure 6: Blockdiagram of lane steering intuition

```
initial steering: 90 ➔ centre
start_speed = 0
get_image:
with lane_steering_model (image):
        predict lane_steering:
                robotic_car_: lane_steering
```

5. DISCUSSION

The model was trained with only 1005 images, under varying lightning condition. To measure the performance of the model, the data was initially split into 80-20 of training to testing (for validation) ratio. The model was trained for 20 epochs, As seen in the plot in figure 5, the model performed well. The output of the lane steering navigation is a steering angle. A simple yet efficient algorithm was developed to steer the robotic in the lane given the real timeline detection.

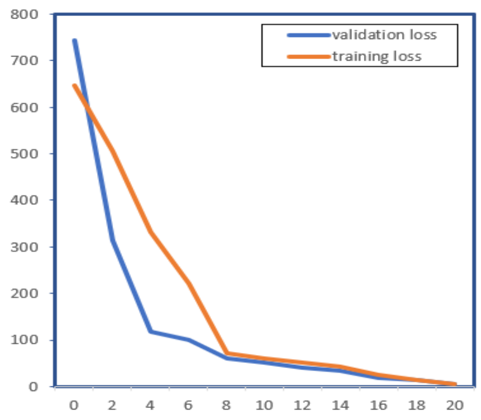


Figure 7: Plot of validation loss and training loss

The object classifier trained, was able to perform real time detection, but at the expense of slowing down other processes. This is so, because Raspberry pi is limited in computational power and could not perform inference fast enough using the trainedmodel.To fill the limitation created by raspberry pi, a TPU device was employed. Google Edge TPU, allows for running machine learning model without the need of expending large compute power.

The robotic car drives autonomously without an outside help using the output of the Nvidia model (lane steering prediction). Although the object recognition model is deep learning based, the algorithm for decision making for traffic sign and pedestrian detection was hand crafted.The car stops at the detection of pedestrian and stop sign, turns left or left at the detection of the signs, and slows down at the detection of number 30, 40, and 50 (to the corresponding speed).

6. CONCLUSION

Although, the design and development of real autonomous vehicles is complex than the context of this research, as a lot of work must go into safety, precision, and development of algorithms. Application of deep learning to the motion planning and driving the car (howbeit autonomously) would eliminate unnecessary downtime and result in realizable results.The adaptation of using deep learning to heavy automobiles would prove even more effective with other sensors in addition to dash camera as the perception. With many input perception sensors, the accuracy and response of the autonomous systems would be optimized. The CNN has proven to be efficient with low loss, and efficient use of training time to get accurate results [16]. With the use of image augmentation, the use of excess data collection on a pre-trained model is eliminated. With this technique, there is time optimization, and the bulk of the time could be channeled into hardware precision [16].

Finally, this paper embodies vision systems with deep neural network to learn features that would ordinarily be difficult for conventional vision algorithm to attain. The approach, and model developed could be adapted to any form of decision-making problem. Deep learning is receiving major attention from researchers, engineers and top tech companies, and its application would soon find its place in every sector.

7. REFERENCES

[1] Matthias H. "Self-Driving Vehicles in Logistics". DHL Trend Research. 2014.

[2] Robotics Business Review. "Infographic: Hopes and Fears Around Self-Driving Cars". 2019. [Online]. Available: https://www.roboticsbusinessreview.com/unmanned/unmanned-ground/infographic-hopes-and-fears-around-self-driving-cars/

[3] The Guardian. "Arizona Suspends Uber's self-driving car testing after fatality". 2018. [Online]. Available: https://www.theguardian.com/technology/2018/mar/27/arizona-suspends-ubers-self-driving-car-testing-after-fatality

[4] Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. NIPS'12 Proc 25th Int Conf Neural Inf Process Syst 1:1097–1105

[5] O' Mahony N, Murphy T, Panduru K, et al (2016) Adaptive process control and sensor fusion for process analytical technology. In: 2016 27th Irish Signals and Systems Conference (ISSC). IEEE, pp 1–6

[6] Towards Data Science. "Introduction to Convolutional Neural Network (CNN) using Tensorflow". 2020. [Online]. Available: https://towardsdatascience.com/introduction-to-convolutional-neural-network-cnn-de73f69c5b83

[7] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[8] Towards Data Science. "A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way". 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[9] Tensorflow. "An end-to-end open source learning platform". 2021. [Online]. Available: https://tensorflow.org

[10] OpenCV. "OPENCV" 2021. [Online]. Available: https://opencv.org

[11] Computer History Museum. "Guido Van Rossum". 2018. [Online]. Available: https://computerhistory.org/profile/guido-van-rossum/

[12] Google Cloud. "Edge TPU". 2019. [Online]. Available: https://cloud.google.com/edge-tpu

[13] Google Colab. "Welcome to Colaboratory". 2019. [Online]. Available: https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index

[14] PIMORONI. "Raspberry Pi 4". 2019. [Online]. Available: https://https://shop.pimoroni.com/products/raspberry-pi-4?variant=31856486416467

[15] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, Karol Zieba: End to End Learning for Self-Driving Cars. CoRR abs/1604.07316 (2016)

[16] CV ZONE. "Computer Vision Zone.". 2020. [Online]. Available : https://computervision.zone

[17] He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).