Car Monitoring System in Apartment Garages by Small Autonomous Car using Deep Learning

Leonardo León-Vera[1,2] and Felipe Moreno-Vera[1,2]
{lleonv, felipe.moreno.v}@uni.pe

[1] Universidad Nacional de Ingeniera, Lima, Per
http://www.uni.edu.pe
[2] Information Technology and Communications Center, Lima, Per
http://www.ctic.uni.edu.pe

Abstract. Currently in Peru, people prefer to live in apartment instead of houses but in some cases there are troubles with belongings between tenants who leave their stuffs in parking lots. For that, the use of an intelligent mobile mini-robot is proposed to implement a monitoring system of objects, such as cars in an underground garage inside a building using deep learning models in order to solve problems of theft of belongings. In addition, the small robot presents an indoor location system through the use of beacons that allow us to identify the position of the parking lot corresponding to each tenant of the building during the route of the robot.

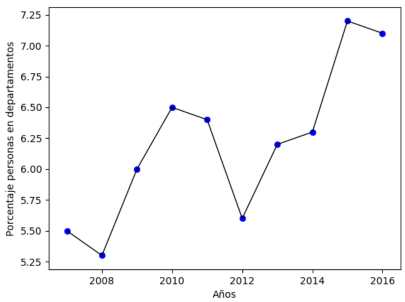
Keywords: Object Detection · Localization · Self-Driving · Low energy · Bluetooth.

1 Introduction

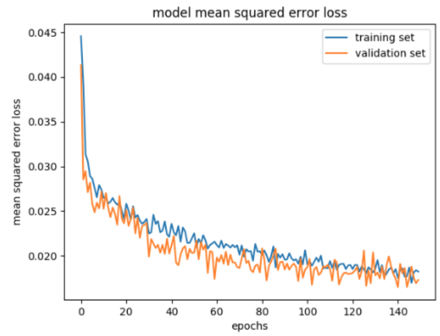
Currently in Peru, families prefer to live in apartments (see Fig. 1(a)), since among the advantages obtained are a more welcoming space and less effort in cleaning unnecessary rooms. In the same way, the size of the apartments is reduced more and more, therefore, families opt for the option of keeping their belongings in their parking lots next to their cars or instead of thems, as long as they are in an underground garage, but exist a problem when some objects disappear.

A mini robot is proposed to constantly supervise cars of the people in their respective parking lots during a schedule that does not bother the tenants of the building. Through autonomous driving techniques and object detection, a computational vision system based on deep learning algorithms is proposed in order to achieve the navigation of the robot and identify mainly the cars in question. Also, The location system construct with Beacons that determine the mini robot relative position in real time.

The navigation of the mini car in [13] is based on images using deep learning and done for more complicated tracks than the garage of an apartment and using different sensor for better acurracy in indoor scene.



(a) Growth of people living in apartments. Source: Private households of Peruvians 2007-2016. INEI.



(b) Graphics of training and validation performance of the model of [1]. Minimum quadratic error loss function vs Epochs.

Fig. 1: Graphics of demand and training

2 Structure of the System

Implementation of a mini-robot that runs through an underground garage and verify that the belongings of the tenants are in their respective parking lots. It is proposed to start the solution to the problem by detecting objects to identify the position of the car and its license plate to obtain information from the owner of the car.

2.1 Data collected from tenants of the building.

Data is collected from the tenants of the building such as the apartment number, owner of the apartment, ID of parking lot, car models, license plate of their car, objects they stored in the parking lot, etc (See Fig. 2). These will be compared with the data obtained by the mini car and shown to the caretaker so that he can draw his conclusions.

ID Depart.	Name	ID parking lot	Car model	License Plate	Car	Bicycle	...
1101	Pedro	23	Toyota	ACM-123	Si	No	...
1502	Juan	34	Gol	ABC-345	No	Si	...

Fig. 2: Data of tenants.

2.2 Navigation of the small car

To solve this task, we use webcams which are much cheaper than the Lidars and Radars which are very used for self-driving due to the important data they can contribute to the autonomous driving model, between 3D maps and calculate the distance towards objects with high precision. In contrast, cameras emulate the way in which people can see the environment giving a better classification and interpretation of textures of images in comparison to the previous ones [4]. Their future will be strongly dependent on the development of the software algorithms controlling the self-driving and how can they process the massive amount of data generated.

A Convolutional Neural Network based on the NVIDIA model[1] for regression task is used to predict the angle of rotation of the wheels of the mobile robot by only obtaining an image of the path (See Fig. 1(b)).

Data Collection Being supervised learning it was necessary to collect a video recording of the route that the mini car must follow to obtain an optimal performance. Being a flat floor, we did not get additional problems which would give problems when testing the system. However, when obtaining an environment where the walls are very similar, we look for a correct positioning of the camera and cut out the image so that it only focus on taking images that correctly identify the curves of straight roads. The input data are the images of three three front cameras separated from each other by a few centimeters to collect more data from the road and speed of the small car at each moment, and values of angles between 0 and 180 as output values for training the model.

Neural Network Architecture A deep sequence model of layers is used in the following manner, with its respective number of filters: Conv24-Conv36-Dropout-Conv48-Dropout-Conv64-Dropout-Conv64-Dropout-FC-FC-FC-FC with non-linear activation function ReLU, which bring good results in computational vision tasks in the Convolutional layers. Regularization methods such as Dropout with a probability of 0.5 are added, due to the overfitting that occurs with the NVIDIA model when iterations are increased considerably.

Neural Network Training The training of the data was done with the board NVIDIA P4000 with a partition of 20% validation set and 80% training set. The MSE loss function and the ADAM optimizer method are used without the need to manually set the speed of the learning rate.

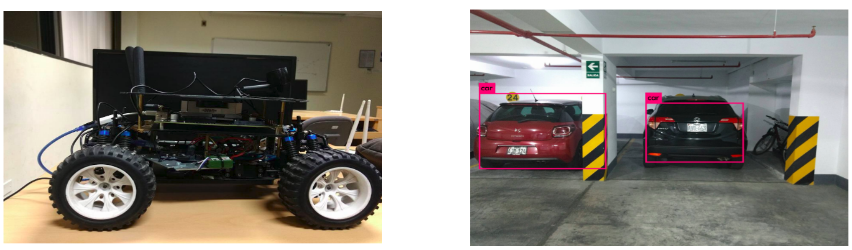
Neural Network Testing Testing of the convolutional neuronal network is performed on the Udacity simulator, obtaining a clear outstanding performance when driving autonomously as can be seen in the video [5] and later applied in underground garage.

2.3 Cars Detections

Cars detection task is in charge of a Convolutional Neuronal Network using Tiny YOLO model[3] that is the best model with the best performance beetween accuracy and inference time in detection objects in real time such as cars, bicycles and others objects (see Fig. 3(b)), model also detects void spaces if probabilty to find any object is lower than the threshold.

Dataset The dataset used to train the model is Visual Object Classes Challenge 2012 (VOC 2012). That present 20 classes, among them is cars (in a future work, we train chairs, tables, bicycles, etc). For future work, the model is going to be trained using ILSVRC[11] 2014 dataset which have better quality on images such object scale, level of image clutterness, 200 classes of objects, among others.

Neural Network Architecture Tiny YOLO model consists in a convolutional neuronal network with 9 convolutional layers of 16, 32, 64, 128, 256, 512, 1024, 512, 425 filters each one. This model is lightweight in comparison with "YOLO9000: Better, Faster, Stronge" [3] and accors.



(a) Mini-Robot for testing in the garage.

(b) Car detection in subway garage using tiny YOLO model.

Fig. 3: Mini Robot and Car detection

Neural Network Test Test of the neural network was performed on an NVIDIA P4000 GPU server with images taken in the garage and on the NVIDIA TX1 board which we used in the mini-robot in the garage with a C920 camera obtaining 15 fps which works without any problem with the tiny YOLO model.

Results of Cars Detection Detection of objects is very well identifying cars of general classes (See Fig. 3(b)), however for images where it is required to identify brand and model of the car is not get results since there is no dataset of images of front or back of the cars with annotations of their marks from which the neural network can learn. The nearest set of images found is Stanford dataset[12], but majority of images are not taken from front or back view. We trained with this dataset, but the images used for test were unlabeled. So, we discard this dataset for the model.

In contrast, identify digits of plates can get a better result because it is enough to locate the plate and perform a digit recognition [10]. ALPR is solved by [2] as well. Dataset for these tasks are SSIG dataset [14], a commercial dataset or UFPR-ALPR dataset[2] recently made with more fully annotated images and more vehicles in real-world scenarios for academic purposes.

After recognizing the digits of the plates, SUNARP online consultation[9] is accessed and information is obtained from the owner of the car (see Fig.4).

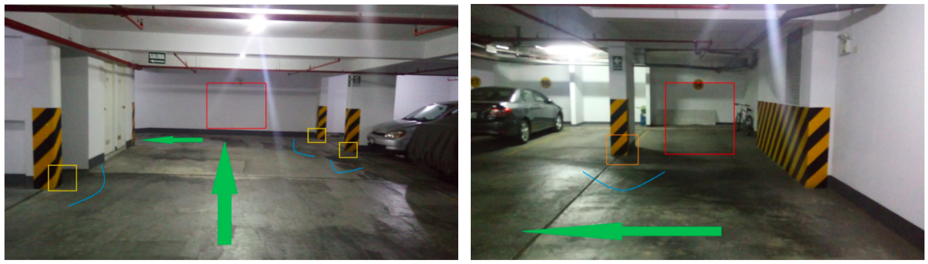
Fig. 4: Query of the owner of license plate in SUNARP online application[9].

2.4 Mini-robot mechanism

Mini-robot is built with base of a Monster Truck 1/18 which includes a motor and 2 servos for the movement of front and rear tires. An Arduino Uno is added that allows communication between the engine and the NVIDIA TX1 board, as you can see in Fig 3(a). The Arduino sends instructions to the engine through its GPIOs where indicates speed of the wheels and angle of rotation of the front wheels. Communication of the board TX1 to the Arduino is serial through a USB which allows to send integer values encoded in characters to indicate the rotation of the front wheels and the instructions to go forward, stop and rewind. Through python program on the TX1 board, instructions are sent to the Arduino and this send to the motor which angle have to turn.

2.5 Tenant Parking Mapping by Beacons

Summary In this section, we present the way that robot knows where is it in real time. This action can be regulated with Beacons that is useful for indoor location in places where the Internet is not enough to make a connection [6], even allowing tracking in real-time applications [7].



(a) Detection of objects, walls and signals of beacons. See the colors: yellow (beacons), red (object detection field), green (direction of route) and celestial (Bluetooth signal).

(b) Positioning of the beacons in the parking lot. See the colors: yellow (beacons), red (object detection field), green (direction of route) and celestial (Bluetooth signal)

Fig. 5: Environment and Beacon's location map

Detection of Beacons In garage studied, there is a separation distance of approximately 8 meters between walls and approximately 2.45 or 7.35 meters separation between vehicles, as shown in the Fig. 5(a) and Fig. 5(b).

As can be seen, Beacons have been distributed efficiently (see Fig. 6) that allows to determinate where it is located in real time, as well as identifies who owns that area due to the match between image recognition and the current position, based on previous work [8].

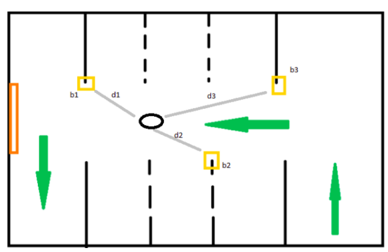


Fig. 6: Locations of the beacons in the garage and variables to be taken for the calculation and identification of the beacons.

Determination of the position of the mini robot In this section, we present the equations that define system to solve, by forming imaginary circles around each beacon, the respective radii are obtained. Then we proceed to calculate the relative position with respect to the global system of beacons based on distance equations forming a system of equations of the form:

$$E_i : (x - x_i)^2 + (y - y_i)^2 = d_1^2$$

for i=1, ..., 3

Where (x, y) is the current position of the mini-robot and the indexes i correspond to the beacons and $r_i, ..., r_3$ correspond to the respective vectors.

For this system formed proceeds to solve: Take (x_i, y_i) as coordinates of each beacon, we deduce $r_i = r_c + d_i$.

Where r_c is the current position of the mini-robot, for all equations the pair of indices i, j are not equal $(i \neq j)$.

So, The module is taken:

$$\|r_i\|^2 = \|r_c\|^2 + 2(r_c)(r_i) + \|d_i\|^2.$$

Calculating:

$$\|r_i\|^2 - \|r_j\|^2$$

Obtains:

$$r_c(d_i - d_j) = \|d_j\|^2 - \|d_i\|^2 + \|r_i\|^2 - \|r_j\|^2 = Y_i$$

By which we would have:

$$x_c(x_i - x_j) + y_c(y_i - y_j) = Y_i$$

Which forms a new linear system that is solved by numerical methods and represented as:

$$\text{Y} = \text{AX}.$$

Where:

$x = (x_c, y_c)^t$: es the column vector of the positions of mini-robot.

A: is the matrix forms by row vectors.

$$\text{A} = \begin{bmatrix} x_1 - x_2 \; y_1 - y_2 \\ x_2 - x_3 \; y_2 - y_3 \\ x_3 - x_1 \; y_3 - y_1 \end{bmatrix}$$

Y: is the column vector of differents between beacons.

3 Conclusions

We see that during the implementation of the robot car, it has to take into account many things that had not been foreseen to be able to handle the autonomous car.

In the work procedure, an optimal performance is obtained for the object detection task using Artificial Intelligence algorithms.

For the determination of positions of the mini robot based on the terrain delimited by columns and walls, it is necessary to specify the locations and the numerical system that solves the system of equations generated in such a way that a minimum error is obtained, which serves to determine and to identify the place where it is when we detect cars and objects of a certain car park, generating the relation object detected - location.

References

1. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, End to End Learning for Self-Driving Cars, arXiv preprint arXiv:1604.07316, 2016.
2. R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonalves, W. R. Schwartz, D. Menotti. A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. CoRR, vol. abs/1802.09567, 2018
3. J. Redmon, A. Farhadi. YOLO9000: Better, Faster, Stronger, CoRR, arXiv preprint arXiv:1612.08242. 2016.
4. D. Santo, Autonomous Cars' Pick: Camera, Radar, Lidar?, https://www.eetimes.com/ author.asp?section_id=36&doc_id=1330069. Last accessed 7 Jun 2018
5. CarND Self-Driving Nanodegree Udacity - Behavioral-Cloning https://youtu.be/e7D23Kdy72Q. Last accessed 7 Jun 2018.
6. Jinglong Li, Meifeng Guo, Siwei Li. An indoor localization system by fusing smartphone inertial sensors and bluetooth low energy beacons.Frontiers of Sensors Technologies (ICFST), 2017. Shenzhen, China.
7. Ievgen Gorovyi, Alexey Roenko, Alexander Pitertsev, Ievgen Chervonyak, Vitalii Vovk. Real-time system for indoor user localization and navigation using bluetooth beacons. Electrical and Computer Engineering (UKRCON), 2017 IEEE First Ukraine Conference.
8. Ryoya Momose, Tomoyuki Nitta, Masao Yanagisawa, Nozomu Togawa, An accurate indoor positioning algorithm using particle filter based on the proximity of bluetooth beacons. IEEE 6th Global Conference on Consumer Electronics (GCCE), 2017.
9. SUNARP, Servicio Gratuito de Consulta Vehicular con datos relacionados a las caractersticas de los vehculos registrados a nivel nacional. Available: https://www.sunarp.gob.pe/ConsultaVehicular. Last accessed 7 Jun 2018.
10. C. T. Hsieh, Liang-Chun Chang, K. M. Hung, Hsieh-Chang Huang. A real-time mobile vehicle license plate detection and recognition for vehicle monitoring and management. Joint Conferences on Pervasive Computing (JCPC), 2009.
11. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015.
12. Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13), 2013.
13. Chengmin Zhou, Fei Li, Wen Cao. Architecture design and implementation of image based autonomous car: THUNDER-1. Multimedia Tools and Applications, 2018. DOI: 10.1007/s11042-018-5816-9.
14. G. R. Goncalves, S. P. G. da Silva, D. Menotti, W. R. Schwartz. Benchmark for license plate character segmentation. Journal of Electronic Imaging, vol. 25, no. 5, pp. 053 034053 034. 2016.