International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Parking for Autonomous Vehicle Using Deep Learning Approach

Ebubekir Ceylan[1], Fuad Aliew[2]

[1]Department of Electronic Engineering, Gebze Technical University, Gebze, Kocaeli Province 41420, Turkey
[2]Yeditepe University, 34755 Ataşehir - İstanbul, Turkey

Abstract
In this study, a model is proposed to allow the vehicle to park safely itself using deep learning approach for autonomous parking problem. The proposed method is included three main steps. In the first stage, the vehicle which a remotely controllable is designed to collect data. The goal of second part is prepared a track and label the collected data in order to collect real data with the designed vehicle. In an addition this collected data, is also added CNR parking dataset which is also based on real camera pictures. It has been used 75% of these data as training and 25% of them as testing. Lastly, a deep learning model has been developed based on CNN and transfer learning according to these data. This model classifies whether parking slot is empty or not. if classification is called free, then the vehicle starts to be parking maneuver, the vehicle doesn't do parking maneuver while it is called busy. As a result, accuracy of the model was calculated about 0.96 which is quite good for park slot occupy detection. It was also performed successfully 7,6 out of every 10-parking maneuver.

Keywords: Autonomous Vehicles, Deep Learning, Autonomous Parking, Self-Driving Car, Convolutional Neural Network

1. Introduction

From past to present, many robots that human life easier have emerged. Some of these can give many examples such as food robots, industrial robots that work more efficiently and faster than humans, drones that make cargo work faster, underwater vehicle that go to deeper points, unmanned aerial vehicles that chase terrorists. New technologic settings are added to improve skills of these robots over years. One of these topics is said sub-branches of artificial intelligence. Especially in recent years, deep learning and computer vision studies have gained momentum due to increase in processor power and decrease in cost. One of the fields of study where most development related to these is also autonomous vehicles. With the development of both hardware and artificial intelligence algorithms, has researched on multiple problems have been made and solutions have been produced in this field. But there are still many different problems to be solved in autonomous vehicle technology. Object recognition, classification, decision making, parking systems can be given as examples for some of these. In this study, the problem of autonomous parking will be discussed.

The problem of parking is mostly caused by the driver's inexperience or carelessness. This situation damages financially both sides and also breaks self-confidence of the driver. This is one of the topics where autonomous systems will outperform than humans. Thanks to autonomous parking technology,

vehicle damage costs can be reduced, the fuel can be saved, and time save can be made by shortening the parking time.

Although this problem is tried to be solved with deep learning or classical algorithms, it still brings security problems. Tools trained only with deep learning or machine learning can be ineffective against unexpected situations or make wrong decisions and cause bad results. On the other hand, systems used only by traditional algorithms bring a lot of computational power and cost as they are based on multi-phase control.

2. Related Work

The first studies on autonomous vehicles were pioneered by the General Motors company which organized the World Fair in 1939 [1]. However, it can be said that interesting studies started in 1977 in Mechanical Engineering Laboratory in Tsukuba, Japan. A vehicle that can reach 20km/h and follows the white line has been developed in this laboratory. In the "European Prometheus Project", which was the longest robot car project up to that time with approximately 1 billion dollars spent in 1995, the vehicle successfully drove up to 96km/h on empty streets. Dickman's famous S-class car autonomously traveled 158 km per hour from Munich to Denmark without human touch, making a total of 1678 km [2]. In 1989, Pomerleau commissioned the first road drive of ALVINN that a land vehicle funded by DARPA in the United States. The difference of this vehicle from previous studies is that by making use of neural networks and preprocessing data from sensors such as cameras and lasers placed on the vehicle, the vehicle has been accelerated to 30 km/h [3]. As we get closer to the present, the designs of autonomous vehicles in the recent past have diversified with the development of technology, reducing the cost and offering new options. As an example, in 2008, Çayıroğlu and Şimşir transferred the recordings from the wireless camera to the computer by adding the ARX-34S receiver and ATX-34S transmitter to the robot they designed with PIC16F877A processor [4]. Fong and Yusoff tried to take images using Wi-Fi to control the vehicle in real time [5]. In another study, Kuo enabled the mobile vehicle to be controlled via the computer with XBee module and used it with a wireless camera for image transfer He made this robot with Arduino and added the wireless controlled cleaning tool feature to this robot [6]. Espes and colleagues designed a mobile robot working over the internet in 2014. In this robot, they developed a cost-effective mobile tool with web control for home display using Raspberry Pi to connect to the internet and Arduino mega for steering wheel control and robot arm [7]. Ali also designed an autonomous vehicle when made position control with fuzzy logic in 2017 and tried to control the vehicle remotely via Bluetooth using Arduino mega in this vehicle [8]. The methods to be used are as important as the design of the vehicle and the technologies used. Although more classical methods were used in the past, today it has begun to leave it to deep learning algorithms instead of traditional methods. To give an example of one of the classical methods in 2019, Alpkiray used PRM and ABC algorithm to create the return route [10]. In the same year, as an example of the deep learning method, Bingöl and his friends designed vehicle using the NVIDIA Jetson TX2 card for a more stable drive and they first collected data by driving the vehicle themselves and developed a deep learning model that learned from this data which allowing the vehicle to reach its goal [9].

There are many problems that autonomous vehicles must overcome on their own without expert driving. One of them is the parking problem. Notomista and Botsch, who found a solution to this problem, approached this issue with machine learning approach. In the vehicle they designed, they divided the maneuvers into 3 separate parts, and they were successful in the self-parking the vehicle with general basic

function model they designed by passing these parts through two different classifiers [11]. Zhang and his friends have offered another solution to the parking problem. With the Reinforcement learning they did, they aimed for vehicle to learn from each attempt through punishment and reward. With this learning model, only the entrance image was given to the vehicle, and it was performed the self-parking process with help of end-to-end learning model [12]. Likewise, Jelena and her colleagues designed a vehicle for embedded automotive platform with end-to-end learning method, but this time using deep neural networks and successfully parked the vehicle [13]. Moon and his colleagues also created a double neural network model by cloning the neural network model and combining this clone with itself. The developed an automatic parking controller that successfully parks with this model [14]. However, the fact that the systems become autonomous and have their own decision-making leads to some quite a lot of CPU power, unexpected crashes, and other problems. In this paper, we propose an autonomous parking model with low parking computational power, easy, inexpensive, and not complex.

In this research, first of all, it will be shown that the vehicle with which features is designed and how data is collected under which conditions. Then, it will be examined how the collected data are labeled and how a successful model is trained from the labeled data. After this process, the performance of this model will be compared with other successful models. Finally, it will be explained how the trained model parks.

3. Methodology

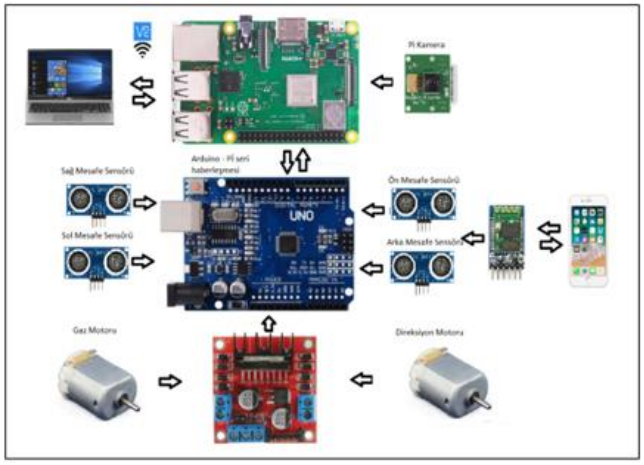

Fig. 1. General wiring diagram of the RC vehicle

3.1. Vehicle Design

The choice of sensors and hardware is as important as the algorithms used in autonomous vehicles. In this study, sensors and other components were selected under these criteria since it was aimed to design an easy to use and cheap vehicle with low computational power. In this work, Raspberry Pi 3 was used as a controller for training deep learning model and to provide the opportunity to manage the system from a single center. Arduino uno was chosen for motor control and also used l298n card which is very popular as motor driver. Ultrasonic sensor has been added in this vehicle to be aware of obstacles during parking maneuver. Pi camera was preferred for data collection of this tool because it is compatible with Rpi. In order to collect data by expert driver, a bluetooth sensor was placed in the vehicle so that the vehicle could be controlled remotely. In addition to these, a 4-wheel drive chassis was selected, which has

the necessary maneuvering for the vehicle to park. In this way, a vehicle was designed with high maneuverability and was collected data on parking maneuvers by controlling the vehicle remotely.
Since the parking area is on the right side of the track as seen in figure 2, the camera is placed at a 45-degree angle to the vehicle. Due to the engine will be operated at low power, a single power source was deemed sufficient, and the vehicle was feed by a 5000 mAh powerbank.

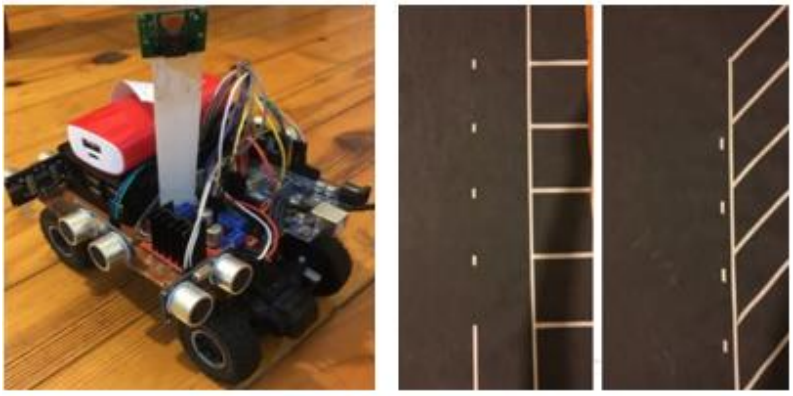

Fig. 2. Front view of RC vehicle and image of prepared track

3.2. Datasets

One of the most important issues while developing the model in real-time object recognition and classification applications is that the dataset should be realistic for desired application. Models which is used by a ready-made dataset sometimes cannot show the same performance in live applications. In order to prevent this, a dataset was combined by collecting both the ready dataset and the expert user dataset in this study. It has been used 75% of this dataset for training and 25% of them for testing. First dataset is CNRPark dataset that provides a great opportunity for research, containing approximately 150,000 labeled images of free and busy parking spaces. There is also dataset which contains about 12,000 labeled is added to the bigger dataset [15]. Obstacles such as electricity poles, trees, other cars and partial occlusion pattern due to different conditions are included in this dataset. It was used to train model with over 5000 labeled data sets in total by taking 4000 images from CNRPark dataset. Since this study is a real-time study, it is necessary to collect data in the parking area of the vehicle. For this reason, the track was established, and the images were recorded while the vehicle was parking by the expert user. The most appropriate 866 images from the collected images were taken as training data. More than 2000 data labels were made from many of these images with more than one labeling process.
The images collected from vehicle controlled by expert driver by phone are shown in figure 3. As important as collecting data set for successful model training, it is equally important to label the dataset correctly and properly. An image can also contain data to be labeled more than once. Therefore, the more diversely labeled dataset are obtained in different environments, the higher the chance of success of the model. In this research, processing of dataset labelling is done by accessing the python application from the command line.

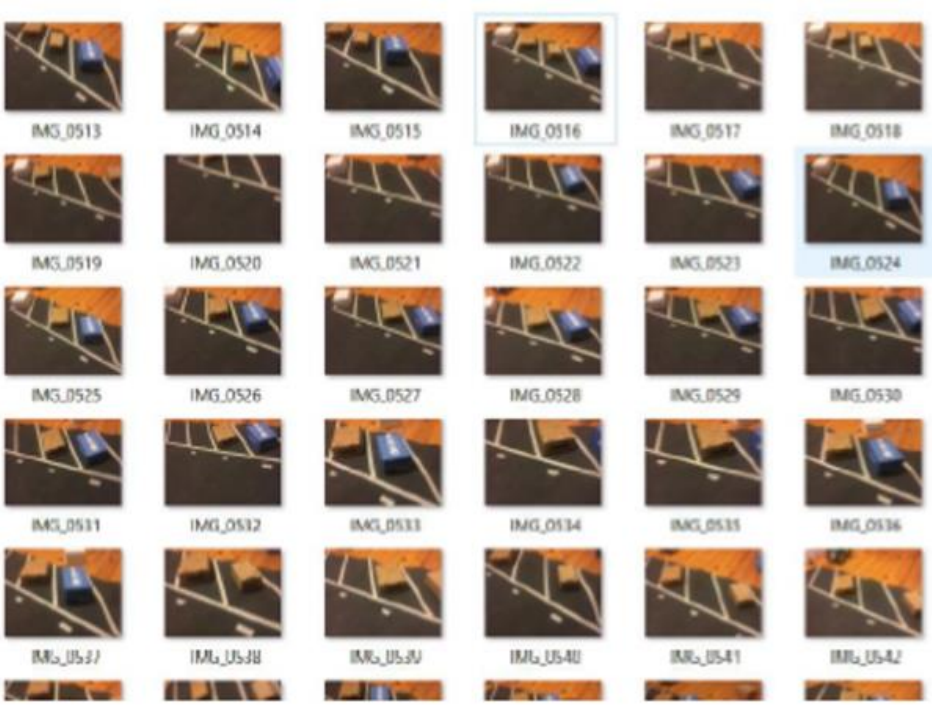

Fig. 3. Some of images collected by expert user remotely

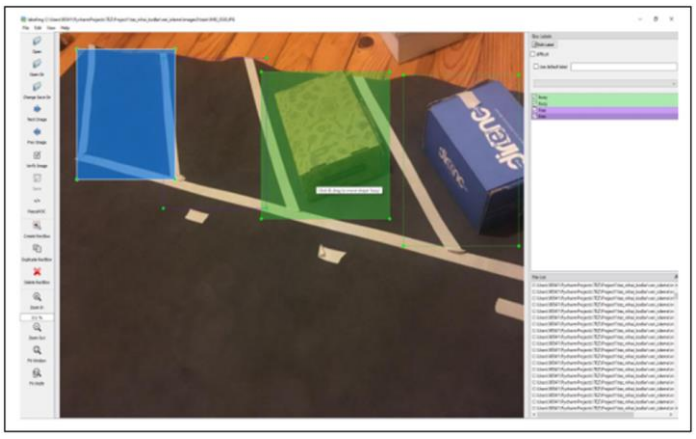

Fig. 4. A example image of labeled parking slot data

After accessing application, an interface like figure 4 appears. The action to be done here is to first show the folder with the raw data in the program. The next step is to show the folder where tagged data will be. The important point is that the labeled data must be in the same folder as the raw data. Otherwise labeling of data will not matter. in figure 4, the objects that exist in a single image are labeled in 4 different ways as empty or full according to their status in the parking lot. Similarly, 866 collected datasets were labeled in this way and more than 2000 labeled data were created.

3.3. Proposed Architecture

Firstly, we need to detect whether the parking slot is free or not, to make self-driving parking. We present AciNet Model, that reduces convolutional layers, also we analyze its presentment according to CNN. Convolutional neural networks are part of deep learning algorithm created on the basis of the working principle of sight, one of the most advanced senses in the world. The architecture of convolutional neural

networks, which is a sub-field of multi-layered neural networks, has extra layers unlike the others. This network architecture was introduced by Yann LeCun in 1998 [16]. In this first study, very successful results were obtained on handwritten numbers. This success laid the foundation for the widespread use of deep learning models today. Even though this first architecture had 4 layers, 2 convolution and 2 communing layers, more successful architectures have been developed over the years thanks to the developing processing power. The first of these was 1.2 million training data with 1000 classes as a data set in the ImageNet (ILSVRC) competition in 2012. The best model in this competition was the 8-layer AlexNet, which has a deeper architecture than the Lenet model [17]. In 2014, the 19-layer VGGNet architecture managed to reduce the error rate up to 6.8%. Since it is known that human performance is at 5%, a success close to human has been achieved. [18]. VGGNet can also be used with 11, 13, 16 layers.
The models described in the sections above and which perform quite well in deep learning applications are described. However, these models need more powerful processing power due to their many layers and high computational power. In this case, it brings a very high cost. For this reason, a deep learning model called AciNet, which is designed with less layers than other models that can work with lower processing power, has been proposed. In this chapter, the presented model performance will be examined and checked.
It is showed the structure of the proposed network architecture in Figure 5. At first glance, it seems to be an 8-layered architecture. The reason for this is not to consume unnecessary processing power since the classification process is only 2-class in the application to be applied.

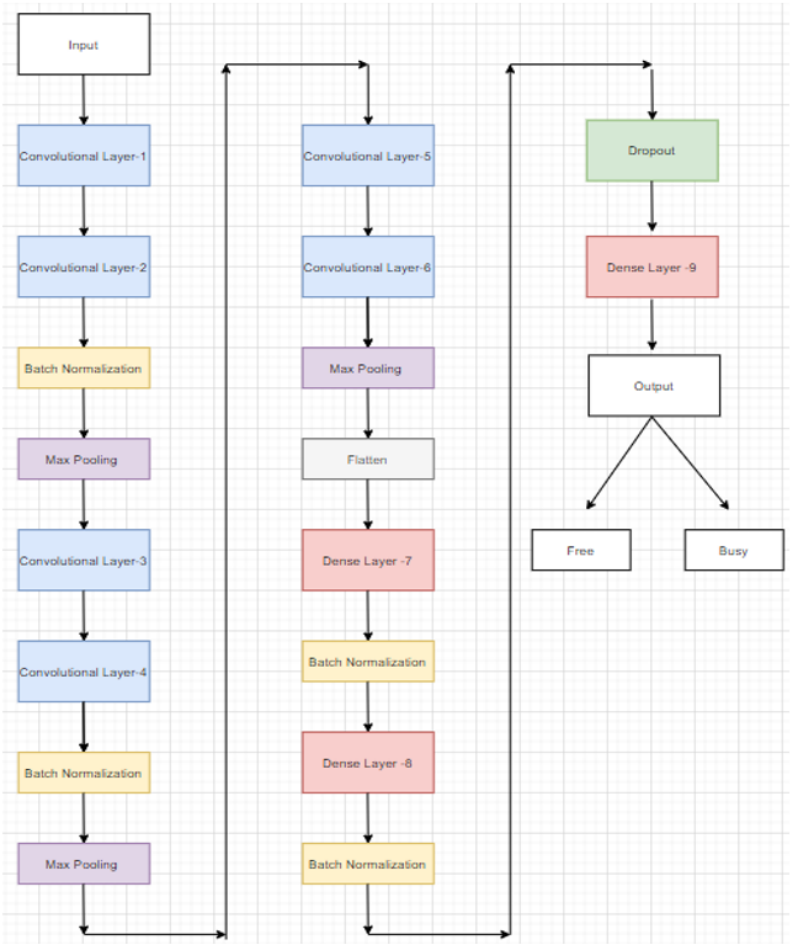

Fig. 5. The network of archtitecture of proposed model

In the proposed network architecture, the inputs are reduced to 32x32 sizes so that they do not take up much memory. When the data enters the training, they are trained in packages with 32 groups. The convolution filter size is chosen as 3x3, and since a 2-class output will be produced in the output layer of 32 to 64 in the convolutional layer and 1024 in the full link layer, the layer values are defined by choosing 2 at the end. In this model structure, unlike other models, the dilution process, which is widely used in deep learning processes, is not used. The reason for this is that there are not enough data sets to require excessive memorization of the model. By doing the dilution process only in the last part, the learning of the model is prevented. Within the framework of this application, it has been experimentally determined that it is better to arrange after each convolution layer rather than dilute it. Therefore, it is aimed to show the best success by performing packet normalization after the first two convolution layers and before the last two convolution layers. The cross-entropy function was used, which is frequently preferred in deep learning applications as a cost function in the AciNet model. In the optimization function selection, SGD optimization is the most suitable for this job since the data will enter the model in 32 mini-packets. To able to reach the local minimum for the error, values which is like 0.001, 0.0001, 0.00001 are taken as the learning coefficient. In this study, 0.0001 was chosen for the model with the best learning. The training period of the model also consists of 130 epoch processes. One of the most important differences of the AciNet model from models such as VGG-16 and AlexNet is that the number of layers and the total used parameters are lower. In this way, less costly operations can be performed in real-time applications by keeping the processing power low.

3.4. Additional methods: Transfer Learning

One of the problems encountered in applications such as real-time object recognition and classification is that even though the success rate of the models is very high, the trained data is not enough to be successful in real life. One of the things to do in such cases is to collect more real-life data or to transfer learning with a small amount of manually collected real-time data [19]. Learning transfer is the study of storing the knowledge gained while solving a problem in machine learning and solving another problem with that stored information and new data specific to the problem [20]. Convolutional neural networks give very successful results in the field of image processing. One of the subjects where these neural networks work quite well is the studies of recognizing and classifying objects. Multilayer neural network models with many different architectures have been developed for object recognition and classification. Some of the most used architectures in studies on this subject are R-CNN, faster R-CNN, SSDlite MobileNet and different advanced versions. Among these models, the model with the highest success rate is Faster R-CNN [21]. The choice of these architectures may differ according to applications where speed is important, or success is important. For example, Faster R-CNN architecture can be used in real-time systems applications, applications where speed and success are at the forefront. However, in applications where systems with lower processing power such as Raspberry Pi are used, SSDlite MobileNet architecture is preferred because speed is not important, and success is important [22]. Although the success of the model was very good in this study, the vehicle could not show the same performance in the field due to the small amount of data. In order to overcome this problem, field studies were continued with transfer learning. In these conditions, the SSDlite MobileNet architecture was chosen as the most suitable model, since the scarcity of data, low processor power and performance are at the forefront.

4. Experiments and Results

4.1. Performans Evalutions

In this section, firstly, the performance of the model will be observed over 4000 CNRPark images, which is a ready dataset. Then, the performance of the model on 4866 data will be examined.

Table 1 AciNet total performance results over all data.

Dataset	Epocs	Train Ac.	Train Loss	Test Ac.	Test Loss
CNRPark	130	0.98	0.05	0.94	0.20
All	130	0.96	0.12	0.96	0.09

When the results in the table are examined, performance of the ready-made AciNet model on the whole dataset has been 2% better than performance in the ready-made dataset. At the same time, there was a reduction in loss of 11%. This difference in the proposed model is due to the addition of the manually collected data set to the ready data, adding diversity to it.

In second part, a comparison of the deep learning models described in the previous sections and the model proposed in this study will be made. In order to make this comparison, all models were trained with the same parameters and all datasets.

Deep neural networks with 5 different architectures were trained using the same numerical values, on the same data set with unique layers, at the same epoch number, with the same loss functions and the same optimization functions. Performances are seen as in Table 3. When the tables and graphs were examined carefully, VGG-13 showed the highest success. Considering the table in terms of layers, the proposed AciNet model showed more success with approximately 10 times less parameters, although it has the same layer as AlexNet. Likewise, considering the number of parameters, AciNet and VGG-11 architectures have at least two parameters. Again, although AciNet has approximately 5 times fewer parameters than the VGG-11 architecture, it performed approximately 2% better and the error was 4% less.

Table 2 Loss and accuracy results of all models

Dataset	Epocs	Train Ac.	Train Loss	Test Ac.	Test Loss
AciNet	130	0,96	0.12	0.96	0.09
VGG-11	130	0.97	0.05	0.94	0.13
AlexNet	130	0.97	0.06	0.95	0.11
VGG-13	130	0.98	0.04	0.98	0.06
VGG-16	130	0.98	0.03	0.97	0.08

Table 3 Comparison of the performance of all models

Dataset	Number of Layer	Number of Parameter	Acc.	Loss
AciNet	8	2.249.699	0.962	0.093
VGG-11	11	10.582.675	0.943	0.134
AlexNet	8	25.657.955	0.950	0.113
VGG-13	13	32.386.403	0.984	0.060
VGG-16	16	37.696.099	0.984	0.086

As seen in at Table 3, when looking at VGG-16, which is the most successful model with AciNet, the VGG-16 architecture is 2% more successful, while the number of parameters is approximately 15 times higher. However, when we look closely at Table 2, it has been revealed that AciNet has produced a much more stable and reliable result on the whole data set on the test data.

4.2. Parking Space Occupy Detection and Parking Manuevers

In this section, the parking maneuver will be performed according to the vehicle suitability determination. To do this, the vehicle will detect whether the parking space is empty with the live image it receives from the camera, and data will be sent to the vehicle's controller maneuver accordingly. According to the data received by the Controller, the vehicle will either perform a parking maneuver or not by until the parking lot is empty.

In figure 6, it is seen that the parking space occupation determination is made instantly with transfer learning. In the area where the parking lot is occupied, it prints out as full in a yellow frame, while it prints out empty in a green frame when the parking lot is not occupied.

Fig. 6. Instant parking slot spot detection with transfer learning

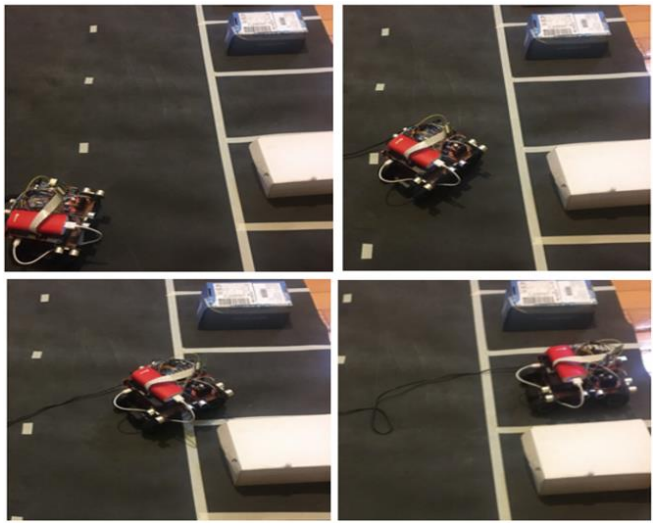

Fig. 7. Vertical parking maneuver

In Figure 7, it is seen that the vehicle made a vertical parking maneuver. In 50 different attempts to park the vehicle vertically, the vehicle parked maneuver with a success rate of 76%. The data showed that after the successful parking space classification performance of the vehicle, the vertical parking maneuver performance was also performed successfully.

5. Conclusions

This research was developed for the purpose of automatic self-parking of the vehicle. This system has two main stages. At first stage, efficient solution-AciNet, for visual detection of a park slot occupy status, was proposed that uses reduced Convolutional Neural Networks. AciNet has been presented as a successful model to understand whether the parking space images obtained by the camera are occupied or empty. As a result, AciNet has demonstrated a performance with %96.2 accuracy compared to the most used architectures in deep learning applications. At second stage, vehicle is doing self-parking by helping of controller which uses an output coming from AciNet model after classification. This has also shown a well performance with %76 over many experiences. These experiments approve that our presented method outperforms very reputable architectures such as AlexNet and VGG and that lower layer's models give good results in applications with low classification.

As a result, vehicle design, data collection, data processing, training of developed model, the success of the model compared to others, and the parking maneuver of the trained model were examined in this research. The results indicated that the performance of AciNet model developed for parking space occupancy detection is much more successful than other known very good models.

Our proposed method shows that potentially provide a cheap and reliable solution according to the success and consumption of less processor power and a smaller number of parameters. However, most importantly, the parking maneuver process is not based on the result of a single trained model, but by first checking the output of the model and providing the parking process according to this output, a more controlled and safer automatic parking process is provided. the proposed approach can be used further for better without additional information and by reducing unnecessary costs.

References

1. Ozguner U., Acatman T. ve Redmil K., (2011), "Autonomous Ground Vehicles", USA, Artech House, 3-5.
2. Web 1, (2020), http://www.idsia.ch/~juergen/robotcars.html, (Erişim Tarihi: 18.06.2020).
3. Pomerleau D.A., (1989), "ALVINN: An Autonomous Land Vehicle in a Neural Network", In Advances in Neural Information Processing Systems, Morgan Kaufmann, San Mateo.
4. Çayıroğlu İ., Şimşir M., (2008), "PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü", Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 24(1-2), 1-16.
5. Fong P. E., Yusoff M. A., (2011), "Real-Time Control of Wi-Fi Surveillance Robot", Proceeding of the International Conference on Advanced Science Engineering and Information Technology, Malasia.
6. Kuo G., Cheng C., Wu C., (2014), "Design and Implementation of a Remote Monitoring Cleaning Robot", International Automatic Control Conference, Taiwan.

7. Espes D., Autret Y., Vareille J., Le Parc P., (2014), "Designing a Low-Cost Web-Controlled Mobile Robot for Home Monitoring", The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Italy.
8. Çetinkaya A., (2017), "Otonom Bir robotun bulanık kontrolör yaklaşımı ile konum kontrolü", Karatay Üniversitesi, YL Tezi.
9. Bingöl S., Kaymak Ç., Uçar A., (2019), "Derin Öğrenme Kullanarak Otonom Araçların İnsan Sürüşünden Öğrenmesi", Fırat Üniversitesi, Fırat Üniversitesi Müh. Bil. Dergisi 31(1), 177-185.
10. Alpkıray N., (2019), "Otonom Keşif Amaçlı Robot Sistemleri İçin Geri dönüş Rotası Hesaplama Algoritması Geliştirilmesi", Cumhuriyet Üniversitesi, YL Tezi.
11. Notomista G., Ve Botsch M., (2016), "A Machıne Learnıng Approach For The Segmentation Of Driving Maneuvers And Its Application In Autonomous Parking", JAISCR, 2017, Vol. 7, No. 4, pp. 243 – 255.
12. Zhang P., Xiong L., Yu Z., Fang P., Yan S., Yao S., Zhou Y., (2019), "Reinforcement Learning-Based End-to-End Parking for Automatic Parking System", sensors 2019, 19, 3996.
13. Kocic J., Jovicic N., Drndarevic V., (2019), "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms", sensors 2019, 2064.
14. Moon J., Bae I., Kim S., (2019), "Automatic Parking Controller with a Twin Artificial Neural Network Architecture", Hindawi 2019, Article ID 4801985.
15. Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Vairo, C. Car parking occupancy detection using smart camera networks and Deep Learning. In Proceedings of the IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 1212–1217.
16. LeCun Y, Bottou L, Bengio Y, (1998), P., "Gradient-based learning applied to document recognition". In Proceedings of the IEEE, 86(11): 2278-2323.
17. Krizhevsky A., Sutskever I., Hinon G., (2012), "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
18. Simonyan K., Zisserman A., (2015), "Very deep convolutional Networks for large-scale image detection", ICLR 2015.
19. Oturak M, (2018), "Öğrenme aktarımı ile sınıflandırma", Yüksek lisans Tezi, Hacettepe Üniversitesi.
20. Goodfellow L., Bengio Y., Courville A., (2016), "Deep learning book", MIT Press, 536.
21. Daş R., Polat B., Tuna G., (2019), "Derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi", Fırat Üniversitesi Müh. Bil. Dergisi 31(2), 571-581.
22. Web 7, (2020), https://github.com/tensorflow/models/blob/master/research/object_detection/ g3doc/tf1_detection_zoo.md/