International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10381-10388
© Research India Publications. http://www.ripublication.com

Product Detection System for Home Refrigerators implemented though a Region-based Convolutional Neural Network

César Giovany Pachón.
Research Assistant, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Javier Orlando Pinzón.
Research Assistant, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Robinson Jimenez Moreno.
Professor, Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá, Colombia.

Abstract

This paper presents the development of a product detection system in a refrigerator based on pattern recognition by the Region-based Convolutional Neural Network (R-CNN) technique. For this case, five types of products were selected, corresponding to butter, juice, milk, sauce and soda, as the objects of interest to be detected and, in addition, if any of the products is not detected, the user can check in a graphic interface which are the ones that are missing and are necessary to buy. A test scenario was implemented in order to observe the behavior of the network, obtaining a 94% of accuracy in the recognition of the objects, while in the final environment, the overall accuracy increase to 96.3%, with detection times between 0.68 and 0.96 seconds and, additionally, a RoI detection between 92% and 100% accuracy was achieved by the network

Keywords: R-CNN, RoI, Detection of Products, Alarm System.

INTRODUCTION

Researchers have been working for many years in the development of systems to facilitate and automate different tasks at home. For example, in [1,2] systems for energy saving, lighting and temperature control are presented, which demonstrates to a large extent the focus of some studies seeking the comfort of people in their homes. On the other hand, developments for houses have focused on safety systems such as those mentioned in [3] and recently in [4], where in the first, classic machine vision techniques are used to detect people and dangerous behaviors in order to guarantee safety of the houses, and in the second, a system based on fuzzy logic is designed for the detection of gas leaks.

Nowadays, there are very few areas that need to be addressed in the development of "intelligent" systems focused on household tasks, also systems for door and outlet control have been already implemented [5]. LG and Samsung have brought to market their first Smart refrigerators, which have integrated cameras that allow the user, from their smartphones, to see the products they have and even schedule the expiration dates of the elements so that they notify them when they are soon to expire. This latest application of intelligent systems opens up the possibility of carrying out developments or improvements such as the implementation of automatic object recognition systems in refrigerators.

One of the techniques most currently used for the recognition of objects are the convolutional neural networks (CNN) that has its beginnings in the 80s, where it was presented as a system for the recognition of visual patterns [6]. In 1989, implementations focused on the recognition of written numbers started to be developed [7]. Recently, some studies highlight the importance of CNN, its general functioning and the operation of each of its layers based on visualization techniques such as deconvnet [8]. Currently, several deep learning systems based on convolutional neuronal networks are being studied for the detection of objects such as those mentioned in [9], where CNNs are implemented for fruit detection, or in [10] for detecting traffic signals. But they are not only focused on detection tasks, since in other works are employed for other uses, such as in [11], where they are applied in the segmentation of brain images, while in [12], they are used to classify different types of radio galaxies. The above shows that the use of CNN is becoming very important in the development of new techniques and technologies.

Currently, there are different types of techniques for the detection of multiple objects in an image, an example of them is presented in [13], where a technique called Region-based CNN (R-CNN) is described which detects objects from proposed regions, presenting as an advantage the simplicity of its implementation, but with high training times. In order to improve the times proposed by the R-CNN, the Fast R-CNN is implemented, where in [14] one of its applications oriented to the detection of pedestrians is shown. More recently, the Faster R-CNN is presented as an improvement of the previous two, obtaining better times in the classification given the selective search that it carries out to detect the regions of interest. This type of architecture has been implemented in face recognition projects [15] or the detection of vehicles in real time [16], presenting as a disadvantage the increase in complexity in its implementation.

In this article the CNN was implemented for the detection of objects in a refrigerator, in order to generate an alarm when any of the products is finished. For the application exposed in this article, it is not necessary to use a fast or faster R-CNN, since in the case of an implementation in a real environment, video processing or relatively low object detection times will not be required.

The article is divided in three sections, in the first part corresponding to materials and methods, the designed architecture, CNN training parameters, databases and CNN tests are presented. The second part focuses on the results obtained in the implementation of the system with the camera in the fridge, activations of the CNN layers, cases of non-detection of products and the graphic user interface. Finally, the conclusions of the presented development are shown.

METHODS AND MATERIALS

Database (case: Test)

For a first approach to the development of an object recognition system in a refrigerator, it is established for the acquisition of the database the location of the camera just in front of the refrigerator and at a fixed distance, as can be seen in figure 1A. In figure 1B one of the captured images is shown, for this case, the images are 320x240 pixels. For the implementation and training of an R-CNN, the regions of interest are required, in this case, five products are selected (Juice, butter, milk, soda, sauce) (see figure 1C).

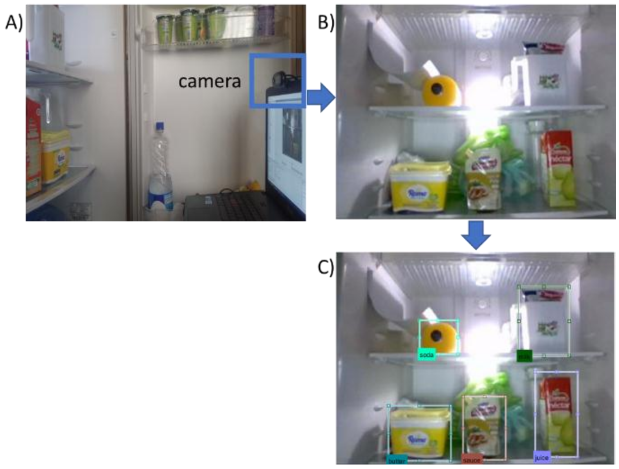


Figure 1. Test Environment

To generate the first database, 50 photos were taken with all the elements of interest in the captures; in order to perform an augmentation of the images, the data augmentation algorithm worked in [17] was used, making changes in lighting in the training images to generalize the R-CNN and make it robust to these changes, so that it is possible to increase the training database up to 200 images.

Architecture

In figure 2, the proposed architecture is shown for the R-CNN where its input is 32x32 pixels with a depth of 3 channels (RGB). This input size is set by looking in the boxes of the objects of interest the smallest size among all the edges. Then, the image goes through two layers of convolution, in which different amounts of filters are applied in order to let the network to learn the characteristics of the image. Subsequently, a MaxPooling is applied to reduce the size of the image and improve the computational cost. A third convolution is applied to continue learning more detailed characteristics, to finally apply again a reduction in size, obtaining an output image of 7x7 pixels, which is the input of the first Fully-connected layer, where it is wanted to learn additional features. To generalize the network for any image and avoid going to overfitting, a Dropout of 50% is applied, in this way a disconnection of half of the neurons is done in a random way, so that the network does not memorize the images and learn general characteristics. Finally, a fully-connected output is added with a size of 6 neurons, referring to each of the existing categories (five objects of interest and the background). This result is entered into the Softmax layer, in charge of normalizing the values to a percentage result of membership of each object to each of the categories [8].

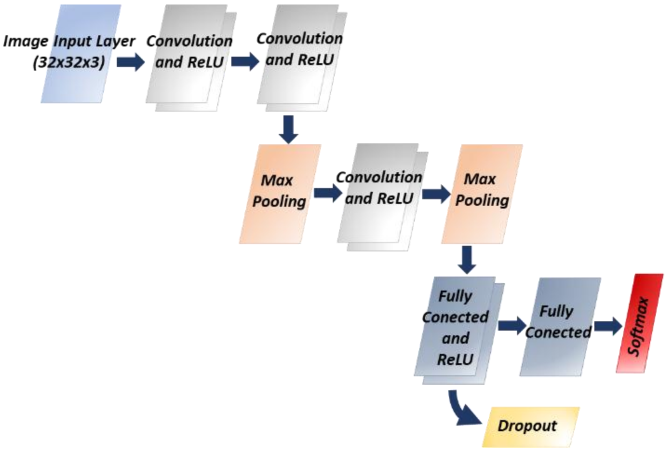


Figure 2. Implemented Architecture.

For each of the layers it is necessary to set the parameters of the network, so in Table 1, the values configured, such as Kernel and filters, are shown, where S refers to the size of the Stride and P to the Padding.

Table 1. Execution time of each CNN configuration

Layer	Kernel	Filters
Input	32x32x3	-
Convolution	4x4 S=1 P=1	32
Convolution	4x4 S=1 P=1	128
Maxpooling	2X2 S=2 P=0	-
Convolution	4X4 S=1 P=1	256
Maxpooling	2X2 S=2 P=0	-
Fully-Connected	1	512
Fully-Connected	1	6
Softmax	-	-

Training options

To perform the training, it is needed 3 things: the database, the CNN and the training parameters. Taking into account the size of the database (200 images), the batch size is set to 40 images, so that every 5 iterations of training equals 1 epoch. The number of epochs is set from training tests, and the Learning Rate is set to 0.0001 in order that the weights not to vary drastically during training (see Table 2).

Table 2. Training options

Batch size	40
Training Epochs	250
Learning Rate	0.0001

Training, testing and RoI detection (case: Test)

In figure 3, the training of the R-CNN is shown, obtaining a training loss of 0.0035 and precision in the classification of training images of 100% as a result of the last iteration.

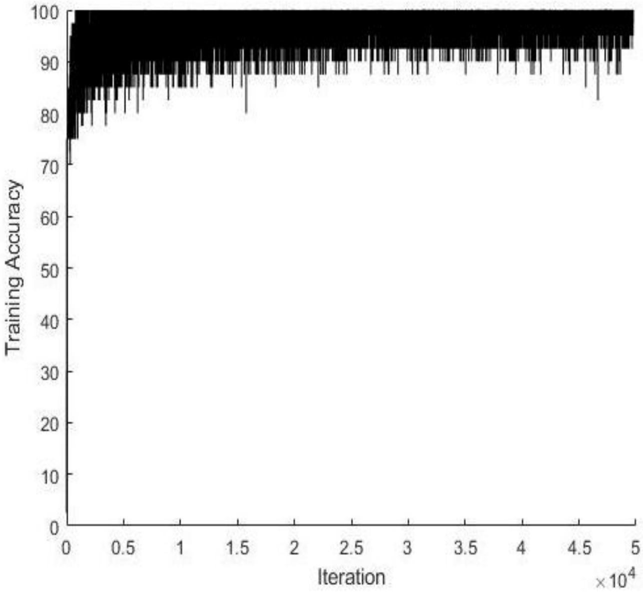


Figure 3. Training for the test environment.

In order to verify the correct functioning of the R-CNN, a test database of 40 images is set, which contains all the objects of interest. It is proceeded to build a confusion matrix, in which the images are evaluated and in what categories they were classified, where, in the diagonal, it is shown the number of objects correctly classified in each of the categories (see figure 4). Additionally, it can be seen that any of the objects were misclassified as another object, but not found in the image.

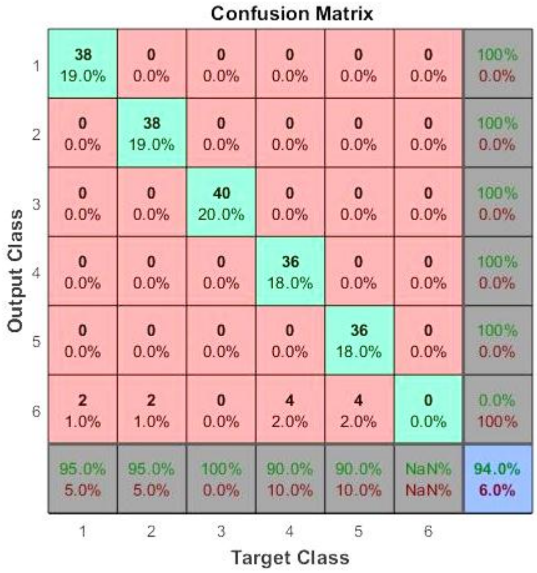


Figure 4. Confusion matrix of the test environment, where category 1 = butter / 2 = juice / 3 = milk / 4 = sauce / 5 = soda.

A total 94% accuracy of the network is reached, where the elements with the lowest percentage of precision are soda and salsa, this may be caused by different reasons, such as the size of the database implemented or the initial weights. It should be noted that for this application that precision is significantly high. figure 5 shows the accuracy of the Region of Interest (RoI), observing that significant high values are presented, between 0.86 and 0.95.

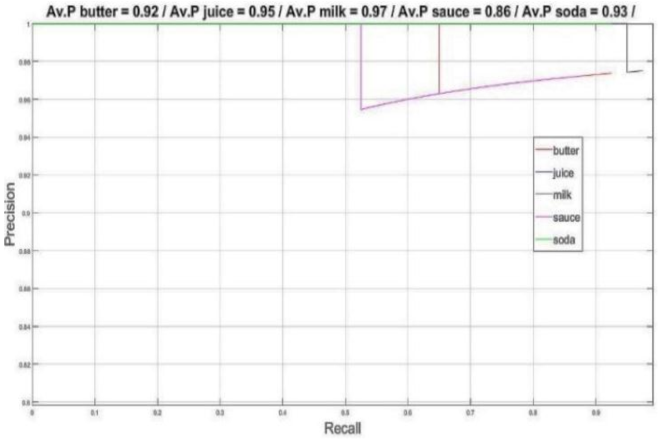


Figure 5. RoI detection in the test environment.

Database (Case: Real workspace)

Based on the previous results, it can be concluded that the implemented architecture is robust enough for the application. Now it is sought to locate the camera in a position inside the refrigerator and generate a wider database, in order to make the R-CNN more robust and that can be implemented in a real application. In figure 6A the position of the camera in the fridge door is shown, this in order to have an appropriate and practical system for a real implementation. In figure 6B, one of the captured training images is observed; it should be noted that for the acquisition of this new database, the number of objects of interest was varied, and even the background objects were changed in order to be able to generate a bigger database. A total of 1300 images are obtained, of which 1170 are used for training and 130 for the test, it should be noted that in each of the test images there are five objects of interest, so that when generating the confusion matrix and calculating the accuracy percentage of the CNN, it will be affected in the same way by all the objects. In figure 6C, it can be seen the boxes of interest that will be entered for training.

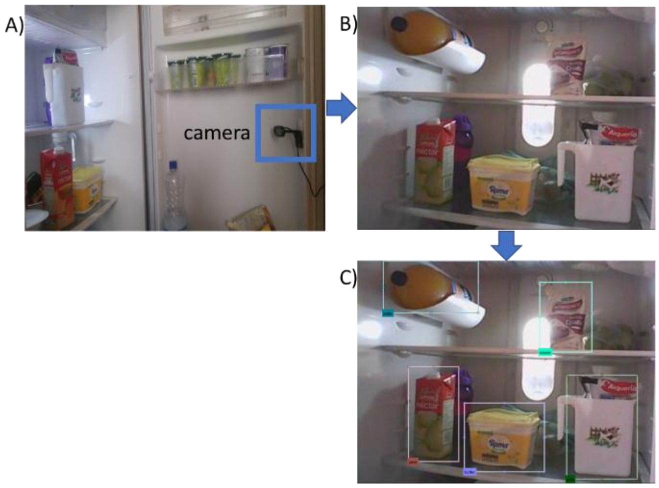


Figure 6. Work environment with the camera in the fridge.

Training, testing and RoI detection (Case: Real workspace)

It should be noted that the training parameters change, since now the training database is 1170 images, the batch size is set to 45 and the other values remain the same. In figure 7, the training of the R-CNN is shown, where a result in the training loss of 0.0015 and 100% accuracy in the classification of the training images were obtained in the last epoch.

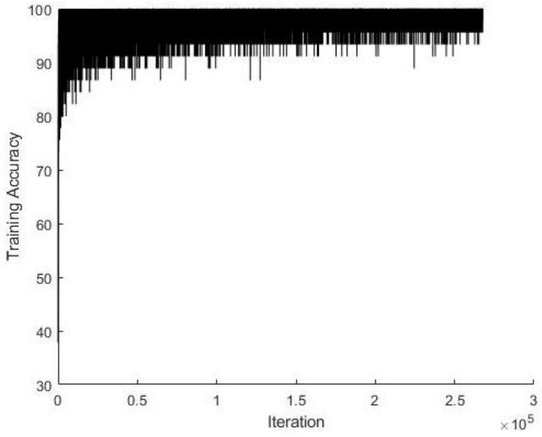
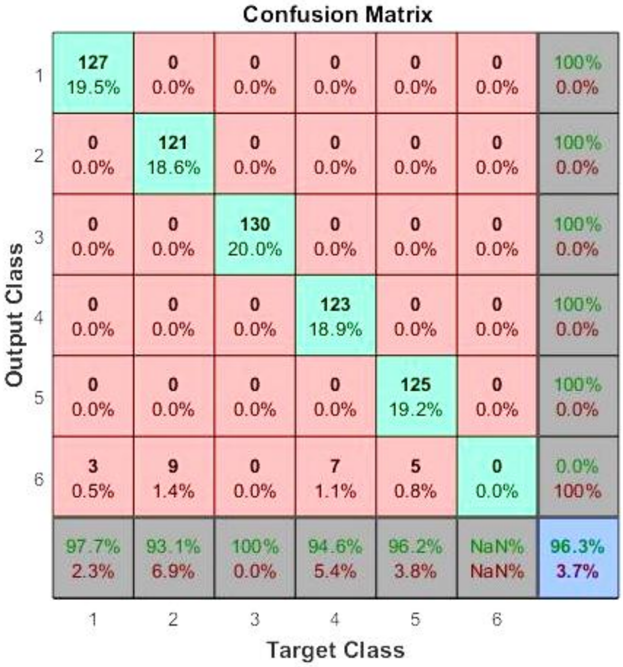


Figure 7. Training of the CNN in the work environment.

To carry out the verification of the R-CNN with this new training, a confusion matrix is made, in this case with 130 test images (see figure 8). There is an increase of 2.3% in accuracy compared to the previous training, showing an increase in accuracy in all categories except juice with 93.1% accuracy. It is noteworthy that no image of any category was confused with another, the error rates in the predictions that are presented in each of the categories, except milk, are due to products that were not recognized.



Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	
1	127 19.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	121 18.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	130 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	123 18.9%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	125 19.2%	0 0.0%	100% 0.0%
6	3 0.5%	9 1.4%	0 0.0%	7 1.1%	5 0.8%	0 0.0%	0.0% 100%
	97.7% 2.3%	93.1% 6.9%	100% 0.0%	94.6% 5.4%	96.2% 3.8%	NaN% NaN%	96.3% 3.7%

Figure 8. Confusion matrix of the final work environment, where category 1 = butter / 2 = juice / 3 = milk / 4 = sauce / 5 = soda.

The precision in the RoI presents an increase having values between 0.92 and 1, this shows that the generated bounding boxes have a high overlapping, this may be because the sizes of the boxes in the dataset do not vary significantly (see figure 9), so the precision when locate it increase.

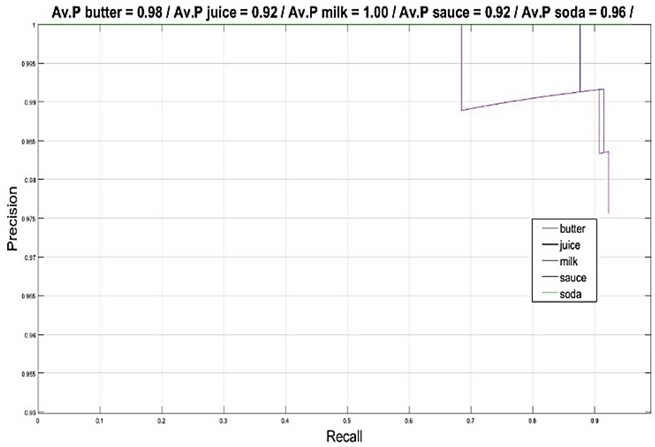


Figure 9. RoI detection in the final work environment.

RESULTS AND DISCUSSIONS

RoI detection and layer activations

Once the R-CNN is trained, tests are carried out in the real practice environment. It seeks to verify the correct functioning of the system, its detection of the objects of interest, in addition to the activations that are generated in the main layers of the R-CNN. In figure 10A, one of the test cases is observed, and in figure 10B the detection of RoIs with their respective scores.

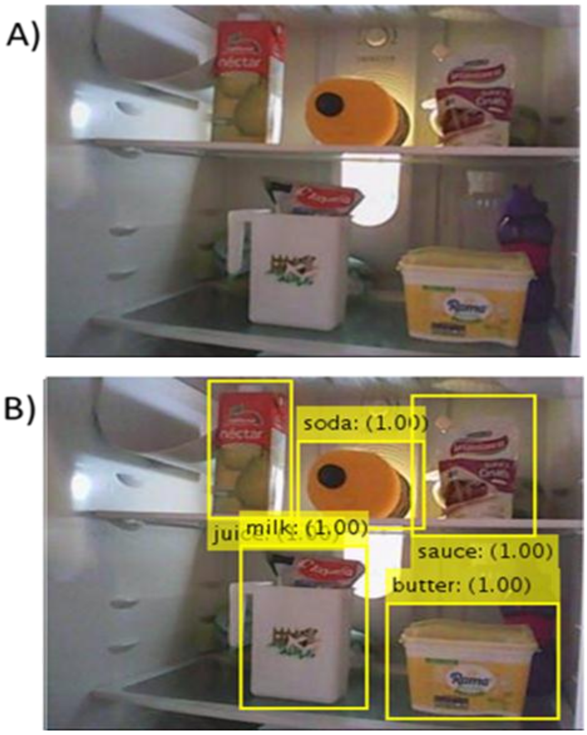


Figure 10. Test image and RoI detection.

All the scores for this test had a membership value of approximately 1. In figure 11 the strong activations of the R-CNN are marked in violet, these activations are presented in a large part in the objects of interest, demonstrating that the R-CNN was able to learn and to differentiate the objects in the image.

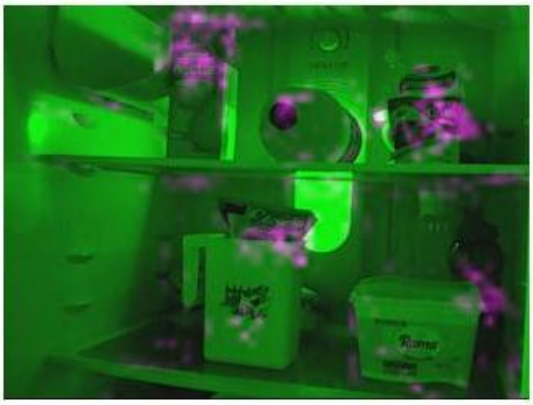


Figure 11. Activations of R-CNN trained in the test image.

In figure 12, it can be seen the activation in each convolution layers after being rectified. In the first column is the original image of each of the objects to be recognize. In the second it can be seen how activations are generated not only of specific parts of the objects, but also of the background. In the third, the learning of edges of some of the objects is emphasized. In the fourth, the internal and external edges of the objects are more clearly defined, and finally, in the last layer, the activations are weak, possibility related to internal characteristics of each object.

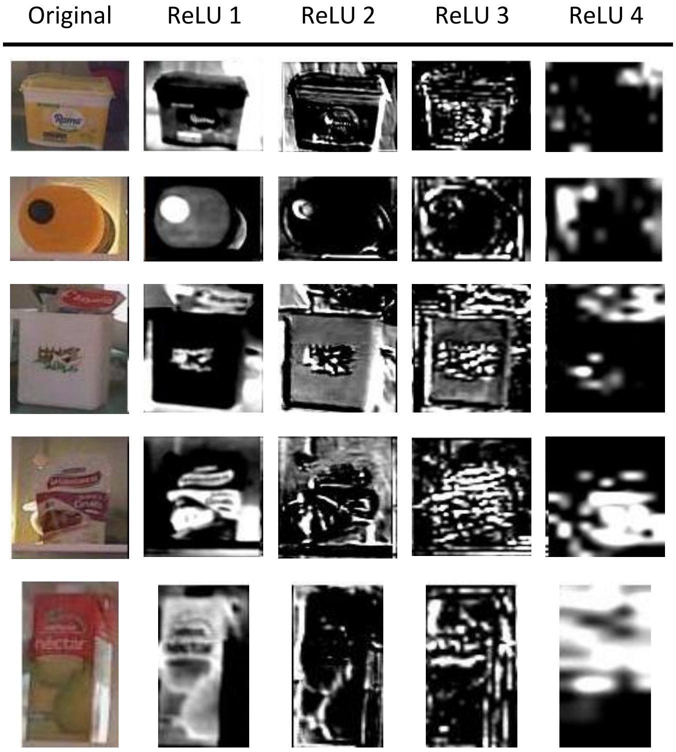


Figure 12. Activations per convolution layer of each detected element.

Positive cases, no detection and detection times

During the R-CNN tests, most of the cases correctly located the objects with its RoI also right recognized (see figure 13A), but some cases of error were also presented (see figure 13B). These cases of error were mostly presented because the background lighting made the edge of the objects unclear and in other cases because the edges of the object merged with the background and activation was not generated.

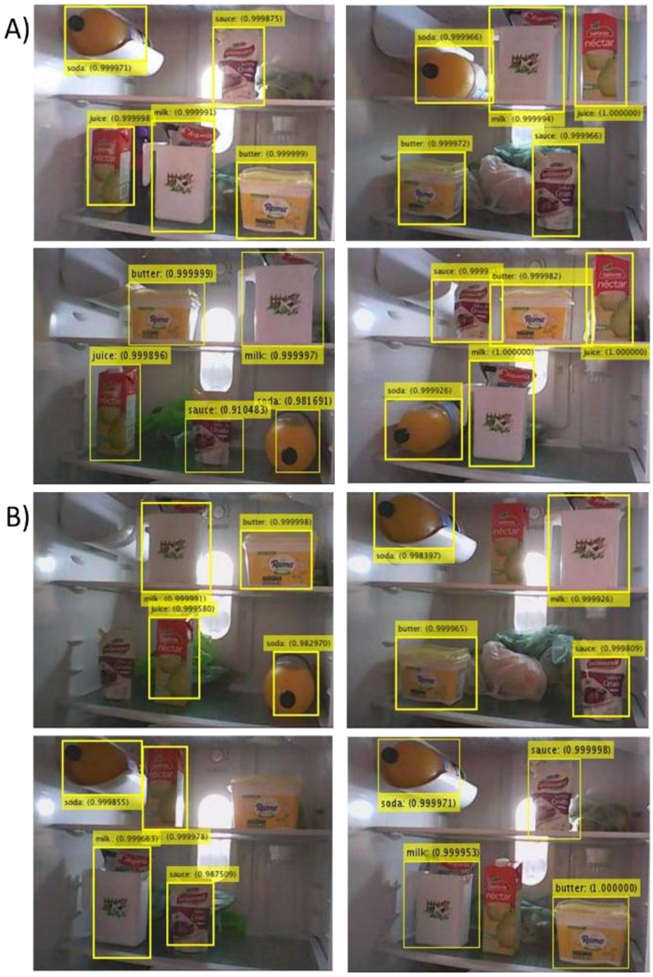


Figure 13. Cases of right and wrong detection.

An important part of the implementation of this type of systems is to know the ranges of execution times that it would take for the R-CNN to detect one or more objects within the image, in order to conclude if its implementation is viable in the application for which it was designed (see Table 3).

Table 3. Object detection times

	TIME (s)
0 Objects	0.68-0.71
1 Object	0.7-0.72
2 Objects	0.84-0.85
3 Objects	0.90-0.92
4 Objects	0.88-0.92
5 Objects	0.93-0.96

Graphic User Interface

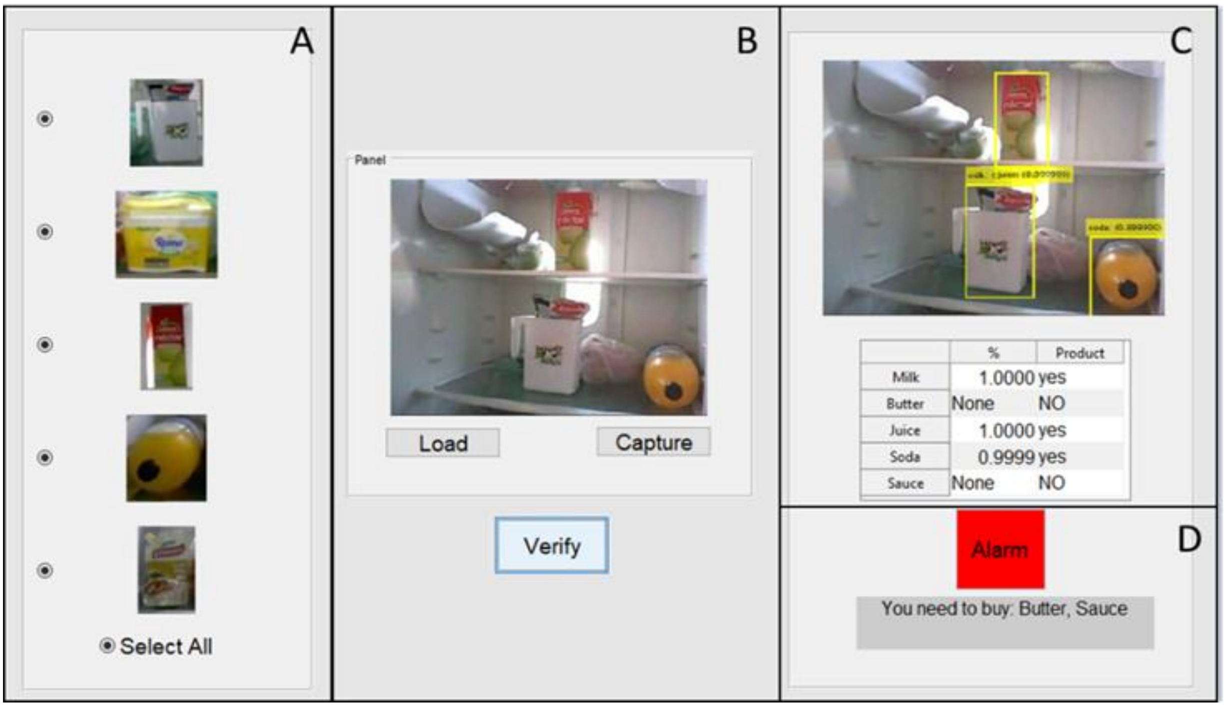
A graphic user interface was designed, which allows selecting the products from which it is required to generate a purchase need alarm in case there is not in the detection (see figure 14A). The image captured by the camera is shown in Figure 14B. When the verification is made in the section that can be observed in figure 14C, the detected RoIs are shown, their accuracy and whether or not the product is found. Finally, if all the elements selected in section A are found, a sign is generated indicating that there is no need to buy products; In the event that the opposite occurs and there is no product, the alarm message changes color and will tell the user that there is the need to buy products and says which ones are missing (see figure 14D).

CONCLUSIONS

The implementation of a test case for the design of the architecture, selection of training parameters and database, allowed to generate a detailed approach to the problem and a more precise result in the implementation in the final work environment, also it allow to show that a larger and more varied database was necessary, and that the architecture had an adequate functioning. In the test, an accuracy of 94% was achieved and with the improvements implemented in the database for the implementation in the final work environment, 96.3% accuracy was obtained.

Taking into account the high percentage of precision of the implemented system and that the processing times for the detection of objects are less than 1 second, it can be concluded that the implementation of this system in refrigerators is viable, thus providing a novel and automatic system for recognizing objects of interest with a configurable alarm that alerts the end user when products are required or not.

Based on the activations of the different layers of the R-CNN, it can be seen that in ReLU 4 the activations are not relevant and the network is not learning significant details. Although the accuracy is high, it allows to show a point of improvement in the architecture. Another point to keep in mind is that, in the acquisition of the database, the background objects and the quantity of objects of interest in each image were varied, but, in order to further generalize the R-CNN, it is necessary to perform tests in other types of refrigerators.



	%	Product
Milk	1.0000	yes
Butter	None	NO
Juice	1.0000	yes
Soda	0.9999	yes
Sauce	None	NO

You need to buy: Butter, Sauce

Figure 14. Graphic user interface for the detection and generation of alarm of elements in inventory

ACKNOWLEDGEMENT

The authors are grateful to the Nueva Granada Military University, which, through its Vice chancellor for research, finances the present project with code IMP-ING-2290 (being in force 2017-2018) and titled "Prototype of robot assistance for surgery", from which the present work is derived.

REFERENCES

[1] Han, D., and Lim, J., 2010, “Smart home energy management system using IEEE 802.15.4 and zigbee,” IEEE Transactions on Consumer Electronics., 56(3), pp. 1410-1410. doi: 10.1109/TCE.2010.5606276.

[2] Mao, X., Li, K., Zhang, Z., and Liang, J., 2017, “Design and implementation of a new smart home control system based on internet of things,” International Smart Cities Conference., pp. 1-5. doi: 10.1109/ISC2.2017.8090790.

[3] Cucchiara, R., Prati, A., and Vezzani, R., 2007, “A multi-camera vision system for fall detection and alarm generation,” *Expert Systems.*, 24(5), pp. 334-345. doi: 10.1111/j.1468-0394.2007.00438.x.

[4] Kumar, K., Sen, N., Azid, S., and Metha, U., 2017, “A Fuzzy Decision in Smart Fire and Home Security System,” Procedia Computer Science., 105, pp. 93 – 98. doi: 10.1016/j.procs.2017.01.207.

[5] Piyare, R., 2013,“Internet of things: ubiquitous home control and monitoring system using android based smart phone,” International Journal of Internet of Things, *2*(1), pp. 5-11. doi: 10.5923/j.ijit.20130201.02.

[6] Fukushima, K., and Miyake, S., 1982, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” In Competition and cooperation in neural nets., pp. 267-285. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-46466-9_18.

[7] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D., 1989, “Backpropagation applied to handwritten zip code recognition. Neural computation,” 1(4), pp. 541-551. doi: 10.1162/neco.1989.1.4.541.

[8] Zeiler, M.D., and Fergus, R., 2014, “Visualizing and understanding convolutional networks,” In European conference on computer vision., pp. 818-833. Springer, Cham. doi: 10.1007/978-3-319-10590-1_53.

[9] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C., 2016, “Deepfruits: A fruit detection system using deep neural networks,” Sensors., 16(8), p. 1222. doi: 10.3390/s16081222.

[10] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., and Hu, S., 2016, “Traffic-sign detection and classification in the wild,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition., pp. 2110-2118. doi: 10.1109/CVPR.2016.232.

[11] Moeskops, P., Viergever, M.A., Mendrik, A.M., de Vries, L.S., Benders, M.J., and Išgum, I., 2016, “Automatic segmentation of MR brain images with a convolutional neural network,” IEEE transactions on medical imaging., 35(5), pp. 1252-1261. doi: 10.1109/TMI.2016.2548501.

[12] Aniyan, A.K., and Thorat, K., 2017, “Classifying Radio

Galaxies with the Convolutional Neural Network", The Astrophysical Journal Supplement Series., 230(2), p.20. doi: 10.3847/1538-4365/aa7333.

[13] Girshick, R., Donahue, J., Darrell, T., and Malik, J., 2014, "Rich feature hierarchies for accurate object detection and semantic segmentation," In Proceedings of the IEEE conference on computer vision and pattern recognition., pp. 580-587. doi: 10.1109/CVPR.2014.81.

[14] Li, J., Liang, X., Shen, S., Xu, T., Feng, J., and Yan, S., 2018, "Scale-aware fast R-CNN for pedestrian detection," IEEE Transactions on Multimedia., 20(4), pp. 985-996. doi: 10.1109/TMM.2017.2759508.

[15] Jiang, H., and Learned-Miller, E., 2017, "Face detection with the faster R-CNN," In Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on., pp. 650-657. IEEE. doi: 10.1109/FG.2017.82.

[16] Zhang, Y., Wang, J., and Yang, X., 2017, "August. Real-time vehicle detection and tracking in video based on faster R-CNN," In Journal of Physics: Conference Series., 887(1), p. 012068. IOP Publishing. doi:10.1088/1742-6596/887/1/012068.

[17] Useche, M.P.C., Arenas, J.P. and Moreno, R.J., 2017, "Implementation of a data augmentation algorithm validated by means of the accuracy of a convolutional neural network," Journal of Engineering and Applied Sciences., pp. 5323-5331. doi: 10.3923/jeasci.2017.5323.5331.