Car Detection for Autonomous Driving using Deep Learning

Muhammad Amirul Hakimi Bin Zaprunnizam
Hochschule Hamm Lippstadt
Lippstadt, Germany
muhammad-amirul-hakimi.bin-zaprunnizam@stud.hshl.de

***Abstract*—One of the foundational components of autonomous vehicle technology is the image-based car detection algorithm. Modern deep learning techniques are very effective at detecting cars, but it is still tricky to recognize cars correctly in unfavorable circumstances like crowded roads and poor lighting. It's crucial to be able to deduce global contextual information from scant visual clues in order to be resilient in these difficult settings. In this research, we suggest a straightforward Proposed-Net-based car-detecting system. Utilizing the data set, the performance of this approach is assessed. The testing results were state-of-the-art and demonstrated that our method is resistant to obstruction and poor lighting.**

I. Introduction

The development of self-driving vehicle systems based on machine learning and artificial intelligence technologies is currently a hot topic of research. Although a fully functional self-driving car is still some time away, the majority of the world's major automakers are in the advanced stages of development. However, state-of-the-art of autonomous driving is prohibitively expensive because it frequently relies on specialized infrastructure and technology. For example, we are going to use LIDAR for navigation, GPS for localization, and LRF for obstacle detection. The goal of this research is to learn to drive by identifying cars using only a single camera.

Deep learning has lately acquired prominence in research and has been integrated into a wide range of applications, as well as demonstrating its potential in real-world scenarios over time. DL uses transformations and graph technologies in tandem to generate multi-layer learning models. The most recently developed DL approaches have achieved excellent performance across a wide range of applications. In the medical profession, for example, autonomous vehicles, audio and speech processing, visual data processing, and natural language processing. Today's technological evolution is extremely rapid. One can consider how to build a system that can think like a person. As a result, the basic architecture for deep learning is inspired by the human brain. One example of CNN are AlexNet and ResNet.

In general, this paper will include an explanation of Deep Learning, an introduction to Proposed-Net and also example of deep learning with autonomous driving.

II. Deep Learning

In this section, I will introduce a basic explanation of deep learning. In the following subsection, you will find some of the components of deep learning.

Deep learning is a subset of machine learning, which is a subset of artificial intelligence as shown in Figure 1 below. A method called artificial intelligence allows a machine to imitate human behavior. Machine learning uses algorithms that have been taught with data to create AI. Deep learning is a subset of machine learning that takes its cues from the way the human brain is organized. This structure is referred to as an artificial neural network in the context of deep learning. Deep learning is better explained in this paper, along with how it differs from machine learning. [1]

- **Artificial Intelligence** - Engineering of making intelligent machines and programs.
- **Machine Learning** - Ability to learn without being explicitly programmed.
- **Deep Learning** - Learning based on Deep Neural Network.

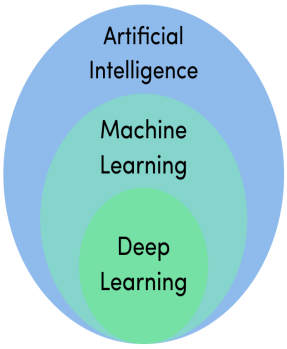

Fig. 1. Layer of deep learning [2]

For instance, we develop a device that can distinguish between tomatoes and cherries. If machine learning were used, we would need to provide the machine with the features that can be used to distinguish between the two. These characteristics may include the size and type of stem on them. On the other hand, with deep learning, the neural network

selects the features on its own without assistance from a person. Of course, having that level of independence comes at the expense of needing a lot more data to train our computer.

A. Neural Network

In this subsection, I will tell about neural networks function. As an example, we have three students and they each need to write number four on a piece of paper. Notably, they all write it differently. The numerals are simple for the human brain to recognize, but we are unsure that computers are able to define it as human. In this situation, we can use deep learning and it can play a role. This neural network was trained to recognize handwritten numbers. For example, each number is represented by a 28 by 28-pixel picture. that adds up to 784 pixels in total.

The central component of a neural network, the neuron, is where information processing happens [3]. A neuron in the first layer of our neural network receives data from each of the 784 pixels. Thus, the input layer is formed. On the other end, we have the output layer, where each neuron represents a digit and is connected via hidden layers. The information is transmitted through connecting channels from one layer to another. Each of them has a value associated with it, which is why it is referred to as a weighted channel.

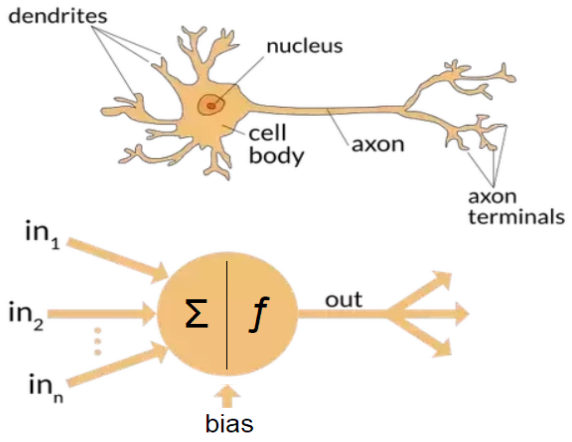

Fig. 2. Biological neuron used in neural network [4]

Every neuron has a distinct number called bias assigned to it. The weighted sum of inputs that reach the neuron are increased by this bias, which is subsequently used by a function called the activation function. If a neuron is activated, it depends on the outcome of the activation function as shown as Figure 2 above. Each active neuron transmits information to the layers below it. Up until the second-to-last layer, this continues. The input digit is represented by the one output layer neuron that was triggered. A well-trained network is created by regularly adjusting the weights and bias [3].

Now we are going to see where can we implement deep learning. When most people talk with customer service representatives, the dialogue feels so genuine. They aren't even aware that the other party is using a bot. Neural networks are used in medicine to analyze MRI pictures and find cancer cells by detecting them. Autonomous vehicles now, what formerly seemed like science fiction. Apple, Tesla, and Nissan are just a handful of the businesses developing autonomous vehicles. Therefore, deep learning has a broad application but also has significant drawbacks.

Data is the first, as we already covered. While neural networks may be trained on very little amounts of data, deep learning is the most effective method for handling unstructured data. Assume for the moment that we always have access to the required level of data processing. Not all machines are capable of doing this. Consequently, we arrive at our second constraint: computational power. Graphical processing units, which have thousands of them as opposed to CPUs and GPUs are obviously more expensive, are needed for training and neural networks. and now it's finally time for training. Training deep neural networks can take hours or even months. The length of time grows as the network's layer count and data volume do.

III. Convolution Neural Network

Convolution Neural Network (ConvNets or CNNs) is commonly used to build by the visual cortex in the human visual system in order to detect sub-regions from an input image. CNN is a structured version of this network that uses convolution operations to create multi-layer neural networks as shown in Figure 3. In fully linked networks, a high number of parameters are used to learn from incoming input. Color (RGB) images with dimensions of 300 pixels in width and 300 pixels in height, for example, would result in a learning network with 270,000 parameters. These networks are incredibly intricate. The convolution technique, on the other hand, allows CNNs to substantially reduce the number of parameters by applying a kernel or filter layer to the input images.

Convolutions neural networks are neural networks that are generally used to classify photos, cluster images based on similarity (photo search), and recognize objects within scenes. Convolution neural networks, for example, are used to recognize faces, individuals, street signs, cancers, and many other features of visual data. [5]

The effectiveness of convolutional networks in image identification is one of the primary reasons why the world has realized the value of deep learning. CNNs are, in some ways, the reason deep learning is well-known. CNNs are driving significant progress in computer vision (CV), which

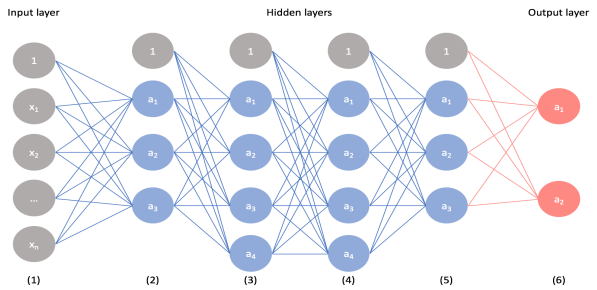

Fig. 3. Neural Network [4]

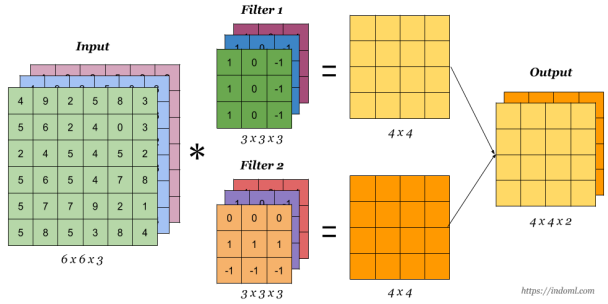

Fig. 4. Convolution Step [6]

has obvious implications in self-driving cars, robots, drones, security, and medical diagnosis.

A. How CNN works

Convolutional neural networks (CNN) are a form of neural network that was created to handle computer vision applications. This is because CNN's architectural design includes additional capability for dealing with pixels found in images. CNN enables picture encoding and network feed while lowering the parameters required to set up the model. Normal neural networks just lack the computational complexity needed to process picture data. While increasing the number of neurons and layers in a neural network to deal with individual pixels is an option, it will return us to the dilemma of over-fitting, in which the neural network fails to generalize to new unseen inputs.

CNNs have three main sorts of layers involved in their construction. Convolutional layers, pooling layers, and fully connected layers are the three types of layers. These layers are then piled in a particular order to create our CNN.

1) Convolution: As images pass through a convolutional network, they are described in terms of input and output volumes, which are expressed mathematically as matrices of multiple dimensions which is WidthxHeightxRGBChannel. As an example below it is 6x6x3. The next step is, the input will be filtered to 3x3x3 because that is the kernel size. Then the product of the multiplication is summed up to create a single scalar value. The end result after summing up the resulting product entry value to the output feature map [3].

Those are the initial, raw sensory characteristics given into the convolutional network, and the aim of the ConvNet is to determine which of those values are significant signals that help it identify images more accurately.

2) Pooling: The pooling layer reduces the number of parameters in that activation by sampling the input along the spatial dimension. The function of the pooling layer is to lower the spatial size of the convolved feature. Because of the reduction in dimensionality, the amount of computing resources required to process the data is reduced. This contributes to the model's practical training and the extraction of rotational and positional invariant leading characteristics. The two most frequent pooling strategies are max pooling and average pooling. . An example of pooling can be seen in Figure 5 below [3].

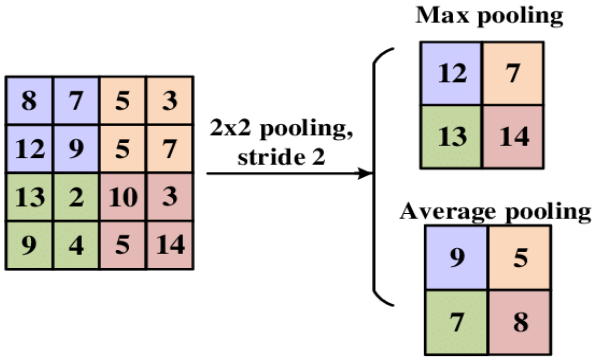

Fig. 5. Pooling [6]

3) Fully Connected Layer: The fully linked layer comes after the convolution and pooling layers. Both layers' output create high-level characteristics of the incoming image. After convolution and pooling, the fully connected layer is the next step for the neuron to train and identify the image. The fully connected layers will then attempt to produce output predictions [3].

B. Optimization

By identifying a local or global minimum, optimization functions help to avoid becoming stuck during the training stage [1]. The Adam optimization function, a popular approach, determines the global minimum.

C. Activation Function

Each Neural Network uses an activation function to process its input signal and bias. There are numerous forms of activation functions, which are either linear or nonlinear in structure [1]. The basic types of activation functions are as follows:

- Sigmoid function

Condition	Vehicle	Non-Vehicle
Sunny	20%	20%
Cloudy	20%	20%
Clear	20%	20%
Poor	20%	20%
Rain	10%	10%
Low-resolution camera	5%	5%
Tunnels	5%	5%

TABLE I
DISTRIBUTION OF DATA SET

- ReLU function
- Softmax function

IV. IMPLEMENTATION

A. Data Set

A group of researchers in Madrid that is called Grupo de Tratamiento de Imagenes (GTI) has a focus on vision-based vehicle categorization tasks. To analyze this method, they have recorded sets of various vehicles and derived them into images [7].

In total, they have 7325 images captured, where 3425 images captured are vehicles rears from and 3900 images are roads that do not contain any vehicle. The orientation of the vehicle adjacent to the camera is a key factor influencing the image of the car's rear. The data set divides the images into 4 sections of the range which are:

- Moderate
- Close and Left
- Close and Right
- Far

Several cases of the same vehicle with different boundary hypotheses are provided. The photos are 64x64 and clipped from 360x256 pixel sequences captured on motorways in Spain, Belgium, and Italy.

The entire group of photographs is chosen to cover a wide range of driving circumstances, particularly those related to weather. Table I shows how the data sets are compiled. There are several conditions captured in order to make sure it is suitable to perform in robust conditions. 2000 images are divided equally for vehicle and non-vehicle. This is a binary categorization project that divides the world into two categories: vehicles and non-cars. As seen in Figure 6, the automobiles have a label of 1.0, while the non-cars have a label of 0.0.

B. Model Created

After the data set has been compiled, a proposed model was created. I have decided to make my own model to avoid any heavy computation cost. In the model, it has 4 layers as shown as in Figure 7. The first 3 layers of convolution was created with kernel size 3x3 and the last layer is for classifier and

Fig. 6. GTI Datasets

predicted the probability. Also, in the last layer, max pooling is included. In the model, a dropout was also added to minimize the risk of overfitting. This model is suitable to use for binary classification and small training samples.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lambda (Lambda)             (None, 64, 64, 3)         0

 cv0 (Conv2D)                (None, 64, 64, 16)        448

 dropout (Dropout)           (None, 64, 64, 16)        0

 cv1 (Conv2D)                (None, 64, 64, 32)        4640

 dropout_1 (Dropout)         (None, 64, 64, 32)        0

 cv2 (Conv2D)                (None, 64, 64, 64)        18496

 max_pooling2d (MaxPooling2D  (None, 8, 8, 64)         0
 )

 dropout_2 (Dropout)         (None, 8, 8, 64)          0

 fcn (Conv2D)                (None, 1, 1, 1)           4097

=================================================================
Total params: 27,681
Trainable params: 27,681
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. Architecture Layer

C. Training Phase

After I have created the model, I trained the model to see how this model works. The proposed model was trained using the GTI dataset. The distribution of data is 80% for training and 20% for validation. The number of epochs is set to 5, 10, and 12 during the training phase. This is done to get the best result of training. Adam is the optimizer that was used. The training accuracy and validation result are shown in Table II.

Meanwhile, in Figure 8, an accuracy and loss graph was created to see the behavior of the model. The orange line on the graph represents test accuracy, while the blue line

Model	ResNet		
No. epochs	5 epochs	10 epochs	12 epochs
Training Accuracy	95.81%	96.97%	97.56%
Validation Accuracy	97.27%	97.54%	97.14%

TABLE II
ACCURACY RESULT FOR MODEL

represents train accuracy. We can see that the accuracy value are nearly identical, indicating that the model is well-trainec

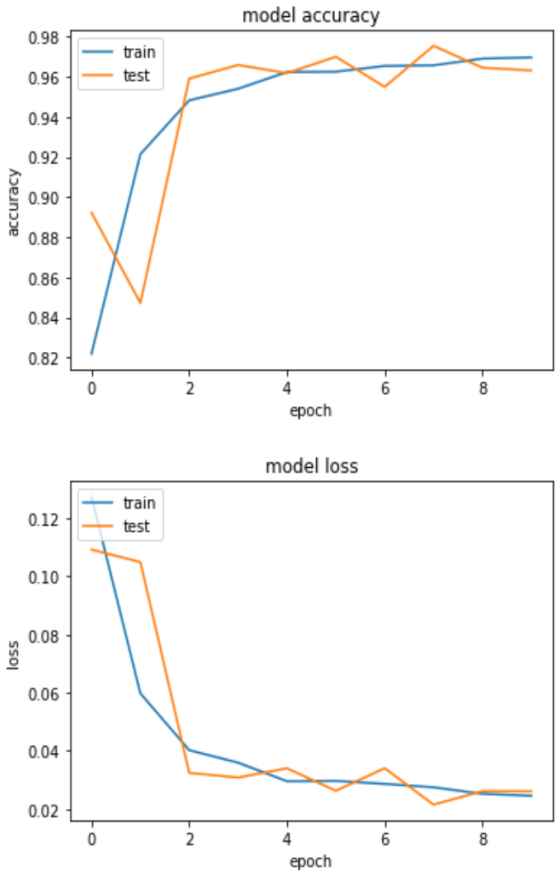

Fig. 8. Accuracy and loss training graph

D. Testing Phase

After the training phase has demonstrated a good learning process, the test phase is validated and tested on the 733 samples. To demonstrate whether the predicted image is correct, show the true image of the random image of the test sample.

E. Simulation

The next step was to search for cars in the full test image, to create bounding boxes. This was done by initially producing a detection map using the trained model, and later projecting the featured labels of the detection points to the coordinate space of the original image, transforming each point into a 64x64 square and keeping those squares within features area bounds as shown in Figure 9. To avoid any false positives, I created

Fig. 9. Car Detection

a heat map using the overlapping squares and added some thresholds to it. Then, a bounding box was drawn for every detected heat source as shown in Figure 10. The last step is to count the actual value of bounding boxes for detected vehicles.

V. CONCLUSION

In the nutshell, this paper explains the implementation of Autonomous Driving using Deep Learning. One sector that can be found in autonomous driving is car detection. The explanation should be clear for the reader to understand more about Deep Learning. Furthermore, this paper also shows how I implement Deep Learning in the project. The diagrams and the tables should help the reader with the visuals to imagine and understand more Deep Learning.

VI. ACKNOWLEDGEMENT

I am eternally grateful to Prof. Achim Rettberg, whose inspiration, encouragement, guidance, and support from the beginning to the conclusion helped me to gain awareness and open my eyes to the importance of Deep learning in general. I'd also like to express my gratitude, respect, regard, and benefits to everybody who helped me in any manner during the task's execution. I did everything in my power to obtain a

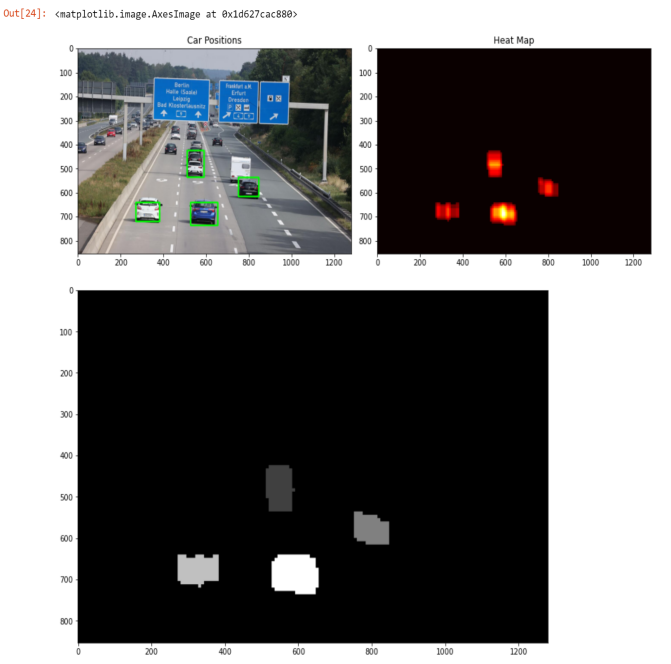

Fig. 10. Tranform the results into heat map

better understanding and share it in my paper, and I hope the reader can benefit from it, particularly in this area.

REFERENCES

[1] Ismael, S.O., 2020. Deep Learning and Image Processing for Handwritten Style Recognition: Deep Learning and Image Processing for Handwritten Style Recognition.

[2] Patro, D.S. (2020) Rundown on deep learning, DEV Community, Available at: https://dev.to/siddhantpatro/rundown-on-deep-learning-d25 (Accessed: January 10, 2023).

[3] O'Shea, K. and Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

[4] Difference between a neural network and a deep learning system (2022) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/difference-between-a-neural-network-and-a-deep-learning-system/ (Accessed: January 10, 2023).

[5] Ray Barua, S., 2019. A Strategic Perspective on the Commercialization of Artificial Intelligence: A socio-technical analysis (Doctoral dissertation, Massachusetts Institute of Technology)

[6] Yingge, Huo, Ali, Imran, Lee, Kang-Yoon. (2020). Deep Neural Networks on Chip - A Survey. 589-592. 10.1109/BigComp48618.2020.00016.

[7] Arróspide, J., Salgado, L. and Nieto, M., 2012. Video analysis-based vehicle detection and tracking using an MCMC sampling framework. EURASIP Journal on Advances in Signal Processing, 2012(1), pp.1-20.