

A REVIEW OF DEEP LEARNING FOR SELF-DRIVING CARS: CASE STUDY

Author(s): **Joel Murataj**^a, **Prof. Dr. Eng. Abdulsalam Alkholidi**^{b,c}, **Prof. Dr. Habib Hamam**^d,
PhD Afrim Alimeti^e

^a Software Engineering Department, Canadian Institute of Technology, Albania

^b Software Engineering Department Canadian Institute of Technology, Albania

^c Electrical Engineering Department Faculty of Engineering – Sana'a University - Yemen

^d Département de Génie Électrique, Faculté D'Ingénierie, Université de Moncton, Canada

^e Industrial Engineering Department, Canadian Institute of Technology, Albania

Abstract

Deep Learning (DL) is a subfield of Machine Learning (ML) that deals with algorithms inspired by the structure and function of the brain. DL uses complex algorithms and deep neural nets to train a model. It consists of the learning of artificial neural networks that consider algorithms inspired by the human brain by learning how to use a large amount of data. It includes machine learning, where machines can learn by experience and get skills without human intervention. The importance of deep learning is the ability to process a large number of characteristics allowing deep and powerful learning when dealing with ambiguous data. This paper aims to study and analyze to be updated existing papers related to the deep learning field and introduce our contribution. An additional aim of this review paper is to concentrate on the self-driving cars case study and introduce the new approach with high performance.

Keywords: *Neural Network, Architectures, Big Data, Machine Learning, Autonomous Driving*

1. Introduction

We start this study by mentioning an Artificial Neural Network (ANN) is a part of computer architecture or system that is designed to emulate the way information is analyzed and interpreted by the human brain. It is the basis of artificial intelligence (AI) and solves problems that are hard or impossible to be done by human or statistical standards. The main remarkable difference between deep learning and classical machine learning is its execution with a large size of data. When data is few, deep learning algorithms do not proceed well. This is due to deep learning algorithms demanding a large quantity of data to recognize them. We can introduce the most important examples or applications of deep learning as follows: Machine translation, virtual assistants, visual recognition, self-driving cars and drones, robotics, facial recognition, medicine and pharmaceuticals, customer service, fraud detection, news aggregation, digital marketing, natural language processing, colorizing videos and images, military, and health care. The evolutionary history of DL started several years ago.

The limitations of deep learning are data, computational power which has of cores as compared to CPUs, and training times. Artificial Intelligence means enabling the machine to think while ML has the statistical tool that explores and analyzes the data and can be supervised by the past labeled data and considered as a subset of AI, unsupervised clustering, and reinforcement semi supervising. Machine learning was developed from object recognition in essence. It assumes that smart machines should be able to learn and adjust to their climate through experience [1-3]. Deep learning is a subset of ML and creates multi neural networks architecture and has various techniques as Artificial Neural Network, Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). However, Data Science (DS) technique applies all previous ones and uses some mathematical tools like statistics, probability, linear algebra, and differential calculus. An artificial neural network consists of several interconnected, functional elements, or neurons that perform and process the information by using parallel information-processors, to solve and classify problems by regression method. The idea is to split the feed in data into discrete value or classes that maps inputs to outputs.

DL is based on the functioning of the human brain. Let's understand how a Biological Neural Network (BNT) looks like (dendrite, cell nucleus, axon, and synapse) where Artificial Neuron contains input, nodes, weights, and output. There are several steps to process the perceptron learning model are given as demonstrated in Fig. 1:

- **Step 1** calculating the weighted sum of the inputs.
- **Step 2** passes the calculated weighted sum as in the input to the activation function to generate the output [4-5].

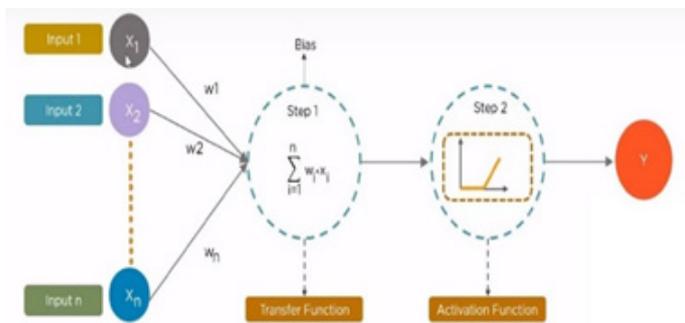


Fig. 1. The Perceptron Learning Model.

The inputs could be pixels values of an image ($x_1, x_2, x_3, \dots, x_n$), these values can be 256 or 512 pixels.

- **Step 3** all inputs values are multiplied by the weight ($w_1, w_2, w_3, \dots, w_n$) and are considered as numeric values and could be any value.

In the transfer function

- The weighted sum sigma is used as a central unit

$$\sum_{i=1}^n w_i * x_i$$

- Then bias is added where the basis is common for each neuron (1 basis/ neuron).

- **Step 4** in the activation function:

The output (step 1) of the transfer function feeds the activation function.

An activation function takes the “weighted sum of input plus the bias” as the input to the function and decides whether it should be fired or not. A standard integrated circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on the input. Several types of activation functions exist, namely: Sigmoid

Function, Threshold Function, ReLU Function, and Hyperbolic Tangent Function, and the output differ from one activation function to another. Moreover, the activation function requires nonlinearity. The most used activation function is the Sigmoid Function because it exists between (0 to 1). It is used for models where we have to predict the probability as an output. It exists between 0 and 1 as shown in Fig, 2.

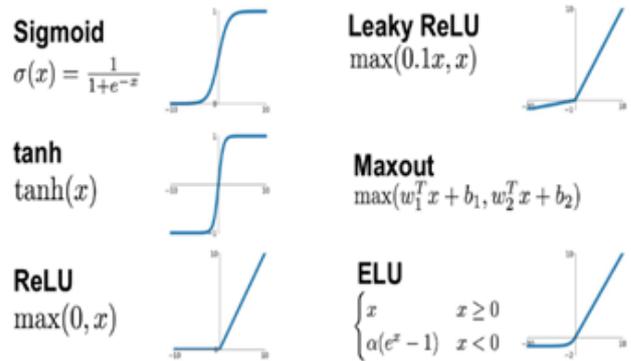


Fig. 2. Activation functions [6].

There are three layers: input layer, hidden layer, and output layer, respectively.

If we compare between traditional machine learning and deep learning algorithms, the last one's performance is increasing the amount of data while the traditional one becomes approximately constant when it reaches the threshold of training data [7]. Top applications of deep learning are in the following fields, namely: in customer support, medical care, self-driving cars (Apple, TESLA, and NISSAN, etc), medical image processing, natural language processing, visual recognition, digital marketing, automatic machine translation healthcare, fraud detection, language translations, and pixel restoration photo descriptions, demographic and election predictions, deep dreaming, and robot navigation. This study will include some of these applications.

DL is growing fast but has some limitations:

- Data: where DL needs a massive volume of data to train.
- Computational power: training and neural network required graphical processes in units which have thousands of cores as computers to CPUs that's more expensive.
- Training time: the time increasing with the

amount of data and numbers of layers in the networks

We need deep learning to:

- Process huge amounts of data: ML algorithms work with a huge amount of structured data but DL algorithms can work with an enormous amount of structured and unstructured data.
- Perform complex algorithms: ML algorithms cannot perform complex operations. To do that we need DL algorithms.
- Achieve the best performance with a large amount of data: As the amount of data increases, the performance of ML algorithms decreases. In order to make sure that the performance of a model is good, we need DL.
- Features extraction: ML algorithms extract patterns based on labeled sample data, while DL algorithms take large volumes of data as input, analyze the input to extract features out of an object, and identify similar objects.

Deep learning approaches are:

- Supervised learning where the input variables introduced as X is structured to the output values that are represented by Y using the following function $Y=f(X)$ [8].
- Unsupervised Learning (UL) analyzes and clusters unlabeled data sets using machine learning methods. These algorithms find hidden patterns in data without the assistance of humans (thus the term "unsupervised").
- Reinforcement learning (RL) is a branch of machine learning that studies how intelligent agents should operate in a given environment to maximize the concept of cumulative reward.
- Hybrid Learning (HL) applies to systems that allow the use of both generative (unsupervised) and discriminatory (supervised) elements.

An artificial neural network is the functional unit of deep learning, using artificial neural networks which mimic the behavior of the human brain to solve complex data-driven problems. Now, deep learning itself is a part of machine learning which falls under the larger umbrella of artificial intelligence. Artificial intelligence machine learning and deep learning are interconnected

fields where machine learning and deep learning aid artificial intelligence by providing a set of algorithms and neural networks to solve data-driven problems. Deep learning uses artificial neural networks that behave similarly to the neural networks in the human brain. A neural network functions when some input data is fed to it. This data is then processed via layers of perceptron's to produce the desired output.

2. Literature Review

In recent times many researches have been published toward developing deep learning algorithms. The most reported are shown in [5, 9-20]. The IEEE transactions article developed by Saptarshi Sengupta et al. [5] mentioned that several problems were solved and developed by using deep learning algorithms in the last decade - the automated identification of data patterns; and it can do so with a precision that far exceeds that of human beings. Outside the world of conventional, handcrafted computers, it has solved problems algorithms of learning about aching of patients for practitioners attempting to make sense of the data flow that is now inundating our society.

In the paper developed by Katleho L Masita et al., they discussed some of the most relevant and recent advances and contributions made to the study of the use of deep learning in object detection. In addition, as seen, the results of multiple studies indicate that the application of deep learning in object detection greatly exceeds traditional methods based on handcrafted and learned characteristics [9]. Another paper published by Aman Bhalla et al. [10] illustrated a computer vision model that learns from video data is proposed in this research work. It includes image processing, image augmentation, behavioral cloning, and neural network model convolution. The architecture of the neural network used to detect paths in a video segment, road linings, obstacle positions, and behavioral cloning is used for the model to learn in the video from human behavior.

We summarize some researches published recently on deep learning as demonstrated in Table I.

TABLE I. SUMMARY OF SOME RESEARCHES IN DEEP LEARNING

Involved Authors	Findings & Conclusions	Ref
Saptarshi Sengupta et al	The article introduces an overview of multilayer artificial neural networks that contain deep learning. They also demonstrated some new automatic architecture optimization protocols that use multi-agent approaches. In addition, authors the combined swarm intelligence in Deep Learning approaches and how the influence of one enriches other when applied to real-world problems	[5]
Ben Bright Benuwa et al.	This article studies and analyzes the principles regarding learning algorithms for deep architectures. In addition, this study shows the prior work conducted in deep learning. There still remains a great deal of work to be done in improving the learning process.	[11]
Xiaoxuan Liu	This study explores the diagnostic accuracy of deep learning algorithms versus healthcare professionals in classifying diseases using medical imaging. This paper improved in quality of published previously researches. 58 (71%) of the 82 studies satisfying the inclusion criteria were newly identified in the updated search, suggesting that the past year has seen a substantial increase in the number of studies comparing algorithm accuracy with healthcare professionals.	[12]
Ajay Shrestha, and Ausif Mahmood	The study reviews some optimization methods to develop the accuracy of the training and minimize training time. The authors search into the mathematics beyond training algorithms used in existing deep networks. They tested some state-of-the-art ways to overcome these challenges with different optimization algorithms.	[13]
Antonio Hernández-Blanco et al.	The main objective of this study is to distinguish the educational data mining tasks that have benefited from Deep Learning of recently published research and to illustrate some of the existing state-of-the-art and setting the future aspects of this important field of research.	[14]
Mohamed (Behada) Jamshidi et al.	The main goal of the suggested structure is to enhance the accuracy and hasten recognition and classification of the issues created by the virus by utilizing DL-based methods. The proposed structures and platforms in this study are suitable for dealing with COVID-19 issues. Several methods have been developed, incorporating COVID-19's diagnostic systems, such as LSTM, GAN, ELM, and RNN.	[15]
Jürgen Schmidhuber	This study is reviewed deep supervised and unsupervised learning, reinforcement learning and evolutionary computation, and an indirect quest for deep and wide networks encoding short programs. The main goal of this study is to assign credit to those who contributed to the present state of the art.	[16]
Yann LeCun et al.	The most used form of ML, deep or not, is supervised learning. Authors want to build a system that can classify images that contain: a house, a car, a person, or a pet. The first step is to collect a large data set of related objects, each labeled with a specific category. During training, the machine shows an image and produces an output in the form of a vector of scores, one for each category.	[17]
Reem Ibrahim Hasan et al.	This review paper shows the recently published studies, developed over three years leading up to 2020, for training, increasing, feature fusion and extraction, crop recognition and counting, and plant disease detection. It includes how these techniques can be used to feed deep classifiers and their effects on the accuracy of classifiers.	[18]

During the last decade, self-driving vehicle technology has developed remarkably despite some obstacles encountered, that will be inevitably overcome shortly. The second part of this study will focus on this self-driving vehicle algorithm. This means that we will introduce the recently published studies as related work. Then we will introduce our contribution through the series of simulations we released. The article published by Sorin Grigorescu et al. has introduced an overview of deep learning techniques using

autonomous driving. The performance survey and computational requirements serve as a system-level design reference for AI-based self-driving cars [19], where authors started by introducing self-driving AI-based architectures, convolutional and repetitive neural networks, and a deep reinforcement learning model. The method is based on surveyed driving scene perception, path planning, behavior judging, and motion control algorithms. Researchers investigated both standard planning pipeline and perception, where

each method is built using deep learning methods, and End2End systems as well, which directly map sensory information to steering commands that connect sensory information directly to directional commands.

In the article developed by Jelena Kocić et al. [20], a single solution for an end-to-end autonomous driving deep neural network is presented. The main goal of our work was to achieve autonomous driving using a light deep neural network suitable for deployment on compact car platforms. There are many end-to-end deep neural networks used for autonomous driving, where the input to the machine learning algorithm is camera images and the output is steering angle prediction, but these convolutional neural networks are significantly more complex than the network architecture that they proposed.

3. Simulations Results And Discussion Of The Proposed Approach: Self-Driving Car

Series of simulation results are released to show the performance of the proposed approach. Indeed, we concentrated on the following features namely: line lane detection, traffic signs detection, and steering angle. In this subsection, we are going to introduce each feature separately:

A. Lane Line Detection

Detecting lane lines in the road is very important for the car to drive. Using the camera in the front of the car can help to detect the lane line.

To identify the lines in the road, we can use edge detection and gradient. Edge detection is about sharp changes in intensity in adjacent pixels (a sharp color change). Gradient measures the change in brightness. There is a strong gradient (0 to 255 pixel) and a small gradient (255 to 0 pixel). Five steps are required:

- **Step 1:** To convert the image to grayscale; so, in one channel. It is processed faster than a normal image with three channels.
- **Step 2:** To reduce the noise, i.e. converting the colors to be smoothed (blurred).
- **Step 3:** To determine rapid changes in over image. Derivative $f(x,y)$ measures adjacent changes in all directions x and y . There are high threshold and low thresholds. If the gradient is between the

low and high thresholds, it is accepted only if it is connected to a strong edge (the rapid changes in brightness). The rapport between the thresholds is $1/2$ or $1/3$.

- **Step 4:** To create a triangle toward the view of the car, because we know that the lane lines are toward the car. Computing the bitwise of both images and taking the bitwise of each homologous pixel in both arrays ultimately makes the image only show the region of interest traced by the polygonal containing the task.

- **Step 5:** To detect straight lines and mainline. It finds the line which describes the points best.

The image presented in Fig. 3 is converted to grayscale color and smoothed.

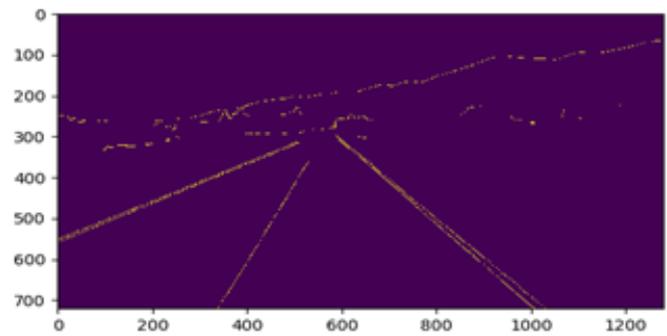


Fig3. Grayscale image.

The region of interest, which shows a triangle in front of the car as demonstrated in Fig. 4

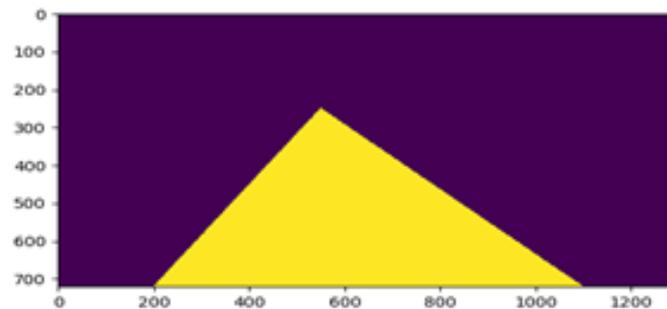


Fig. 4. Region of interest.

The bitwise function called compares by bytes the image and the triangle image and focuses only on the lane lines as shown in Fig. 5.

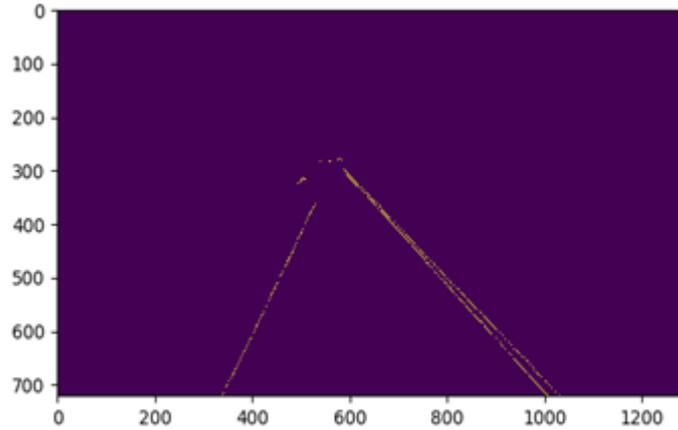


Fig. 5. Bitwise function.

Hough space can detect the line, by representing all the points in lines in the Hough space graph. One point can be represented by one line in Hough space. In this graph, we can see where the lines of the point are nearly each other. Those are represented by m and b values for the line. Therefore, a lane line can be created by using the “(1)”:

$$y = mx + b \quad (1)$$

To avoid the infinitive form “(1)”, we change the formula to [21]:

$$p = x \cos \alpha + y \sin \alpha \quad (2)$$

Figure 6 (a & b) introduces the Hough space: (a) Line created by the point; (b) Points represented in Hough space.

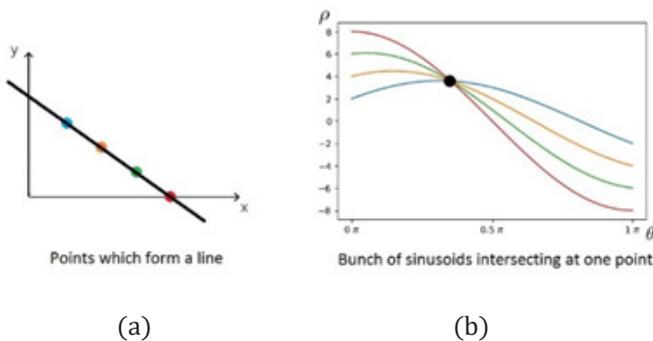


Fig. 6. Hough Space: (a) Line created by the point; (b) Points represented in Hough Space.

The Hough Space function detects the lane line during road all the time. For finding lane lines, the process just needs to work with pixels as demonstrated in Fig. 7.



Fig. 7. Detecting lane lines.

B. Traffic Signs Detection

Traffic sign detection can be achieved successfully by deep learning. Convolution neural network is the best for processing data. Fig. 8 shows the distribution of the training dataset.

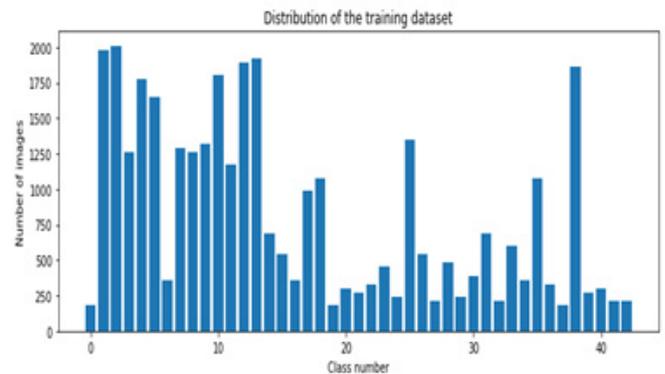


Fig. 8. Distribution of the training dataset.

Due to performance issues and in order to make edges between pixels more visible, the depth of the image is changed from three-dimensional to one-dimensional (change their color to gray) for all the images indicated in Fig. 9.

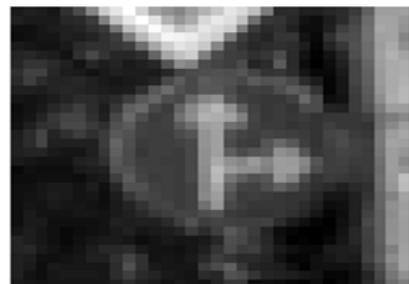


Fig 9. Gray image of a traffic sign

After changing the dimension of the image, for a clearer difference between the edges, we need to equalize the photo as demonstrated in Fig. 10. This helps the model find the features, as the model should focus on the edges of the image and differences of the pixels.

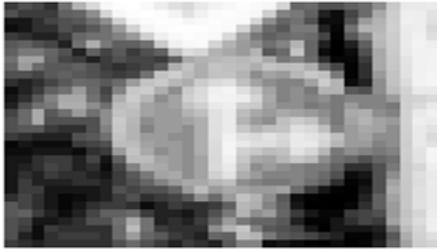


Fig. 10. Equalized traffic sign image

The entire dataset is preprocessed in the same way, before starting the training. A very important part diversifies, is data augmentation. Data augmentation is a widely used technique, which helps expand and diversify your training data, providing an added variety of important features, that the network needs to extract. It gives variety to the data and gives more data for the model to work and generalize the new data that can exist. Some of the training data consist of width shift and height shift (zoomed and rotated). This makes the model learn more and train in different data. We have used Keras for the augmentation of the data. The result is shown in Fig. 11.

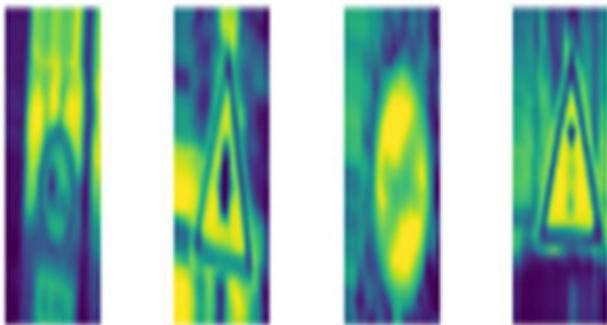


Fig 11. The result of the augmentation process

For the traffic sign, it used the leNet architecture. The input dimensions are 32 pixels width, 32 pixels height, and one depth. LeNet architecture is a convolution neural network structure. Google designed it. It is used in different applications and it is very influential even from the last obtained result. In this model, four convolution layers using the Relu activation function, two MaxPooling

layers, and an output layer using the Softmax activation function, as we have a multi-class categorization. Categorical Cross-Entropy is used as a loss function as we need to categorize multiple classes and check for the lowest error. After the data is processed by the model, it is classified into one category as numerical. Each number means a type of traffic sign. Figure 12 shows the progress of the training process. Epochs are the iteration, which is executed on the data. In the last epoch, the accuracy is 0.98 and loss is 0.05 for training data and 0.02 for validation loss, and 0.99 for validation accuracy. Those are very good results. It means that we have used the right model to train the data. It is difficult to find the right number of nodes and hidden layers. The result shows, that it is the best. We have tried a different number of layers until we got the best one.

```
Epoch 1/10
2000/2000 [*****] - 33s 16ms/step - loss: 0.8530 - accuracy: 0.7565 - val_loss: 0.0652 - val_accuracy: 0.9816
Epoch 2/10
2000/2000 [*****] - 33s 16ms/step - loss: 0.1960 - accuracy: 0.9385 - val_loss: 0.0442 - val_accuracy: 0.9862
Epoch 3/10
2000/2000 [*****] - 33s 16ms/step - loss: 0.1294 - accuracy: 0.9681 - val_loss: 0.0479 - val_accuracy: 0.9868
Epoch 4/10
2000/2000 [*****] - 33s 17ms/step - loss: 0.1079 - accuracy: 0.9663 - val_loss: 0.0513 - val_accuracy: 0.9878
Epoch 5/10
2000/2000 [*****] - 33s 17ms/step - loss: 0.0917 - accuracy: 0.9715 - val_loss: 0.0284 - val_accuracy: 0.9914
Epoch 6/10
2000/2000 [*****] - 33s 17ms/step - loss: 0.0796 - accuracy: 0.9756 - val_loss: 0.0320 - val_accuracy: 0.9921
Epoch 7/10
2000/2000 [*****] - 33s 16ms/step - loss: 0.0670 - accuracy: 0.9787 - val_loss: 0.0580 - val_accuracy: 0.9873
Epoch 8/10
2000/2000 [*****] - 33s 17ms/step - loss: 0.0648 - accuracy: 0.9884 - val_loss: 0.0880 - val_accuracy: 0.9864
Epoch 9/10
2000/2000 [*****] - 33s 17ms/step - loss: 0.0599 - accuracy: 0.9818 - val_loss: 0.0892 - val_accuracy: 0.9884
Epoch 10/10
2000/2000 [*****] - 33s 16ms/step - loss: 0.0578 - accuracy: 0.9831 - val_loss: 0.0238 - val_accuracy: 0.9937
```

Fig. 12. Training process. Screenshot by author.

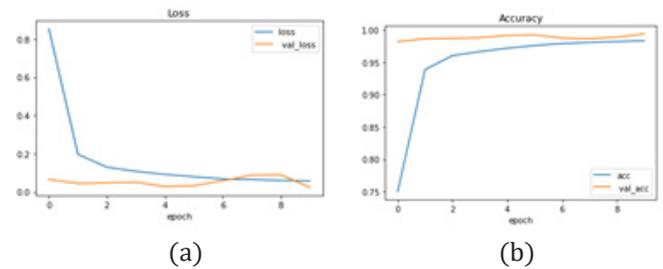


Fig. 13. Plots: (a) Loss plot; (b) Accuracy plot.

Figure 13 shows two plots to present the results. We created those two plots to see how the model has worked to the final epochs. It shows if the model is overfitted or underfitted. As the line loss and validation loss are not too far from each other in the last epoch (iteration), it means that the model is good. Loss and validation loss are reduced along the epochs and accuracy and validation accuracy are raised, it means that the model is learning the features well, as, in the end, this is what I need that the model learns the features of the image and

not memorize them. Gradient descent works on reducing the training error and regularization aim to reduce the generalization error.

C. Steering Angle Detection

For finding a steering angle, deep learning is the best. The model works with the data and learns the pattern from them. Steering Angle Detection is an important part of the self-driving car. It shows the car how to drive on the road. It is used a Udacity simulator. As the car has three cameras in the front (left, center, right), which capture images in every single step. Figure 14 shows an image taken from the application.



Fig. 14. Udacity.

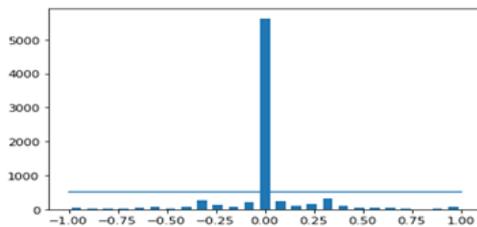


Fig. 15. Distribution of training dataset: No balanced dataset.

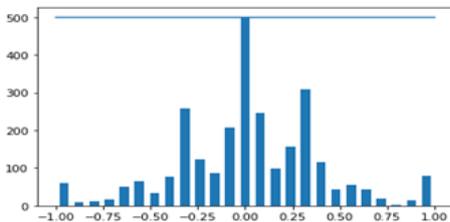


Fig. 16. Distribution of training dataset: Balanced dataset

This dataset Fig. 15 is not balanced, so the dataset, in this case, should be balanced firstly as demonstrated in Fig. 16 and some of the data will be removed.

The dataset should be divided into training dataset and validation set, where 20% is for validation dataset as presented in Figs. 17.

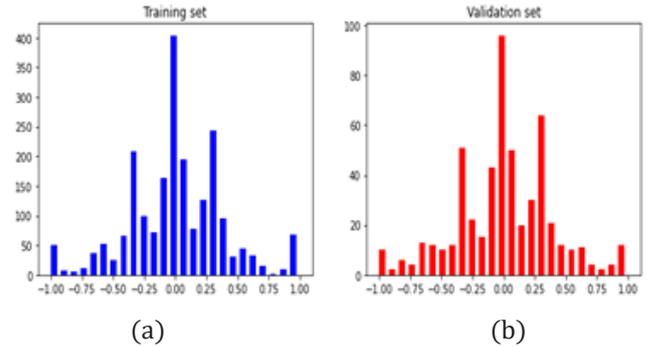


Fig. 17. Splitting dataset: (a) Training dataset; (b) Validation dataset.

To produce a variety of data and make the model focus on the pattern and not on memorizing the data. Data augmentation helps a generalization of the model. For example, an image can be flipped as shown in Fig. (18 b).

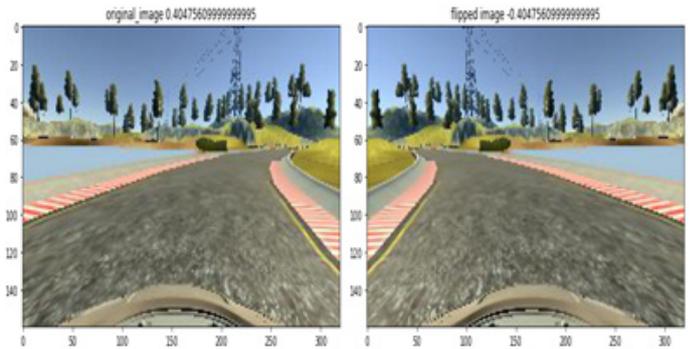


Fig. 18. Augmentation process: (a) The original image; (b) The flipped image.

Afterward, the image should be preprocessed, its brightness should be increased. Also, it should be smoother, with less noise, and resized to focus on the most important features for better performance, see Fig. (19 b).

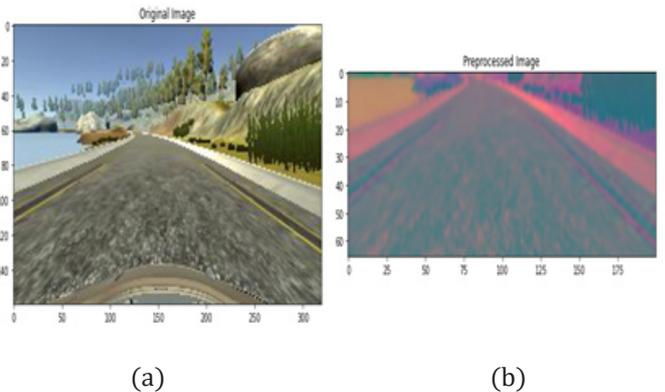


Fig. 19. Preprocessed process: (a) Original image; (b) Preprocessed image

```
Model: "sequential_10"
```

Layer (type)	Output Shape	Param #
conv2d_46 (Conv2D)	(None, 31, 98, 24)	1824
conv2d_47 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_48 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_49 (Conv2D)	(None, 3, 20, 64)	27712
conv2d_50 (Conv2D)	(None, 1, 18, 64)	36928
flatten_8 (Flatten)	(None, 1152)	0
dense_29 (Dense)	(None, 100)	115300
dense_30 (Dense)	(None, 50)	5050
dense_31 (Dense)	(None, 10)	510
dense_32 (Dense)	(None, 1)	11

Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0

Fig. 20. NVidia architecture.

The model shown in Fig. 20 contains five convolution layers, flatter to change the dimensionality to one for the three fully connected layers, and in the end is one output layer. As in this situation, we should output the predicted steering angle, which is between -1 to 1, we cannot use categorical in this case, but we can use Mean Square Error (MSE). MSE corresponds to how far the actual value is from the estimated value [22]. As we have large data and we need to learn more features from images, NVidia is the best model and a powerful neural network to be used. The NVIDIA DRIVE PX2 driverless car platform can perform 30 trillion deep learning operations per second and can achieve Level4 autopilot. NVidia is proven effective for behavioral cloning and it is implemented in real-life cars [23]. Mean Square Error validates how far the predicted value is from the predicted values. By checking the distance, it can help the model to know how large the error is and it shows the model that it needs more to learn because it is making wrong predictions as introduced in Fig. 21.

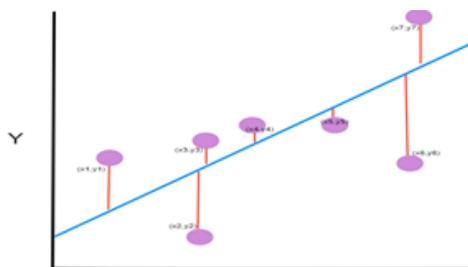


Fig. 21. Mean Square Error.

The formula for calculating the mean square error is [24]:

$$MSE = \frac{1}{N} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

Where variable N is several data points, the first y is the actual value and the second y is the estimated value.

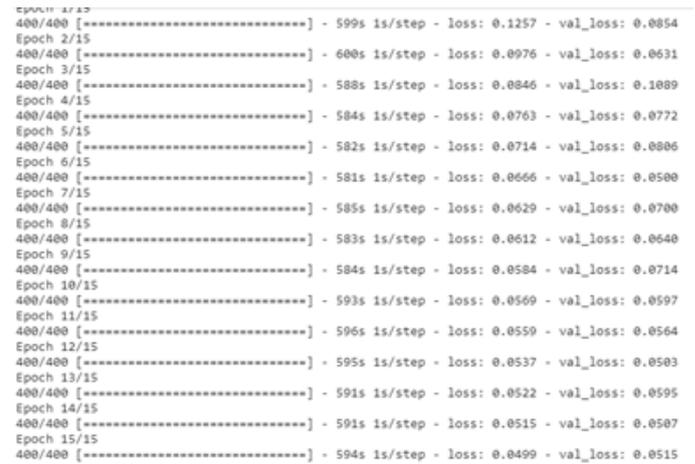


Fig. 22. History of processing the data: Model processing the data.

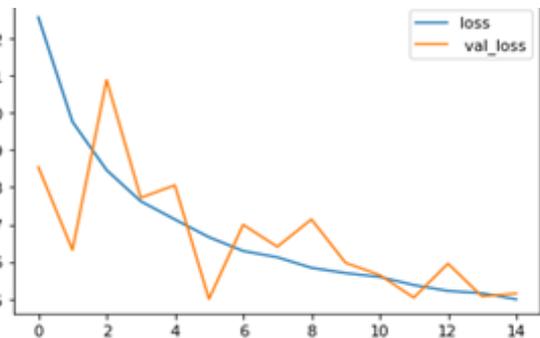


Fig. 23. History of processing the data: The result of the model.

The screenshot shown in Fig. 22 shows that the model is working well and the loss is reducing along with the validation loss. The plot shows that the loss has been reducing all the time in each epoch and validation loss has had an increase in epoch 3, because the model has had an error on not predicting well the inputs, therefore the model learns from this error and updates its parameter. In the last epoch loss and validation, loss is a small number, which means that the model has learned a lot and it has had a good result on predicting. The loss and validation loss are not too far away from each other, so the generalization error is low as shown in Fig. 23.



Fig. 24. Result of self-driving car. Screenshot by author.

The screenshot demonstrated in Fig. 24 shows a part of the self-driving car during the simulator.

4. Conclusion

This study showed the features of deep learning. It helps in solving different problems that an autonomous machine needs. This approach showed how self-driving car has evolved in different fields. The main companies that used this technique are in the race, which one will be the first to have a self-driving car in the market. We think it will take some more time for perfecting the self-driving car but I think it will be the best invention of the future. All the vehicles have cameras in their cars, which makes creating the dataset easier. Different self-driving test cars are around the world but they have done some accidents. If all the vehicles would be autonomous, from our perspective and from the presented studies it will be a good solution. That would make the cars communicate with each other and make the driving easier and safer. A self-driving car is a future and it should be implemented, as there are many human errors that have caused so many tragic accidents. Deep learning is so near to implementing the level 5 autopilot. Maybe in the future, the best approach would be combining supervised learning and unsupervised deep learning together. Therefore, after the generated model of supervised learning predicts the image, the supervised model comes into play and checks how well the prediction is done and if the car is going in the right direction. Proposed future work is implementing supervised and unsupervised working together and finding if this would work better and trying this model in real life and finding a better technology for sensors as well.

REFERENCES

- [1] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, Oct. 2000.
- [2] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of the International Conference on Machine Learning*, New York, NY, USA, July 2008, pp. 160–167.
- [3] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.
- [4] Rafael C. Gonzalez and Richard E. Woods, *Digital image processing*, page 884, third edition, Pearson Prentice Hall, 2018. <https://www.codecool.ir/extra/2020816204611411Digital.Image.Processing.4th.Edition.www.EBooksWorld.ir.pdf>
- [5] Saptarshi Sengupta et al., *A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends*. IEEE TRANSACTIONS ON XXX, VOL. XX, NO. YY, MARCH 2019.
- [6] <https://himanshuxd.medium.com/activation-functions-sigmoid-relu-leaky-relu-and-softmax-basics-for-neural-networks-and-deep-8d9c70eed91e>
- [7] Amitha Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview", In book: *Advanced Machine Learning Technologies and Applications*, DOI: 10.1007/978-981-15-3383-9_54, January 2021. (https://www.researchgate.net/publication/341652370_Deep_Learning_Techniques_An_Overview#fullTextFileContent).
- [8] Amitha Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview", In book: *Advanced Machine Learning Technologies and Applications*, DOI: 10.1007/978-981-15-3383-9_54, January 2021. (https://www.researchgate.net/publication/341652370_Deep_Learning_Techniques_An_Overview#fullTextFileContent).

- [9] Katleho L Masita; Ali N Hasan; Thokozani Shongwe, "Deep Learning in Object Detection: a Review", IEEE, icABCD, DOI: 10.1109/icABCD49160.2020.9183866, 2020.
- [10] Aman Bhalla, Munipalle Sai Nikhila, and Pradeep Singh, "Simulation of Self-driving Car using Deep Learning", IEEE, 3rd International Conference on Intelligent Sustainable Systems (ICISS), DOI: 10.1109/ICISS49785.2020.9315968, 2020.
- [11] Ben Bright Benuwa et al., "A Review of Deep Machine Learning", International Journal of Engineering research in Africa, ISSN: 1663-4144, Vol. 24, pp 124-136, doi:10.4028/www.scientific.net/JERA.24.124, Trans Tech Publications, Switzerland, 2016.
- [12] Xiaoxuan Liu et al., "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis", Published by Elsevier Ltd., pp. e271-e297, This is an Open Access article under the CC BY 4.0 license., www.thelancet.com/digital-health Vol 1, October 2019.
- [13] Ajay Shrestha, and Ausif Mahmood, "Review of Deep Learning Algorithms and Architectures", IEEE. Translations, pp. 1-29, 10.1109/ACCESS.2019.2912200, IEEE Access http://www.ieee.org/publications_standards/publications/rights/index.html, 2017.
- [14] Antonio Hernández-Blanco et al., "A Systematic Review of Deep Learning Approaches to Educational Data Mining", WILEY Hindawi, Volume 2019, Article ID 1306039, 22 pages, pp. 1-22, <https://doi.org/10.1155/2019/1306039>, 2019.
- [15] Mohamed (Behada) Jamshidi et al., "Artificial Intelligence and COVID-19: Deep Learning Approaches for Diagnosis and Treatment", IEEE Access, Vol. 8, pp. 109581-109595, <https://creativecommons.org/licenses/by/4.0/>, 2020.
- [16] Jürgen Schmidhuber, "Deep learning in neural networks: An overview", ELSEVIER, Neural Networks, pp. 85–117, 2015.
- [17] Yann LeCun et al., "Deep learning", Macmillan Publishers Limited, Vol. 521, pp. 436-444, doi:10.1038/nature14539, 2015.
- [18] Reem Ibrahim Hasan, Review of the State of the Art of Deep Learning for Plant Diseases: A Broad Analysis and Discussion, Plants, pp. 1-25, doi:10.3390/plants9101302, 2020.
- [19] Sorin Grigorescu et al., A survey of deep learning techniques for autonomous driving. Computer Science, Machine Learning, Journal of Field Robotics, Online ISSN:1556-4967, 2019, DOI: 10.1002/rob.21918, arXiv:1910.07738 [cs.LG, 2020].
- [20] Jelena Kocić et al., An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms. Appl. Sci. 2020, 10, 4400; doi:10.3390/app10124400, pp. 1-27, 26 June 2020. www.mdpi.com/journal/applsci.
- [21] <https://medium.com/@tomasz.kacmajor/hough-lines-transform-explained-645feda072ab>.
- [22] Moshe Binieli. Machine learning: an introduction to mean squared error and regression lines (16 OCTOBER 2018). from freeCodeCamp website: <https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/>.
- [23] Jianjun Ni (Hohai University), Yanan Chen, Yan Chen, Jinxiu Zhu, Deena Ali, Cao Weidong (Hohai University). A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods. (16 April 2020) ResearchGate.
- [24] Moshe Binieli. Machine learning: an introduction to mean squared error and regression lines (16 OCTOBER 2018). from freeCodeCamp website: <https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/>.