Video-Trained Autonomous Car

Taha Yunus Hamamcıoğlu - 64190006
Fatma Büşra Yaman - 64190002

Introduction

The Video-Trained Autonomous Car Project is based on the idea of developing a final product carried out by implementing the cutting-edge artificial intelligence technology to hardware. In this sense, an autonomous car which is trained by self captured videos is developed. The power of deep neural networks is aimed to be proved by getting the car working only by feeding the neural networks with raw images.

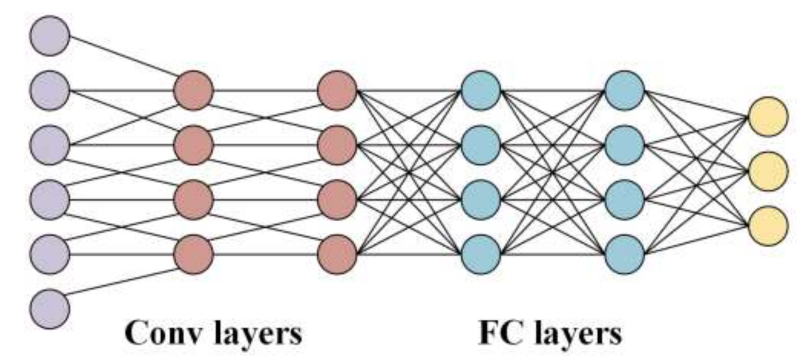
Overview

The focus of this project is applying the latest software technologies to hardware by building an autonomous car that is trained using video data. The autonomous car is capable of making decisions using deep learning models in tandem with the Intel Realsense camera that is capable of extracting the depth information as well as colour information. The car is trained and tested on modular roads that can be rearranged to create diversified paths. For the training, unique models were created that separates us from the rest.

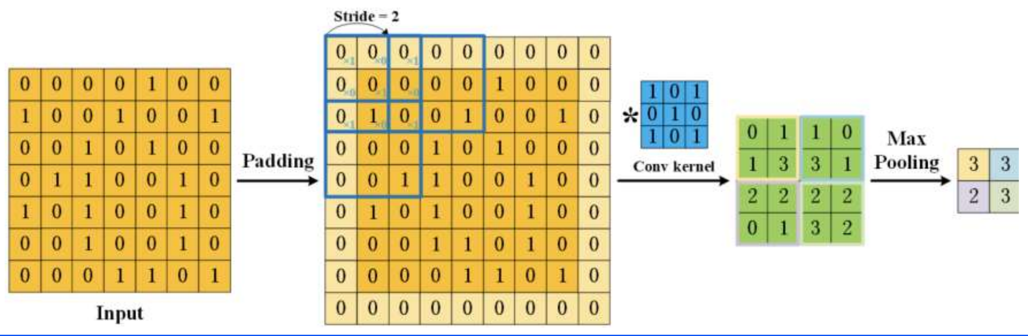
Mathematical Background

The CNN is a kind of feedforward neural network that is able to extract features from data with convolution structures. Convolution is a pivotal step for feature extraction. When setting a convolution kernel with a certain size, we will lose information on the border. Hence, padding is introduced. Besides, to control the density of convolving, the stride is deployed. After convolution, feature maps consist of a large number of features that is prone to causing overfitting problem. As a result, pooling is proposed to obviate redundancy, including max pooling and average pooling. [1]

A neural network structure: [1]



The procedure of a CNN: [1]



The convolution for one pixel in the next layer is calculated according to the formula. [2]

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k]$$

(f:input image, h:kernel, the indexes of rows and columns of the result matrix are marked with m and n respectively.)

Forward propagation consists of two steps:

$$\mathbf{Z}^{[l]} = \mathbf{W}^{[l]} \cdot \mathbf{A}^{[l-1]} + \mathbf{b}^{[l]} \qquad \mathbf{A}^{[l]} = g^{[l]}(\mathbf{Z}^{[l]})$$

(Z:result of previous layer with W tensor, b:bias, g:activation function)

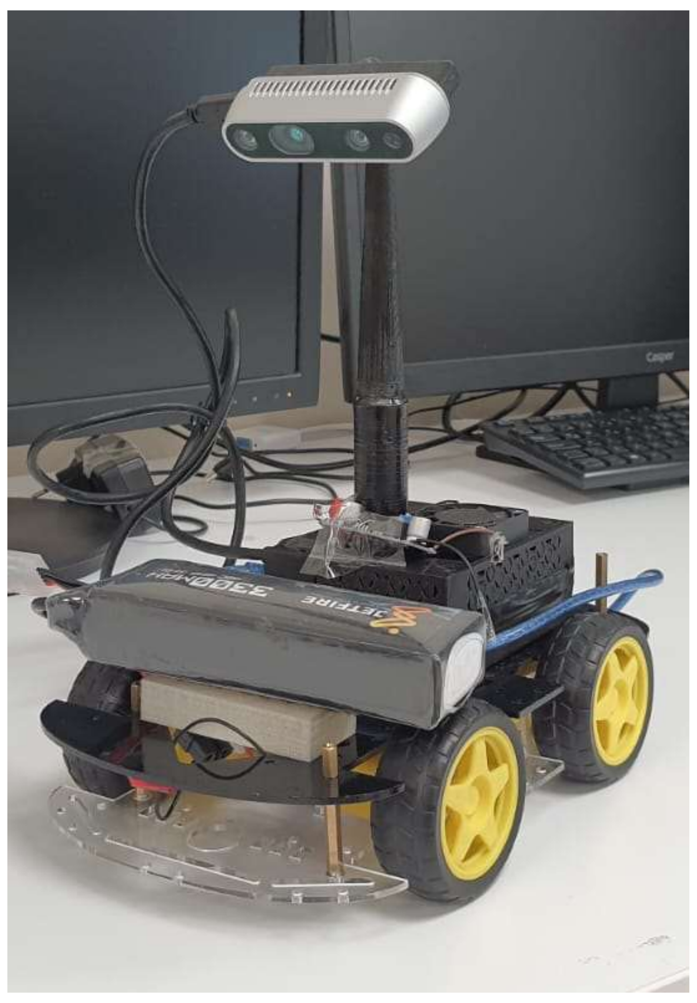
Back propagation:

$$\mathbf{dZ}^{[l]} = \mathbf{dA}^{[l]} * g'(\mathbf{Z}^{[l]})$$

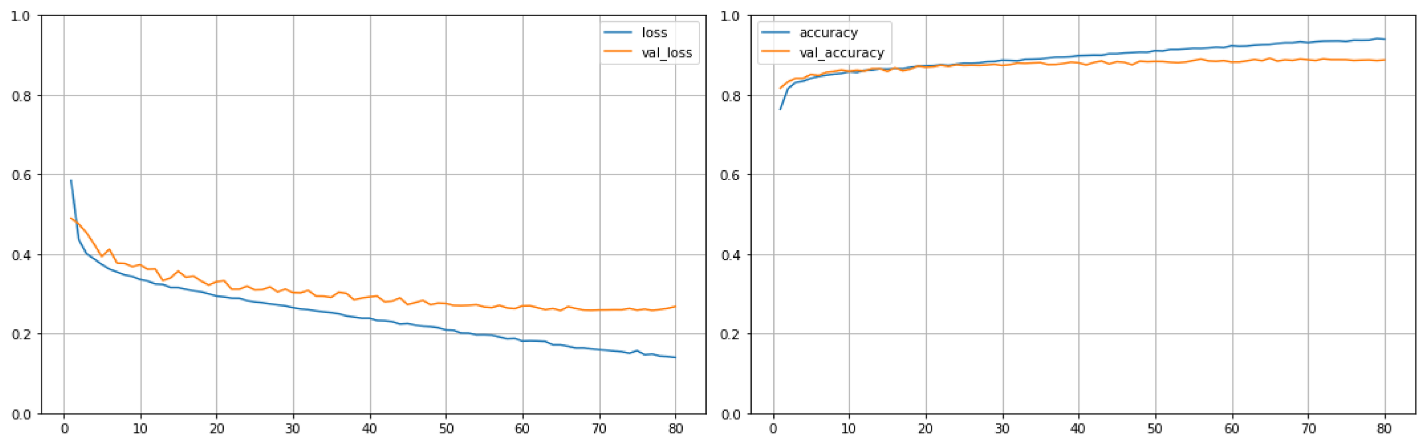
$$\mathbf{dA} += \sum_{m=0}^{n_h} \sum_{n=0}^{n_w} \mathbf{W} \cdot \mathbf{dZ}[m,n]$$

(dW[l] and db[l] are derivatives associated with parameters of current layer, dA[l -1] will be passed to the previous layer, dA[l] is the input, W:filter, dZ[m,n] is a scalar that belongs to a partial derivative obtained from the previous layer.)

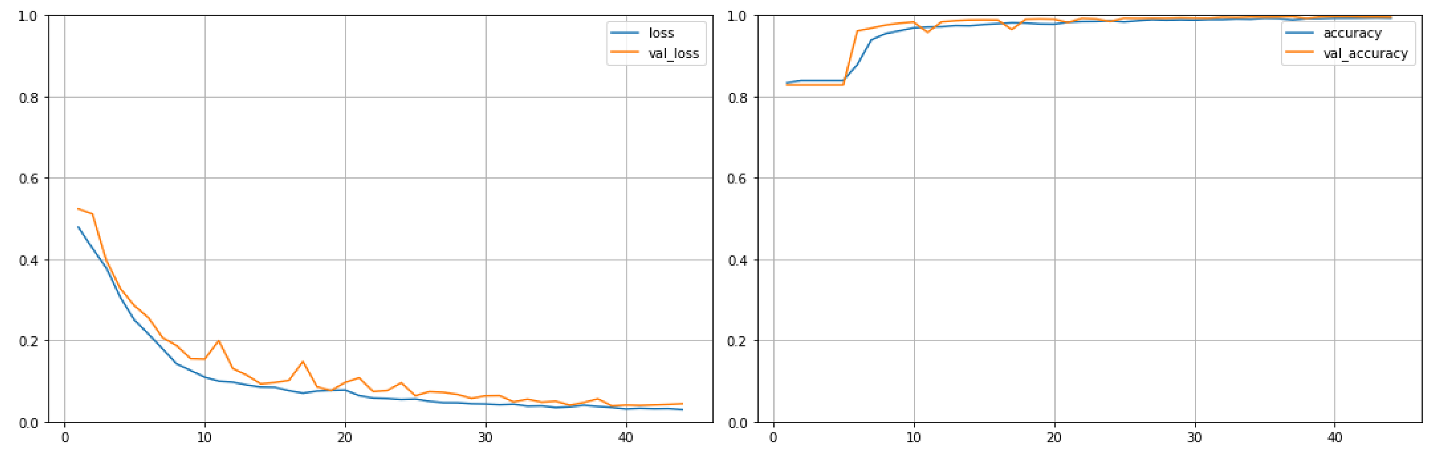
Results



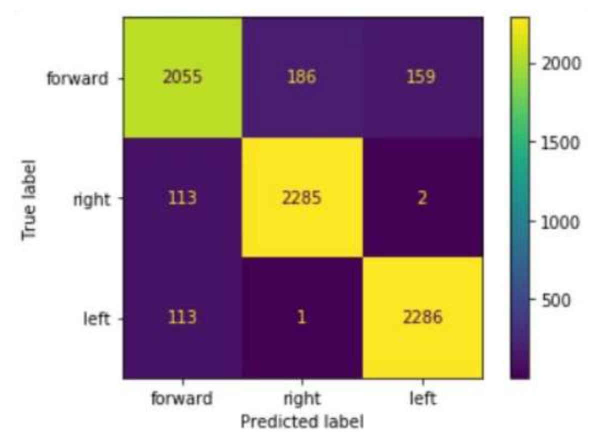
Training-Validation Losses and Accuracies for RGB Images:



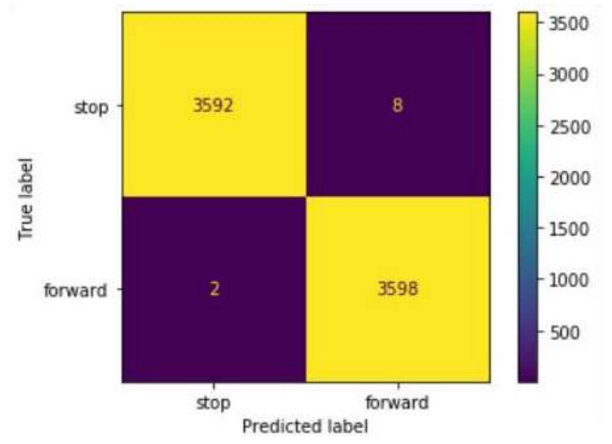
The Training-Validation Losses and Accuracies for Depth Images:



Confusion Matrix – RGB:



Confusion Matrix – Depth:



RGB Test Accuracy:
0.92

Average Accuracy: 0.96

Depth Test Accuracy:
1.00

Final Training Dataset:

Frame Label	Count (RGB - Depth Pair)
Forward (155, 155)	15.775
Right (-100, 160)	14.317
Left (160, -100)	14.317
Stop (0, 0)	9.302
TOTAL:	53.711

Final Testing Datasets (RGB – Depth separately):

Frame Label	Count (RGB)
Forward (155, 155)	2400
Right (-100, 160)	2400
Left (160, -100)	2400
TOTAL:	7200

Frame Label	Count (Depth)
Others	3600
Stop (0, 0)	3600
TOTAL:	7200

- Lane Keeping Rate: **100%**
- Obstacle Avoidance Rate: **98%**
- Car is working in real-time **15 FPS**

Conclusions

The project aimed to develop an autonomous car by using advanced artificial intelligence technologies and training it with self-collected data. The project consisted of five work packages, consisting of literature review, designing and assembling the car, creating the dataset, developing deep learning models, and optimizing their speed and accuracy. The first four packages were completed successfully. Reviewing 15 articles were enough for the first package. For the hardware section, we created a car that can work with a remote controller and without it for the inference that can loop around the paths consecutively despite some hiccups along the way. Data collection part was finished swiftly by collecting around 50 thousand frames after creating the puzzle-like structure of the road and preparing the proper hardware and software for it. Our dataset is more diverse than most other research mentioned here owing it to our modular roads. Deep learning part package was achieved by obtaining the expected accuracy, by having the most of the items in the diagonal of the confusion matrix, by working in real-time, by having an obstacle avoidance rate of more than 98%, and lane keeping rate of 100%. Optimization part was accomplished completely since the FPS is not 25 as intended even with using TFLite, but the car can drive around without any problems. Although so many problems were encountered, project team dealt with all of them and develop the desired final product. The project is successfully completed with great effort of the team members.

References

[1] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems.

[2] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.