Research Journal of Applied Sciences 13 (1): 41-46, 2018
ISSN: 1815-932X

Convolutional Neural Network Training for Robotic Applications in 3D Environments

[1]M. Robinson Jimenez, [2]S. Oscar Aviles and [3]Diana M. Ovalle
[1, 2]Faculty of Engineering, Nueva Granada Military University, Bogota, Colombia
[3]Faculty of Engineering, University Distrital Frco. Jose de Caldas, Bogota, Colombia

Abstract: This study presents two training schemes of three deep convolutional neural network architectures applied to object recognition, based on the depth information supplied for a 3D camera. For this case, the depth information allows to make the set of training images of each network, its architecture and its characteristics, generating a dynamic recognition application by variation of the image capture point. The best scheme is selected to add a weighting layer with saturationn for obtain a final architecture that recognize objects to different distances with a 91.69% success that mean a maximum error of 8.31%.

Key words: Convolutional neural network, robotic applications, 3D environment, characteristics, dynamic, information

INTRODUCTION

In recent years, deep learning techniques have led the field of artificial intelligence (Schmidhuber, 2015) and presenting developments in various fields of science and engineering. Its ability to recognize patterns allows applications from the analysis and extraction of characteristics of large amounts of data (big data) (Zhang *et al*., 2016) to new control systems for vehicular traffic (Fadlullah *et al*., 2017). In relation to this last application, specific techniques of deep learning such as Convolutional Neural Networks (CNN), allow the implementation of artificial intelligence systems based on images for example, the detection of pedestrians (Orozco *et al*., 2016).

The CNNs have been validated in their high capacity of learning based on images (Krizhevsky *et al*., 2012) either in grayscale or in color and have become popular in this field. They have a basic structure of consecutive layers of convolution, linear rectification and pooling (Zeiler and Fergus, 2014a, b) which allow the generation of very deep learning architectures (Qain *et al*., 2016). Thus, one of the current fields of application of CNN is the development of robotic agents based on artificial intelligence for example for tasks of recognition of places that may lead to autonomous navigation by the robot (Mancini *et al*., 2017).

The control of robotic agents through artificial intelligence systems has been under development for several decades within the objectives that it seeks are the autonomy and precision in the execution of tasks for example in product assembly processes (Warczynski, 2000) or cleaning tasks (Hossen *et al*., 2017). By means of convolutional neural networks by Yang *et al*. (2017), it is presented a robotic humanoid development capable of performing natural tasks for a human such as the folding of garments, thus, illustrating the scope of this deep learning technique in robotics.

The generality of robotic activities is presented in three-dimensional environments with exceptions such as the planning of trajectories in a plane. Applications with robotic arms, very common in the industry, require manipulating the gripper in three-dimensional space and positioning them according to the task to be performed. By Redmon and Angelova (2015), a development is presented by convolutional networks for gripping tasks by means of a robotic gripper without including aspects such as the depth of the object.

In general, there are not many developments in the area of robotics associated with applying convolutional neural networks in tasks that include depth data in a 3D scene. Some of the developed developments address different aspects for example by Porzi *et al*. (2017), a new convolutional network model is developed that integrates the depth information which allows to discriminate objects spatially with direct applications in robotics. By Wang *et al*. (2016), it is presented an approximation to the work shown in this study where an RGB-D signal is used

Corresponding Author: M. Robinson Jimenez, Faculty of Engineering, Nueva Granada Military University, Bogota, Colombia

which represents a color image plus the depth information of the scene with which objects are segmented into the image and then captured by a robotic gripper.

In this study, it is presented a proposal for the dynamic recognition by means of convolutional neural networks for cases where the image capture point variation occurs, i.e., the RGB-D sensor goes in the robotic agent and it moves. The above raises the problem of change of perspective of the object to be recognized, thus proposing a solution not found in the state of the art.

MATERIALS AND METHODS

CNN architecture: The solution proposed consists of the training of a convolutional neural network specialized in the different perspectives of the object to be recognized, depending on the depth in which it is, i.e., the training of a set of neural networks is realized where each one will learn to recognize the object from the distance and will be activating each network as it approaches the object.

Two options are proposed for validation, the first consists of a pre-trained convolutional network that through learning transfer, allows to obtain a generic network that will be trained according to depth information, under each perspective of the object. In this aspect, there will be determined three convolutional neural networks of the same characteristics that will be trained with different database given by the distance perspectives of the object. The second option consists in determining each of the three architectures of the convolutional networks independently.

Transfer learning CNN architecture: For applications of object recognition in images, different CNN architectures have been developed, ranging from the most basic that comprise only a convolution layer and a layer of pooling (Porzi *et al*., 2017) to very deep architectures with up to 19 convolution layers. Where different combinations of convolutional layers, Rectifier Linear Unit (RELU) and pooling layers has been used.

In a previous development, a convolutional network was obtained for tool discrimination of five different categories, whose efficiency was 96% in the prediction, the associated architecture is shown in Fig. 1.

Its configuration consists of 2 convolution-maxpooling sets and a convolution-maxpooling set in feature learning stage. The complete architecture is shown in Table 1 where S is the stride used, the input consists in color images of different sizes but normalized by resizing to 64×64 pixel. Under this

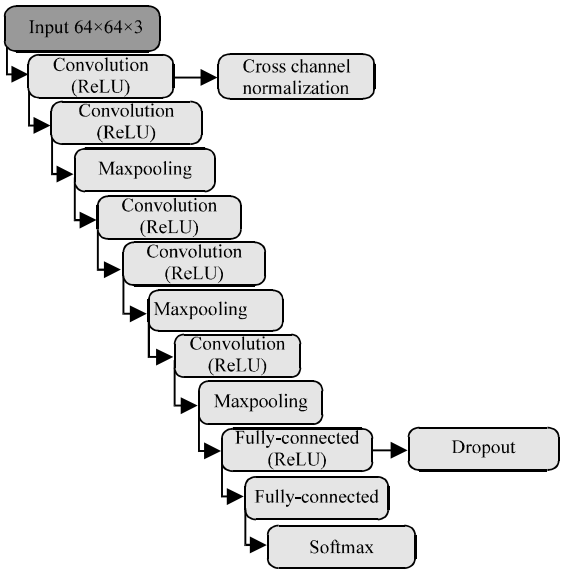

Fig. 1: CNN architecture employed for transfer learning

Table 1: CNN architecture

Layers	Kernel	Filters
Input	64×64	-
Convolution	10510 (S=1)	10×10
Convolution	15×15 (S=1)	15×15
Maxpooling	252 (S = 2)	2×2
Convolution	10×10 (S = 1)	10×10
Convolution	10×10 (S = 1)	10×10
Maxpooling	2×2 (S = 2)	2×2
Convolution	7×7 (S = 1)	7×7
Maxpooling	2×2 (S = 2)	2×2
Fully-connected	1	256
Fully-connected	1	-
Softmax	5	-

Table 2: Images database

Net/Depth	Screwdriver	Nippers	Pliers
1/20 cm	100	100	100
2/40 cm	150	150	150
3/60 cm	200	200	200

architecture which allowed discriminating between scissors, screwdrivers, cutting pliers, pliers and the category "others", the prediction of two types of tools with image captures at different depths was validated as shown in Fig. 2. It can be evidenced that as the object approaches, it does not manage to be identified for this case a 43.2% error is obtained in the prediction.

Because the pre-trained network presents a high performance in the recognition of tools, it is used as the basis to carry out the learning transfer task where for each of the three networks trained by transfer uses the image database illustrated in Table 2.

The difference in the amount of images used by each network according to depth is determined by the amount of characteristics that the filters must learn in training, the closer the image the less variations are found.

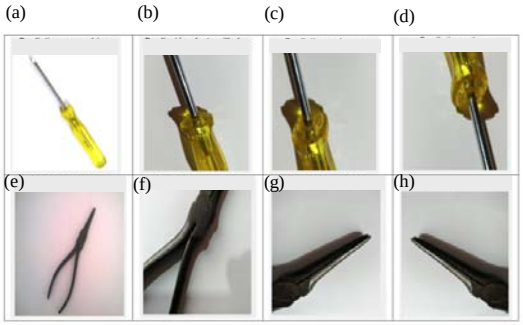

Fig. 2: CNN response to different depths Prediccion: a) Screwdriver; b) Destornillador; c) Scissors; d) Others; e-f) Nippers and g-h) Others

Table 3: Architecture based on depth

	Architecture	
Name/Type	Kernel	Filters
ARQ 1		
Convolution	10×10 (S = 2)	10-10/50
Max pooling	8×8 (S = 2)	20-20
Convolution	5×5 (S = 2)	30-30
Maxpooling	10×10 (S = 2)	40-40
Fully-connected	1	
Softmax	5	
ARQ 2		
Convolution	26×26 (S = 2)	10-10/30
Maxpooling	6×6 (S = 2)	
Convolution	6×6 (S = 2)	
Maxpooling	6×6 (S = 2)	
Fully-connected	1	
Softmax	5	
Max pooling	5×5 (S = 2)	
Convolution	7×7 (S = 3)	
Fully-connected	1	
Softmax	5	

CNN architectures based on dept: For the second option, three architectures were implemented to test their performance in depth recognition. The structure shown in Fig. 1 is maintained for the network oriented to recognize from far (60 cm). The other two architectures have the structures shown in Table 3.

The differences in the architectures are due to the complexity of the learning that each of the networks must handle according to the amount of information in the image. The greater the distance from the image capture to the object, the more information enters to the network which must clearly discriminate the object from the background, for example, a big part of the background will be taken which must be discriminated from learning with respect to the object.

CNN architecture training: Each of the implemented architectures was trained with the input dataset shown in Table 2 where that was set in 80% data and 20% of validation. To analyze the training behavior of each

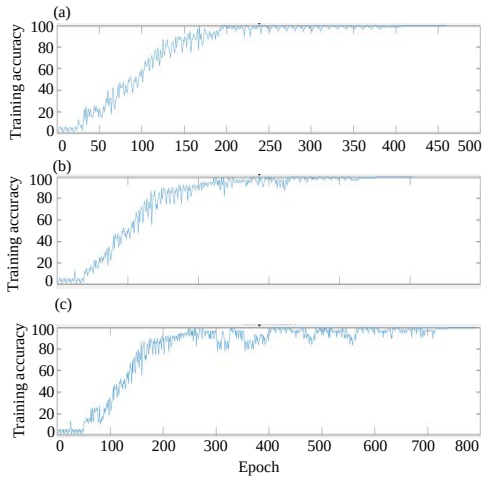

Fig. 3: Training results of transferleraning method: a) Training architecture 1; b) Training architecture 2 and c) Training architecture 3

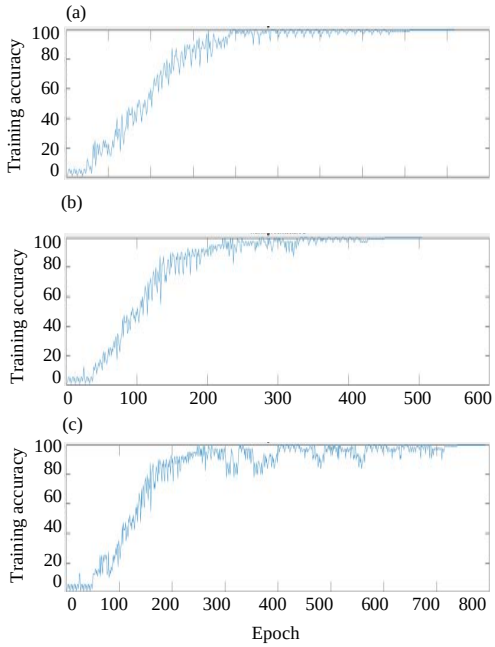

Fig. 4: Training results of based on depth method; a) training architecture 1; b) Training architecture 2 and c) Training architecture 3

architecture, it was evaluated the relationship training accuracy vs. epochs. Figure 3 illustrate the resulting training behaviors of architecture 1-3, respectively for transfer learning and Fig. 4 illustrate the same result but for architectures based on depth. In Fig. 3 and 4, it can be seen that architecture 1 had a faster learning curve than architecture 2 and 3 due to the fact that architecture 1 only visualizes the segments of objects and a background's small part. On the other

Table 4: Training time (min)

Cases	Architecture 1	Architecture 2	Architecture 3	Prom
TL	5.20	6.17	7.22	6.29
BD	3.13	4.08	7.22	4.81

hand, architecture 3 recognize complex patterns of feature maps, hence, this can be evidenced in its delay in begin to learn due to it must discriminate the background from the object in this case in a greater proportion. However, the three architectures achieved a 100% training accuracy in their trainings for which in practice, it is what is expected to ensure that the network will achieve an efficient performance in the task to which it will be destined.

Likewise, it is evident that the complexity of the architecture and the learning object regarding the image, affect the training time. For the case, architecture 1 requires half of training epochs that architecture 3 needs as can be observed in Fig. 4. Table 4 shows the training time needed in each option, Transfer Learning (TL) and Based on Depth (BD) where the last was faster.

RESULTS AND DISCUSSION

In order to determine the performance of each architecture, the predictive capacity by scenario is evaluated, according to the depth of the image capture, case A by learning transfer and case B by individual design of each convolutional network. In Table 4, it can be seen the error rate obtained which in general allows to conclude that case B gives the best recognition at dynamic depth with less training time and a simpler architecture.

The results shown in Table 5 are obtained from the individual validation of each network. Because the development functionality is oriented to dynamic depth change where the prediction must be performed in real time, a final joint architecture is used with a weighting output layer based on the depth of capture in the input image. Figure 5 illustrates the final convolutional neural structure used.

The additional weighting layer generates a weighted sum of the individual responses of each output of the networks of the final architecture as a function of depth, according to Eq. 1. This weighting was set analytically for the development of the present research:

$$p_c = \sum_{i=1}^{n} (1+O_n)^1/(n-d) \quad (1)$$

Table 5: Prediction error

	Error rate (%)			
Cases	Architecture 1	Architecture 2	Architecture 3	Prom
TL	12.67	10.72	6.6	9.99
BD	10.56	7.87	6.5	8.31

Table 6: Result of screwdriver by depth identification

Depth (cm)	Architecture activated	Weighting
60	3	10
30	2-3	5.66
20	1	10
20	1	10

Where:

P_c = The prediction by category

n = The number of networks that have the final architecture (3 for the case)

O_n = The output of each network by category

d = The normalized distance of the RGB-D capture camera to the object

For the case, the normalized distance is taken as the distance in centimeters of the camera over the minimum distance that distinguishes (20 cm).

The saturation function used which is observed in the output layer in Fig. 4 becomes necessary because the exponent in Eq. 1 tends to infinity when the corresponding network is activated, the upper saturation limit is 10 and the lower is 0.

In Fig. 6, it can be seen the result of the prediction of the convolutional neural architecture designed, based on depth. It is appreciated that the images used are satisfactorily recognized when the RGB-D camera is approached towards the objects of interest. In this case, the error is reduced to 8.31%.

Where for each column of Fig. 6, the depth catch ratio is shown in Table 6. In it is observed that for the training distance values the output is saturated whereas for intermediate values it is weighted by the respective activations For example, for a distance of 50 cm, the resulting weighting is 4.74, derived from the activations of the Architectures 2 and 3 which without becoming saturated their value exceeds the value of the other classes.

The obtained results are demarcated by the range of coverage projected for a didactic robotic arm whose dimensions allowed to set the depth ranges of Table 2. They were also delimited by the RGB-D capture camera used in this case, a Blaster Senz3D creative whose 3D vision range is from 0.2-1.5 m. Both devices can be seen in Fig. 7.

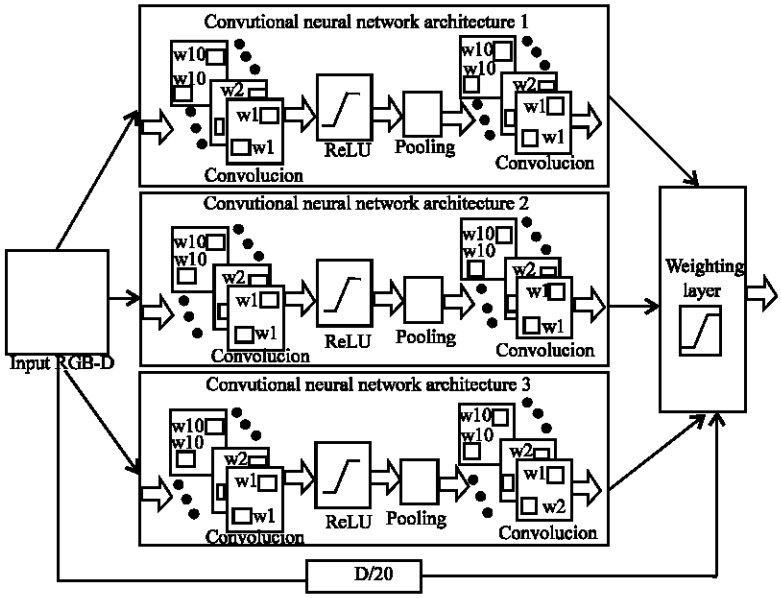

Fig. 5: Final architecture used

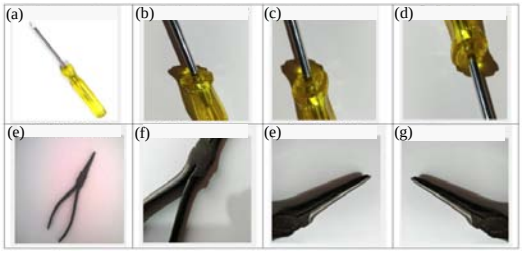

Fig. 6: Depth-based CNN response; Prediction: a) Screwdriver; b) Destornillador; c-d) Screwdriver and e-h) Nippers

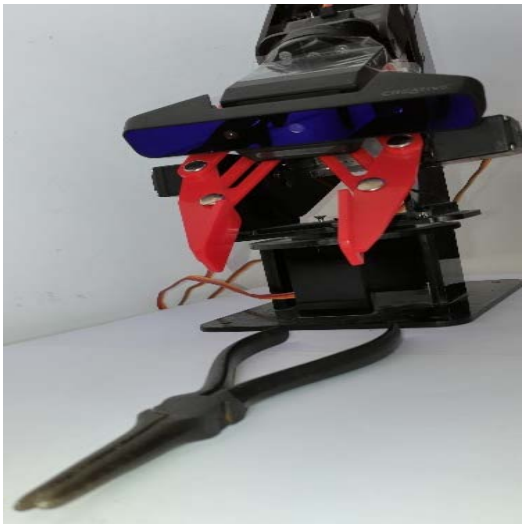

Fig. 7: RGB-D camera used and robotic arm

CONCLUSION

The two types of novel architectures proposed present a solution to the problem of identifying objects using dynamic systems that vary the distance of capture of the image. Although, the first proposed option employs robust convolutional neural networks, its performance is lower than that of the second option with distance-adjusted networks, due in part to the training scheme, since, the learning transfer delivers convolution filters initially more complex than the required for the identification of an object at a short distance.

The final architecture developed presents an overall functionality of a conventional convolutional neural network, however, adding a weighting output layer empowers the network for its application in cases of mobile robotics through machine vision systems, transparently to the user.

The decrease in the error rate in the identification of the tools from different perspectives allows to conclude the effectiveness of the established architecture, however, the final layer is susceptible of being modified by other layers that fulfill the same purpose as it can be a diffuse inference layer.

REFERENCES

Fadlullah, Z., F. Tang, B. Mao, N. Kato and O. Akashi *et al.*, 2017. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE. Commun. Surv. Tutorials, 2017: 1-1.

Hossen, J., S. Sayeed, T. Bhuvaneswari, C. Venkataseshaiah and J. Emerson *et al*., 2017. An automated fuzzy logic based low cost floor cleaning mobile robot. J. Eng. Appl. Sci., 12: 119-126.

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. Imagenet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems, Leen, T.K., G.D. Thomas and T. Volker (Eds.). MIT Press, Cambridge, Massachusetts, USA., ISBN:0-262-12241-3, pp: 1097-1105.

Mancini, M., S.R. Bulo, E. Ricci and B. Caputo, 2017. Learning deep NBNN representations for robust place categorization. IEEE. Rob. Autom. Lett., 2: 1794-1801.

Orozco, I., M.E. Buemi and J.J. Berlles, 2016. A study on pedestrian detection using a deep convolutional neural network. Proceedings of the International Conference on Pattern Recognition Systems (ICPRS-16), April 20-22, 2016, IET, Talca, Chile, ISBN:978-1-78561-283-1, pp: 1-15.

Porzi, L., S.R. Bulo, A. Penate-Sanchez, E. Ricci and F. Moreno-Noguer, 2017. Learning depth-aware deep representations for robotic perception. IEEE. Rob. Autom. Lett., 2: 468-475.

Qian, Y., M. Bi, T. Tan and K. Yu, 2016. Very deep convolutional neural networks for noise robust speech recognition. IEEE. ACM. Trans. Audio, Speech Lang. Process., 24: 2263-2276.

Redmon, J. and A. Angelova, 2015. Real-time grasp detection using convolutional neural networks. Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA'15), May 26-30 2015, IEEE, Seattle,Washington, ISBN:978-1-4799-6924-1, pp: 1316-1322.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks, 61: 85-117.

Wang, Z., Z. Li, B. Wang and H. Liu, 2016. Robot grasp detection using multimodal deep convolutional neural networks. Adv. Mech. Eng., Vol. 8,

Warczynski, J., 2000. Robot fine-motion control. IFAC. Proc. Volumes, 33: 43-48.

Yang, P.C., K. Sasaki, K. Suzuki, K. Kase and S. Sugano et al., 2017. Repeatable folding task by humanoid robot worker using deep learning. IEEE. Rob. Autom. Lett., 2: 397-403.

Zeiler, M.D. and R. Fergus, 2014a. Visualizing and Understanding Convolutional Networks. In: Computer Vision, Fleet, D., T. Pajdla, B. Schiele and T. Tuytelaars (Eds.). Springer, Cham, Switzerland, ISBN:978-3-319-10589-5, pp: 818.

Zeiler, M.D. and R. Fergus, 2014b. Visualizing and understanding convolutional networks. Proceedings of the European Conference on Computer Vision, September 6-12, 2014, Springer, Zurich, Switzerland, pp: 818-833.

Zhang, Q., L.T. Yang and Z. Chen, 2016. Deep computation model for unsupervised feature learning on big data. IEEE. Trans. Serv. Comput., 9: 161-171.