International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 1, February 2020, pp. 1053~1062
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i1.pp1053-1062

Assistant robot through deep learning

Robinson Jiménez-Moreno, Javier O. Pinzón-Arenas, César G. Pachón-Suescún
Faculty of Engineering, Nueva Granada Military University, Colombia

Article Info

Article history:

Received Jun 17, 2019
Revised Oct 11, 2019
Accepted Oct 20, 2019

Keywords:

3D environment
Convolutional neural network
Robotic applications

ABSTRACT

This article presents a work oriented to assistive robotics, where a scenario is established for a robot to reach a tool in the hand of a user when they have verbally requested it by his name. For this, three convolutional neural networks are trained, one for recognition of a group of tools, which obtained an accuracy of 98% identifying the tools established for the application, that are scalpel, screwdriver and scissors; one for speech recognition, trained with the names of the tools in Spanish language, where its validation accuracy reaches a 97.5% in the recognition of the words; and another for recognition of the user's hand, taking in consideration the classification of 2 gestures: Open and Closed hand, where a 96.25% accuracy was achieved. With those networks, tests in real-time are performed, presenting results in the delivery of each tool with a 100% of accuracy, i.e. the robot was able to identify correctly what the user requested, recognize correctly each tool and deliver the one need when the user opened their hand, taking an average time of 45 seconds in the execution of the application.

Corresponding Author:

Robinson Jiménez Moreno,
Mechatronics Engineering Program, Faculty of Engineering,
Nueva Granada Military University,
Carrera 11 #101-80, Bogotá D.C., Colombia.
Email: robinson.jimenez@unimilitar.edu.co

1. INTRODUCTION

In recent years, robotics has moved from the industrial environment to the domestic, dabbling with a wide variety of applications such as entertainment robots [1], pet sitters, floor [2] and pool cleaners [3], lawnmowers, and healthcare assistants [4], among many other applications. For this reason, where the increase in human-robot interaction is evident, it is necessary to develop techniques that provide security tools in said interaction [5]. Within the field of human-robot interaction, at an industrial level, developments have been presented to facilitate this task safely. For example, in [6], the development of a collaborative environment based on restrictions is presented so that a person can share the work area with a set of robotic arms, safely. These applications involve equipping the robot with a certain degree of artificial intelligence for various tasks, such as fine control (Fine motion) of the same [7].

Some of these developments employ non-linear optimization algorithms and machine vision systems to establish a secure interaction space, as discussed in [8]. Additionally, the systems that use artificial vision complement this task by neuronal learning, as illustrated in [9], where the general framework of an assistant robot based on artificial intelligence is set. Within these artificial intelligence techniques, the applications of pattern recognition through convolutional neural networks (CNN) are highlighted, which integrate both aspects, identification in images (vision) and pattern recognition [10].

CNNs have shown high performance in the recognition of objects [11] and today several architectures of this type of network are applied in machine vision applications [12]. Some examples of CNN application are framed in fields such as text classification, with web applications [13] or signature recognition [14], thus showing the versatility of these networks in pattern recognition. However, variations of CNN base architectures are still being developed in order to improve aspects of learning, such as the case of

variations in the depth at which the objects are, as shown in [15, 16]. In turn, many fields of application require virtual test environments in order to validate the algorithms to be used [17], where CNNs also facilitate this task for robotic systems, as set out in [18]. However, in the area of robotics, some works have just started to be developed with this technique. For example, in [19, 20] object gripping algorithms are developed by means of a robotic manipulator, performing recognition by CNN. And in the area of human-robot interaction of the few developments found, is the one presented in [21], where they describe the development of a social robot for interaction in a domestic environment. Another area of interest addressed in this article is the use of CNNs for speech recognition [22], demonstrating the versatility of this network for a wide range of applications.

Due to the great versatility and efficiency of the CNN, in the present paper, these are used in the operation of an assistant robot arm. Said arm is commanded by the training of three CNN networks, one for identification of a group of three tools, another for speech recognition and another for identification of the user's hand. The assistance system will be able to listen to what tool is desired, identify it within a group, take it and deliver it in the hand, if it is open, of a user. This article is divided into 4 sections. Section two shows the CNN architectures developed and implemented in this work. Section 3 shows the results obtained and the respective analysis. In section 4, the derived conclusions are presented.

2. CNN ARCHITECTURES

The proposed elements that make up the complete implementation of the assistant robot is presented in Figure 1. It consists of a user that verbally indicates the desired tool so that through a supervisory camera, which takes the top view of the scene, and with this, the tool and the user's hand are captured. The identification of the speech command, the tool, and the hand is done through the use of CNN. This type of network involves the calculation of particular architectures based on layers of the Convolution-ReLU-Pooling sets [8]. Here, the input volume for the initial layer corresponds to matrices that contain the learning patterns, for the case, images, and audio information, using three components for each one (depth). However, when the volume passes through the next layers, it varies its size, for that reason, the (1) to (3) illustrate how to calculate the volumes for the following layers.

$$W_{n+1} = \frac{W_n - F_n + 2P_n}{S_n} + 1 \quad (1)$$

$$H_{n+1} = \frac{H_n - F_n + 2P_n}{S_n} + 1 \quad (2)$$

$$D_{n+1} = K_n \quad (3)$$

In the equations, n and n+1 are the input and output volume, respectively, W is the width and H, the height of the input image of layer n, P is the zero padding, F is the size of the filter or kernel and S, the step of the filters. In (3), D represents how deep the layer is and K, the number of filters used in layer n [23].

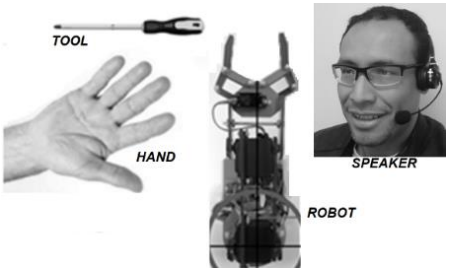

Figure 1. Elements that are part of the work

2.1. Implemented architectures

For the design of the CNN architectures, three databases were built according to the aforementioned elements. Next, the characteristics of each architecture implemented with its respective database are presented.

2.1.1 Architecture 1 (speech)

The database consists of 3 Spanish words ("bisturí" (scalpel), "destornillador" (screwdriver) and "tijeras" (scissors)), that are the tools to be used, from which 155 recordings per word were taken from different users, each audio is 2 seconds long. The database is divided into 2 sets, one for training with a total of 375 audios, and one for validation with 90 samples. Each audio is acquired with a sampling frequency of 16000 Hz. For this, extraction of features of each audio signal is done, in order to obtain a map that allows seeing the behavior of the signal through time at different frequencies. This extraction is carried out by means of an approximation of the MFCC obtained by the Hidden Markov Model Toolkit [24], using (4), where C is the number of cepstral coefficients, in this case, is 13; M is the number of filter channels, whose value is 20; L is the value responsible for incrementing the cepstral values, using a value of $L=22$, which is normally used to generate a good front-end parameterization in speech recognition systems; and m_j is the logarithm of the magnitude spectrum. A feature map as the one shown in Figure 2 is obtained, and, for robustness purposes in terms of the network learning, its first and second derivatives are added to the matrix, forming a three-channel input volume.

$$c_i = \left(1 + \frac{L}{2}\sin\left(\frac{\pi i}{L}\right)\right)\sqrt{\frac{2}{M}}\sum_{j=1}^{M} m_j \cos\left(\frac{\pi i}{M}(j-0.5)\right), \qquad i = 0, \dots, C \tag{4}$$

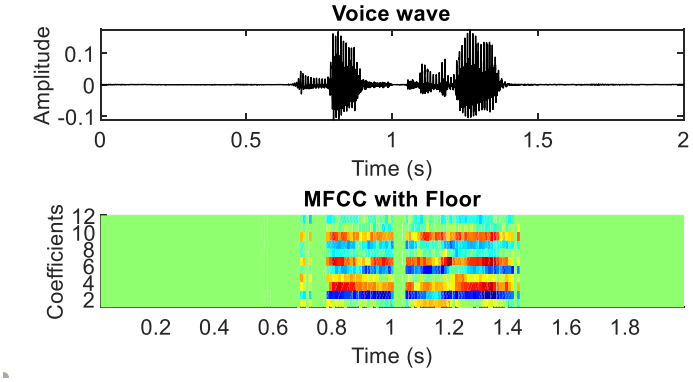

Figure 2. Example of an audio signal from the database and the MFCC obtained

Table 1 shows the architecture used, which is based on a previous development [25]. This consists of 3 sets of Convolution-Convolution-Pooling, with square filters in order to allow the network to learn the behavior of the words in terms of time and frequency at the same time. The second parameter of the kernel is given as S/P, where S is the stride of the movement of the filter and P is the zero-padding applied to the input volume. In addition, 2 fully connected layers are added, allowing a better learning of the combination of the different features. Since in MFCC, coefficient values have variation both negative and positive, the activation function ReLU is not used, in order to keep negative values. With the architecture and the database built, the training is performed for 500 epochs, achieving a 97.5% accuracy with the validation set.

Table 1. CNN Architecture for speech

Layer	Kernel		Filters
Input	12x199x3		-
Convolution	5x5	1/2	32
Convolution	5x5	1/2	32
MaxPooling	2x1	2/0	-
Convolution	3x3	1/1	64
Convolution	3x3	1/1	64
MaxPooling	2x3	2/0	-
Convolution	2x2	1/1	128
Convolution	3x3	1/1	128
MaxPooling	2x2	2/0	-
Fully-Connected	1		512
Fully-Connected	1		2048
Fully-Connected	1		3
Softmax	3		-

2.1.2. Architecture 2 (hand)

For the recognition of the hand, there were established 2 categories: Open and Closed, since it is wanted to know these two gestures. With this, a dataset was built, consisting of 1900 images, which was divided to have a set for training and another for validated the network performance. For training, 700 images of the open hand and 700 of the closed hand were taken, and for validation, 250 images of each category were used. The architecture proposed is based on previous work, shown in [23] and that can be seen in Table 2. It should be highlighted that each convolution and fully connected have a ReLU layer. This architecture, after being trained, achieved an accuracy of 96.25% in the classification of both gestures.

Table 2. CNN Architecture for hand gesture

Layer	Kernel		Filters
Input	64x64x3		-
Convolution	4x4	1/2	20
Convolution	4x4	1/0	20
MaxPooling	2x2	2/1	-
Convolution	5x5	1/0	50
Convolution	5x5	1/0	50
MaxPooling	2x2	2/0	-
Convolution	4x4	1/0	200
MaxPooling	3x3	2/0	-
Fully-Connected	1		200
Fully-Connected	1		2
Softmax	2		-

To improve the recognition of the gestures, a preprocessing algorithm was applied, where by means of acquisition of the mean of the background and morphologic filters, the location of the hand is obtained, in such a way that the algorithm generates a bounding box of the new object in the scene. Then, it is cropped and, depending on the greater value of the box side, the cropped image is pasted in a white square, to the being entered to the network for its recognition. The final result of the recognition is shown in Figure 3, where the yellow box is the first bounding box found. Additionally, the number given in the box is the confidence of the network with respect to the gesture recognized, where the high efficiency of the success is evident.

Figure 3. Hand recognition

2.1.3. Architecture 3 (tools)

The third architecture is responsible for identifying the three tools to be used. For the network training, it is built a database that is composed of 700 images. Due to the fact that the tools have more features to be learned and the need of differentiating some of the similarities that the tools have, it was decided to use a bigger image size, that for this case, it was 128 x 128 pixels in RGB scale. To allow the network to learn global features of the tools, such as edges and shapes, the filters are proposed with sizes bigger than the normally used, as the previous network. In this case, filters that vary their sizes from 7x7 to 12x12 are used. In addition, each convolution layer has an activation function of ReLU. The architecture proposed is shown in Table 3. Figure 4 illustrates the result of the network and the main activations obtained, these allow to demonstrate the clear learning of each tool, where an accuracy of 98% of accuracy in the classification of the three classes was obtained after its training with the database built

Table 3. CNN architecture for tool

Layer	Kernel		Filters
Input	128x128x3		-
Convolution	12x12	1/0	20
Convolution	11x11	1/0	40
MaxPooling	3x3	2/0	-
Convolution	10x10	1/0	80
Convolution	9x9	1/0	160
MaxPooling	2x2	2/0	-
Convolution	8x8	1/0	180
Convolution	7x7	1/0	200
Fully-Connected	1		200
Fully-Connected	1		3
Softmax	3		-

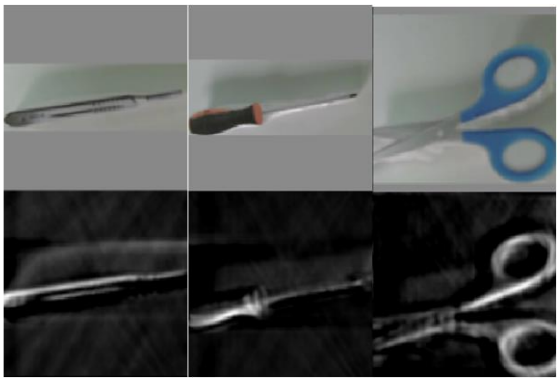

Figure 4. Recognition of tools

2.2. Kinematic model of the robot

The kinematic model of the arm is required to be able to establish the displacement of this within its workspace, in which both the tools and the user's hand are located. In Figure 5, it can be observed the geometric model that allows inferring (5) to (14), through which it can be set the angular movements of the robot. From the top view, the angle of joint 1 (θ_1) and the X component of point P of the final effector are observed, through which (5) and (6) are established.

$$x' = \sqrt{Pz^2 + Px^2} \tag{5}$$

$$\theta_1 = tan^{-1}\frac{Pz}{Px} \tag{6}$$

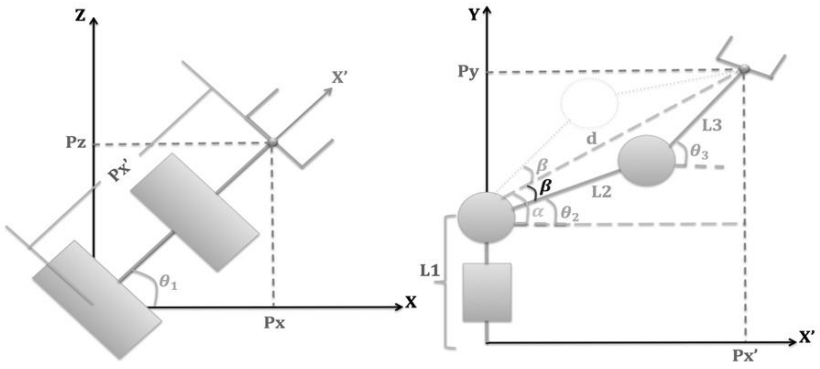

Figure 5. Kinematics of the robotic arm

By means of the frontal view, the angles of joints 2 and 3 are obtained, calculating the length "d", and the angle α through (7) to (10), thus determining the angle θ_3 of joint 3.

$$d = \sqrt{(Py - L1)^2 + Px'^2} \tag{7}$$

$$d^2 = L_2^2 + L_3^2 + 2L_2L_3cos\theta_3 \tag{8}$$

$$sin\theta_3 = \pm\sqrt{1 - cos^2\theta_3} \tag{9}$$

$$\theta_3 = tan^{-1}\left(\frac{\pm\sqrt{4L_2^2L_3^2 - d^2 - L_2^2 - L_3^2}}{d^2 - L_2^2 - L_3^2}\right) \tag{10}$$

The θ_2 angle of joint 2 is calculated using β and α using (11) and (13). Since the angle of the second joint can have two different values, depending on the grip of the manipulator using elbow up or elbow down, the (14) is used.

$$\alpha = tan^{-1}\frac{P_y - L_1}{Px'} \tag{11}$$

$$L_3^2 = L_2^2 + d^2 - 2L_2dcos\beta \tag{12}$$

$$\beta = cos^{-1}\frac{L_2^2 + d^2 - L_3^2}{2L_2d} \tag{13}$$

$$\theta_2 = \begin{cases}\alpha - |\beta| \\ \alpha + |\beta|\end{cases} \tag{14}$$

With these parameters, it is possible to control the movement of the robot, where, once obtained the positions of the geometric center of the tool and the hand, theta angles are set in order to reach those positions, taking into account that the reference is the robot base.

3. EXPERIMENTAL RESULTS

The application starts by saying the desired tool, i.e. the user starts the recording, where their voice is captured. After 2 seconds, the voice signal is sent to the processing algorithm to then be entered to the CNN for speech. Once the network recognizes the word, the user decides if it was correctly recognized or not. If not, it retakes the audio. If so, the algorithm proceeds to identify the tool requested, taking a capture of the tools and then send it to the CNN for tools. When the one wanted is identified and located, the robot takes it. In order to deliver the object, the user put his hand in front of the camera, in order to capture it and recognize if it is open or closed. When it identifies the open hand for 2 seconds, the robot delivers the object to the position of the hand, gives the tool, and then returns to its initial position. When the user closes their hand, the algorithm ends a cycle and waits for the user to starts another recording. The flowchart of this algorithm can be seen in Figure 6.

Figure 7 illustrates the first step of the process, where the screwdriver tool is chosen as an option and the recognition of it successfully is validated. From the capture of the tools, the recognition and position of the screwdriver are validated, as shown in Figure 8. Figure 9 illustrates the process of moving the robotic arm to the location of the desired tool, where it is grasped by the end effector to be delivered to the user's hand, if it is open, as shown in Figure 10. The process of the flow chart of Figure 6 is repeated with each of the three tools, successfully achieving the three basic processes: speech, tool and hand gesture recognition, and managing to deliver each tool. The average execution time of the whole process is 45 seconds. Figure 11 shows the grip and transfer of the tool scissors. The tests were carried out with polystyrene tools, in such a way that they emulated the real dimensions, due to the use of an academic robotic arm with no load capacity and sufficient grip for manipulation of the real tools.

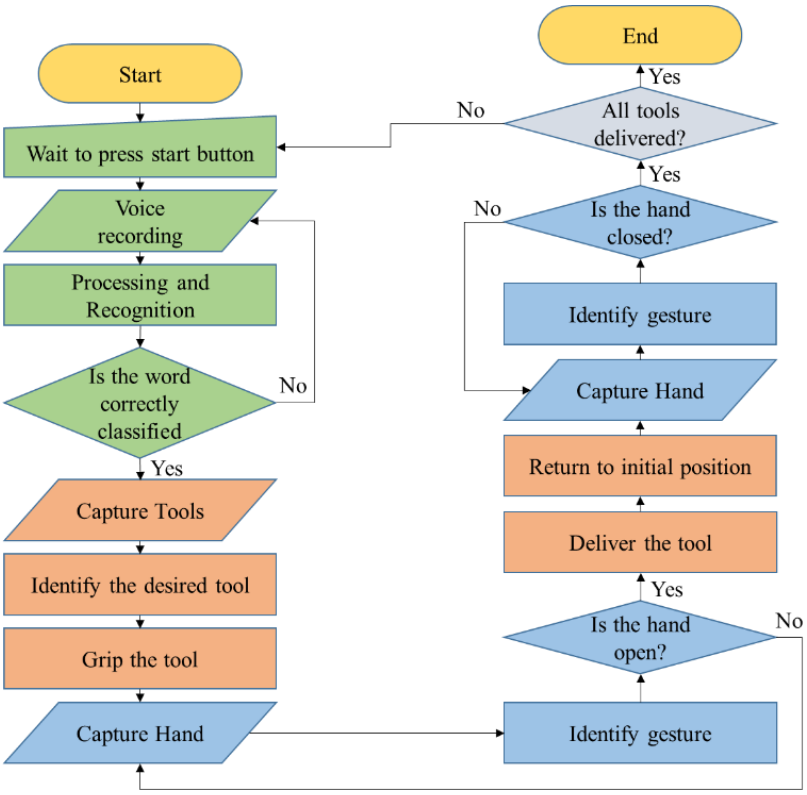

Figure 6. Algorithm flowchart

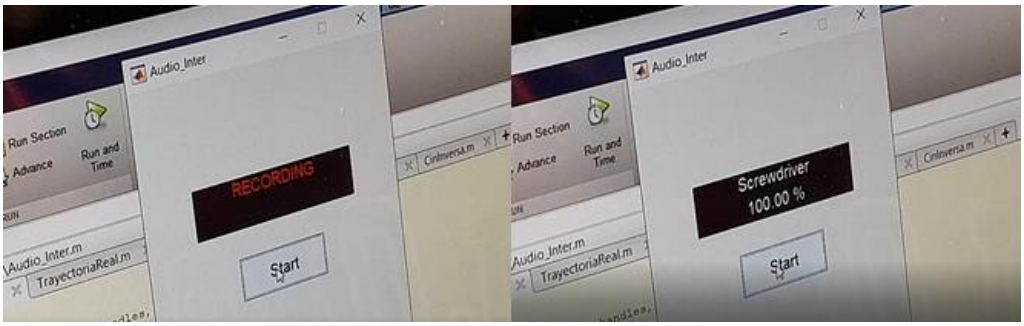

Figure 7. Speech recognition test

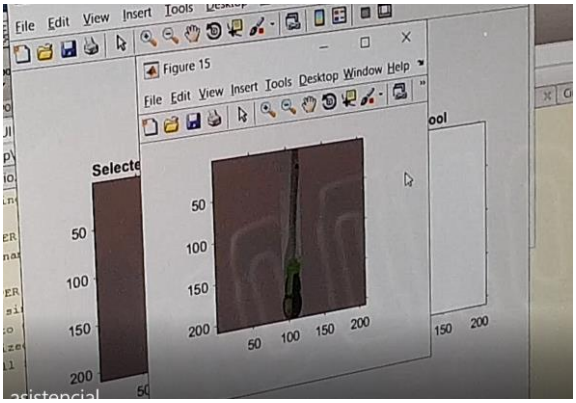

Figure 8. Recognition of the tool

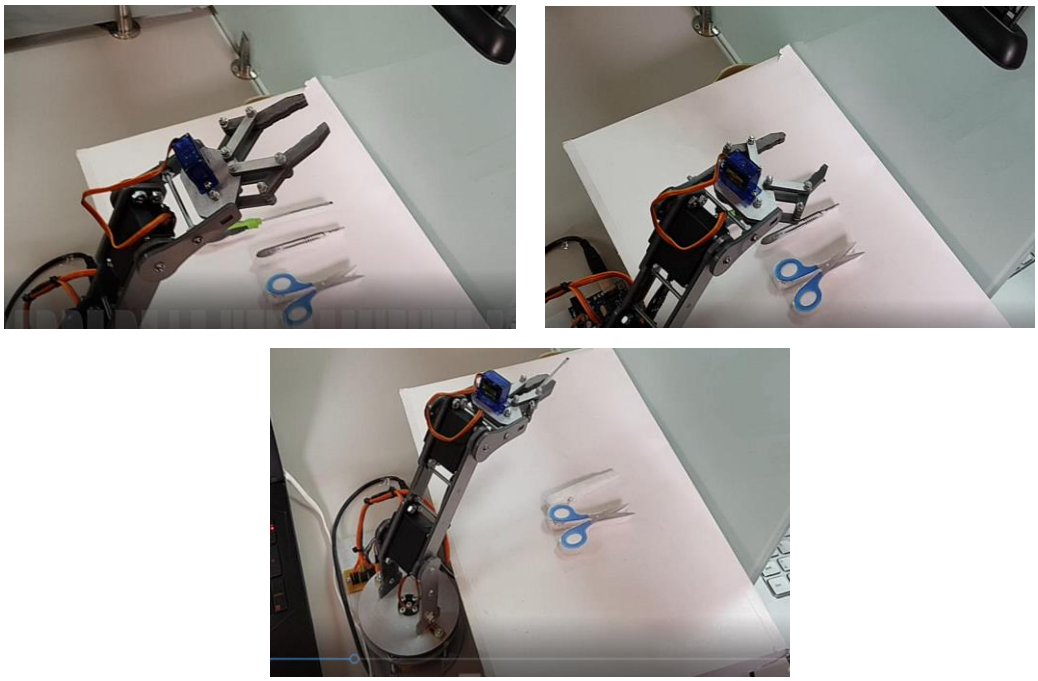

Figure 9. Location and grip

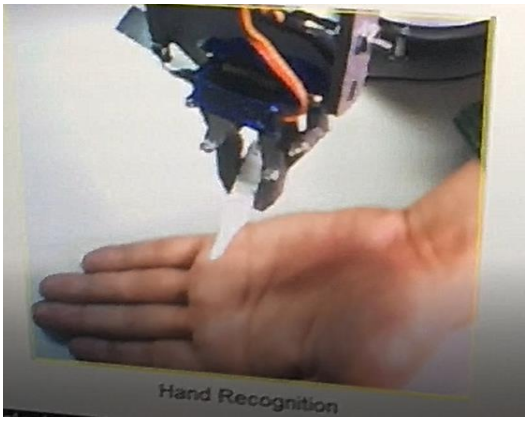

Figure 10. Delivery of the tool

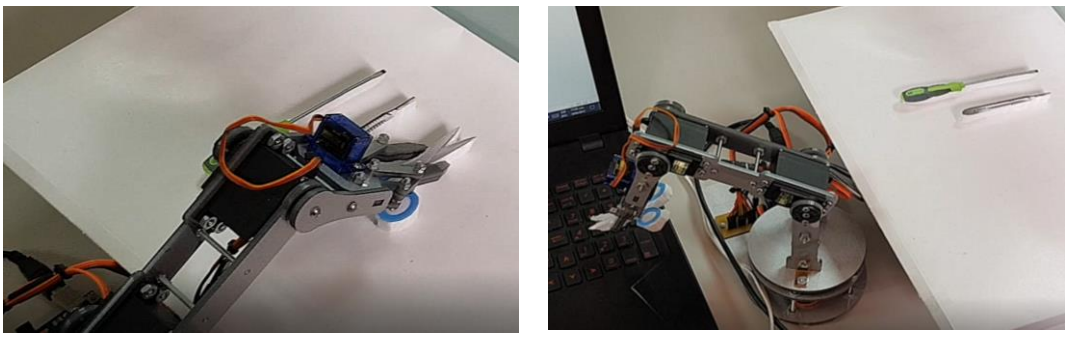

Figure 11. Validation with scissors

4. CONCLUSION

It was possible to obtain a functional application of assistive robotics, which is capable of receiving and correctly identifying a voice command, associating it with a physical object, taking it and delivering it to the hand of a user. In addition, it was evidenced by the versatility of CNN networks for assistive robotics applications, which require pattern recognition algorithms. While the times obtained may seem high, it is noteworthy that a computer was used that does not operate the algorithm in real-time, so that its execution in a dedicated equipment would be reduced. The developed system manages to successfully deliver the trained tools, however, increasing the set of these to more categories, requires redesigning the convolutional networks, which can involve architectures with greater depth and consequently take more time to execute the assistive task.

ACKNOWLEDGEMENTS

The authors are grateful to the Nueva Granada Military University for the support given in the development of this work.

REFERENCES

[1] E. Jochum, P. Millar, and D. Nuñez, "Sequence and chance: Design and control methods for entertainment robots," *Robotics and Autonomous Systems*, vol. 87, pp. 372-380, 2017. DOI: 10.1016/j.robot.2016.08.019
[2] N. K. Sahu, N.K. Sharma, M.R. Khan, and D.K. Gautam, "Comparative Study on Floor Cleaner," *Journal of Pure Applied and Industrial Physics*, vol. 8(12), pp. 233-236, 2018.
[3] V.R. Batista and F.A. Zampirolli, "Optimising Robotic Pool-Cleaning with a Genetic Algorithm," *Journal of Intelligent & Robotic Systems*, vol. 95(2), pp. 443-458, 2019. DOI: 10.1007/s10846-018-0953-y
[4] R.M. Agrigoroaie, and A. Tapus, "Developing a healthcare robot with personalized behaviors and social skills for the elderly," *In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE*, pp. 589-590, 2016. DOI: 10.1109/HRI.2016.7451870
[5] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43-52, 2017. DOI: 10.1016/j.robot.2017.04.004
[6] J. De Gea Fernández, *et al*., "Multimodal sensor-based whole-body control for human–robot collaboration in industrial settings," *Robotics and Autonomous Systems*, vol. 94, pp. 102-119, 2017.
[7] J. Warczyński, "Robot Fine-Motion Control," *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 43-48, 2000.
[8] R.J. Moreno, M. Mauledoux, and O.F. Avilés, "Path Optimization Planning for Human-Robot Interaction," *International Journal of Applied Engineering Research*, vol. 11, no. 22, pp. 10822-10827, 2016.
[9] C. Bousquet-Jette, *et al*., "Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 33-44, 2017. DOI: 10.1016/j.engappai.2017.04.015
[10] M.D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *In European conference on computer vision, Springer, Cham, 2014*, pp. 818-833, Sep 2014. DOI: 10.1007/978-3-319-10590-1_53
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *In Advances in neural information processing systems*, pp. 1097-1105, 2012.
[12] F. Tu, *et al.*, "Deep Convolutional Neural Network Architecture with Reconfigurable Computation Patterns," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25(8), pp. 2220-2233, 2017. DOI: 10.1109/TVLSI.2017.2688340
[13] S. Aich, S. Chakraborty, and H.C. Kim, "Convolutional neural network-based model for web-based text classification," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(6), pp. 5185-5191, 2019. DOI: 10.11591/ijece.v9i6.pp5185-5191
[14] J.O. Pinzón-Arenas, R. Jiménez-Moreno, and C.G. Pachón-Suescún, "Offline signature verification using DAG-CNN," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(4), pp. 3314-3322, 2019. DOI: 10.11591/ijece.v9i4.pp3314-3322
[15] R. Jimenez-Moreno and D. Ovalle Matrinez, "A Novel Parallel Convolutional Network Architecture for Depth-Dependent Object Recognition," *International Review of Automatic Control*, vol. 12(2), pp. 76-81, 2019. DOI: 10.15866/ireaco.v12i2.16467
[16] M. S. H. Al-Tamimi, "Combining convolutional neural networks and slantlet transform for an effective image retrieval scheme," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(5), pp. 4382-4395, 2019. DOI: 10.11591/ijece.v9i5.pp4382-4395
[17] A. Vorotova, V. Finaev, V. Soloviev, and M. Medvedev, "Statistical Data Processing of Two Mobile Objects Behavior in Random Environments Using Simulation Modeling Method," *International Review of Automatic Control*, vol. 12(4), 2019.
[18] J. O. P. Arenas, R. Jiménez, and P. C. U. Murillo, "Faster R-CNN for object location in a Virtual Environment for sorting task," *International Journal of Online Engineering (iJOE)*, vol. 14(07), pp. 4-14, 2018.
[19] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8(9), pp. 1-12, 2016. DOI: 10.1177/1687814016668077

[20] D. Kalashnikov, *et al*., "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.
[21] M.A. Gutiérrez, L.J. Manso, H. Pandya and P. Núñez, "A Passive Learning Sensor Architecture for Multimodal Image Labeling: An Application for Social Robots*," Sensors (Basel)*, vol. 17(2), p. 353, 2017.
[22] Y. Qian, and P.C. Woodland, "Very deep convolutional neural networks for robust speech recognition," *In 2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 481-488, 2016. DOI: 10.1109/SLT.2016.7846307
[23] J.O.P. Arenas, R.D.H. Beleño, and R.J. Moreno, "Deep Convolutional Neural Network for Hand Gesture Recognition Used for Human-Robot Interaction," *Journal of Engineering and Applied Sciences*, vol. 12(11), pp. 9278-9285, 2017.
[24] S. Young, *et al*., *The HTK book*, Cambridge University Engineering Department, 2002.
[25] J.O. Pinzón Arenas, R. Jiménez Moreno, R.D. Hernández Beleño, "Word-based Deep Convolutional Neural Network architecture for Speech Recognition," *In II Congreso Internacional de Ciencias Básicas e Ingeniería*, pp. 1-10, 2018.

BIOGRAPHIES OF AUTHORS

Robinson Jiménez Moreno was born in Bogotá, Colombia, in 1978. He received the Engineer degree in Electronics at the Francisco José de Caldas District University - UD - in 2002. M.Sc. in Industrial Automation from the Universidad Nacional de Colombia - 2012 and Ph.D. in Engineering at the Francisco José de Caldas District University - 2018. He is currently working as a Professor in the Mechatronics Engineering Program at the Nueva Granada Military University - UMNG. He has experience in the areas of Instrumentation and Electronic Control, acting mainly in Robotics, control, pattern recognition, and image processing.
E-mail: robinson.jimenez@unimilitar.edu.co

Javier Orlando Pinzón Arenas was born in Socorro-Santander, Colombia, in 1990. He received his degree in Mechatronics Engineering (Cum Laude) in 2013, Specialization in Engineering Project Management in 2016, and M.Sc. in Mechatronics Engineering in 2019, at the Nueva Granada Military University - UMNG. He has experience in the areas of automation, electronic control, and machine learning. Currently, he is studying a Ph.D. in Applied Sciences and working as a Graduate Assistant at the UMNG with emphasis on Robotics and Machine Learning.
E-mail: u3900231@unimilitar.edu.co

César Giovany Pachón Suescún was born in Bogotá, Colombia, in 1996. He received his degree in Mechatronics Engineering from the Pilot University of Colombia in 2018. Currently, he is studying his Master's degree in Mechatronics Engineering and working as a Research Assistant at the Nueva Granada Military University with an emphasis on Robotics and Machine Learning.
E-mail: u3900259@unimilitar.edu.co