International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 4
ISSN: 2088-5334

Visual Commands for Control of Food Assistance Robot

Javier O. Pinzón-Arenas[a,*], Robinson Jiménez-Moreno[a,*]

[a] *Department of Mechatronics Engineering, Militar Nueva Granada University, Bogotá D.C, 110111, Colombia*
*Corresponding author: *u3900231@unimilitar.edu.co, *robinson.jimenez@unimilitar.edu.co*

***Abstract*— Assistance robots improve people's quality of life in residential and office tasks, especially for people with physical limitations. In the case of the elderly or people with upper limb motor disabilities, an assistance robot for food support is necessary. This development is based on a mixed environment, a real and virtual environment working interactively. A camera located in front of the user is used, at a distance of 60 cm, so that it has an excellent visual range to capture the user's hand gestures for the commands. Pattern recognition based on a deep learning algorithm is made with convolutional neural networks to identify the user's hand gestures. This work exposes the network's training and the results of the robot command's execution. A virtual environment is presented in which a robotic arm with a spoon-like effector is used in a machine vision system that allows eight different types of commands to be recognized for the robot by training a faster R-CNN network for which a database of 640 images is used, achieving a degree of system performance of over 95%. The average time in the execution of a cycle from detecting and identify the command gesture to move the robot towards the food and return in front of the user is 21 seconds, making the development useful for real-time applications.**

***Keywords*— Convolutional neural network; faster R-CNN; assistance robot; virtual environment.**

Manuscript received 27 Jul. 2020; revised 12 Dec. 2020; accepted 15 Mar. 2021. Date of publication 31 Aug. 2021.

I. INTRODUCTION

Advances in different engineering disciplines allow improving the quality of life of people in various aspects, particularly the development of robotics offers extensive possibilities for this purpose. Many investigations are oriented to the development of robots to support human activities as an extension of the industrial environment in which they have been conceived. In Konijn and Hoornab [1], the interaction capabilities of a robotic agent as a primary mathematics tutor oriented to children are explored, where the anthropomorphic robot has image acquisition capabilities, and basic social skills are programmed for interaction with the child. One of the aspects of enhancement in the use of robots is currently healthcare applications. In medicine, there are several cases in which they are used as a surgical assistant [2]–[4], where an important aspect is to foresee the security conditions for people in the use of robots as assistants in interaction tasks, as established in Yamada and Akiyama [5].

To improve people's quality of life, robots have been used as support in the care of non-self-sufficient people. In Lanza *et al* [6], a robotic system oriented to be a human caregiver is presented to monitor patients and help them with their basic needs. For example, one of the basic needs of the human being is to feed; a person with motor deficiencies in the upper limbs or muscular problems of silverware manipulation would find a solution in an assistance robot, which motivated the development of this work.

To improve the autonomy of a robotic agent, the algorithms with which they are programmed include artificial intelligence techniques [7], where even hybrid systems that involve more than one technique are used, for example, neural networks and fuzzy logic, as explained in Baker and Ghadi [8], to move a mobile robotic agent. However, recently, deep learning stands out within artificial intelligence techniques [9], which has shown high robustness in pattern recognition tasks. They have been integrated into the development of tasks that involve the robot and its interaction with humans [10].

Jiménez-Moreno *et al.* [11] exposed an assistance robot able to deliver tools and bring it into user's in hand, where the Deep Learning technique used corresponds to convolutional neural networks (CNN), used here for tool recognition, voice recognition of the person concerning the name of the tool and recognize the user's hand so that the robot leaves the tool in his open hand. Convolutional networks specialize in image pattern recognition [12], and their use in robotic applications is quite extensive. In Useche *et al.* [13], these are used to allow a robot to grasp objects even with occlusions using a robotic arm.

To develop a robotic food assistance agent for people with disabilities, the command of it is proposed using hand signals to minimize the user's movements. Since the focus of the robot is for food, voice commands are not the best option for this task. Many techniques have been used for hand signal recognition, to mention a few, where convolutional networks also include capsule networks [14], LSTs [15], and geometric characteristics by Fisher vector [16]. However, again convolutional networks stand out for their robustness in this task [17]–[19].

Therefore, an assistance robot has been developed and commanded by the signs of one of the user's hands, allowing the generation of robotic actions to feed people. The signs are recognized using CNN. A simulation environment is used to validate the algorithm and to prevent accidents with the user, a fairly common way for robot training [20], [21], as well as the adequacy of the robotic end-effector [22], which in this case is designed spoon type. This work helps to expand the state of the art in the development of assistance robots to improve the quality of life and use deep learning techniques for their training.

This document is organized under the following structure: the introduction in the first section exposes the state-of-the-art developments in the subject. The second section corresponds to the methodology where the virtual environment and neuro-convolutional training are presented. The third section presents the analysis and results of the implemented algorithms, and the final section exposes the conclusions.

II. Materials and Methods

This section exposes the materials and methods exposed in three subsections: virtual environment, dataset, and neural network command for detection.

A. Virtual Environment

For the implementation of the robotic command identification system using hand gestures, a virtual environment is built, within which the robotic agent will perform the respective commands indicated by the user. A 4R robot with a spoon as end effector is used in the environment, responsible for collecting the food on the plates. Also, a camera located in front of the user is used, at a distance of 60 cm, so that it has sufficient visual range to capture the user's hand gestures for the commands. Around the robot, four plates are placed to let the user select which plate they want to eat from. The environment can be seen in Fig. 1.

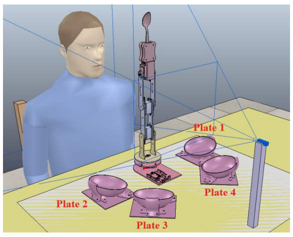

Fig. 1 Implemented virtual environment.

B. Database

It is proposed to recognize eight different types of robotic commands, divided into two categories. The first category refers to the direct control commands of the robot, consisting of four classes: Stop, Start, Pause, and Change. The "Stop" command ends the entire feeding process and can only be used when the robot brings the food to the user. The "Start" command is used to begin feeding the person and resume the execution of the task when it has been paused or when there is a change in the food to be received. The "Pause" command temporarily suspends the task and can be executed at any time within the entire process. The "Change" command allows the user to select the plate from which to receive the food and can be selected when the robot is in the food delivery position. In order to generate the recognition of the commands, it is decided for them to be detected within the image, so the location and labeling of these were done manually. In total, the database contains 480 images for training and 160 for validation. An example of each command in this category can be seen in Fig. 2.

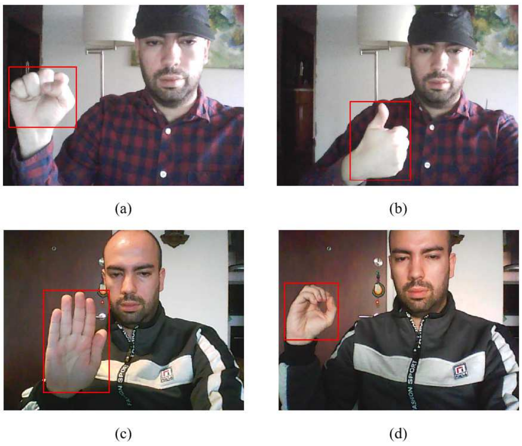

Fig. 2 Direct control commands. (a) Stop. (b) Start. (c) Pause. (d) Change.

The second category consists of the commands for the selection of the plate to which it is wanted the robot to go, containing gestures indicating plates 1 to 4. Like the previous one, the location of each command is done by means of a bounding box with its respective label. This database contains 480 images for training and 160 for validation. Of this, an example of each command can be seen in Fig. 3.

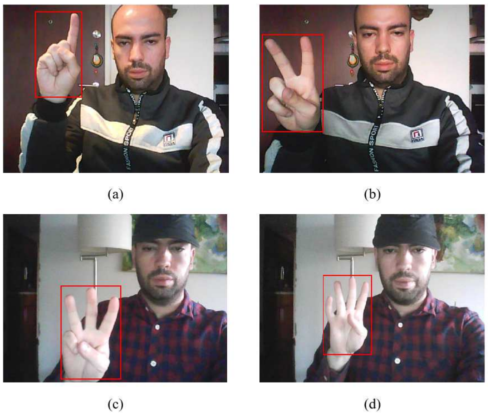

Fig. 3 Plate change commands. (a) Plate 1. (b) Plate 2. (c) Plate 3. (d) Plate 4.

C. Neural Network for Command Detection

To carry out the detection of the commands, it is proposed to use the Faster R-CNN [23], which has high precision in detecting objects and is also fast in real-time location. For structuring purposes of the Faster R-CNN architecture, a pre-trained model of the ResNet-50 [24] is used, demonstrating a high capacity for object classification. This allows the use of transfer learning to take advantage of the different characteristics and patterns learned by the pre-trained model to locate the hand gestures and perform the training only of the region proposal network (RPN) and the classification section of the network.

Since the two categories of commands will be used at different moments in the process, it is chosen to use two identical networks, one for the control commands and one for the plate change commands. This is done in order to have reduced categories in the recognition and to improve the learning of each network, by not having to differentiate possible similar categories, such as Pause and Plate 4, and using four possible predefined anchor boxes for each of the networks, depending on the size and shape of these for each command.

With this in mind, the training of the two neural networks is performed. Based on Ren *et al.* [23], the training of each neural network is done in four stages (training of the classification section, the detection section, and the fine-tuning of each one), as is shown in Table 1. A learning rate of 1x10-3 is used in the first two stages, while in the last two, being a fine-tuning of the previously trained weights, a lower rate of 5x10-4 is used. Since transfer learning is being used, the number of training epochs remains between 8 and 10, as only a small section of the network needs to be trained, so learning is much faster.

TABLE I
TRAINING PARAMETERS OF THE FASTER R-CNN.

	Learning Rate	***Epochs***
Stage 1	$1x10^{-3}$	10
Stage 2	$1x10^{-3}$	10
Stage 3	$5x10^{-4}$	8
Stage 4	$5x10^{-4}$	8

With these parameters, the training is done, obtaining a behavior in each stage for each network, as shown in Fig.s 4 to 6. Fig. 4 shows the result of the loss function for each iteration, in other words, how much it cost for the network to make an erroneous detection. Here, it can be seen that its cost was low, even less than 0.1 for the last stage, so it is possible to observe that the network was kept learning. This can also be perceived in Fig.s 5 and 6, where the accuracy remained above 95% and the root mean square error (RMSE) was below 0.15 in the last stage, i.e., the offset of the location of the network-generated bounding box from the manually labeled one was less than 15%.

After training, the networks are validated with the test sets by means of the precision vs. recall curve, obtaining the results shown in Fig. 7. The capacity of the trained networks shows a high degree of accuracy in locating and classifying each of the commands, with results above 97.5%, wherein some cases, the network, although it located a command, classified it erroneously, as occurred in the commands for plate 1 (P1) and plate 3 (P3).

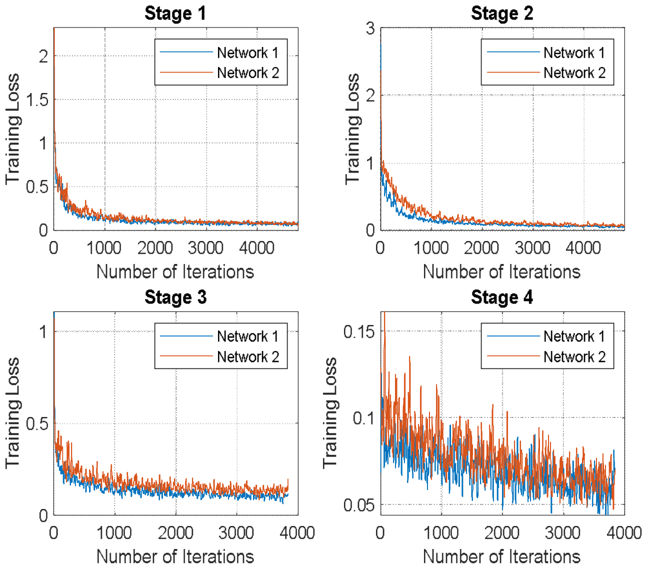

Fig. 4 Losses at every stage of training.

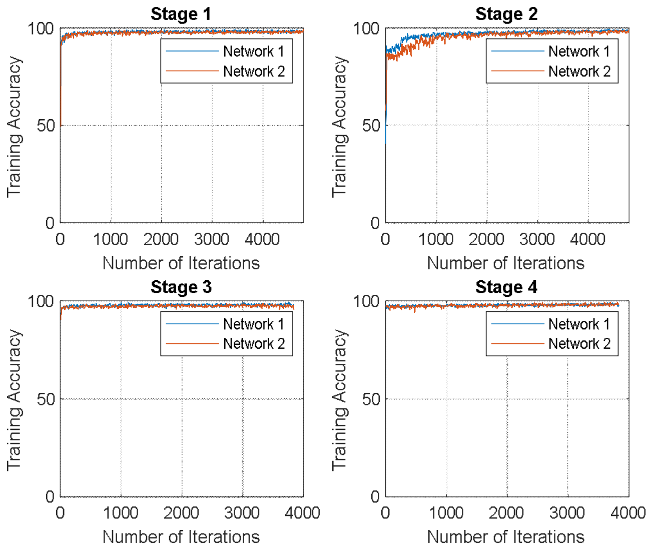

Fig. 5 Losses at every stage of training.

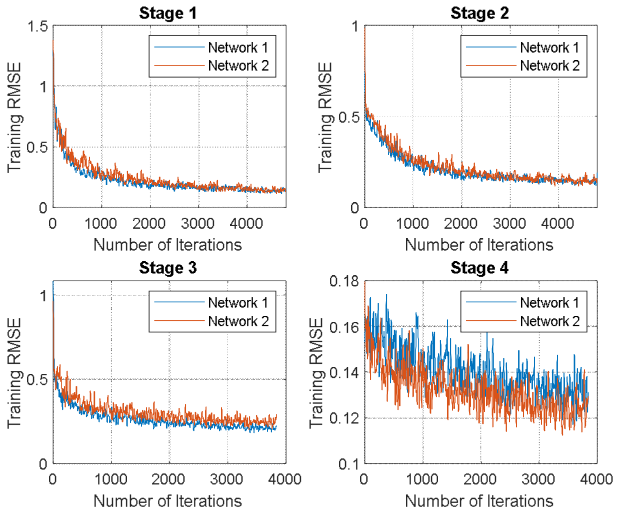

Fig. 6 Root mean square error at each stage of training.

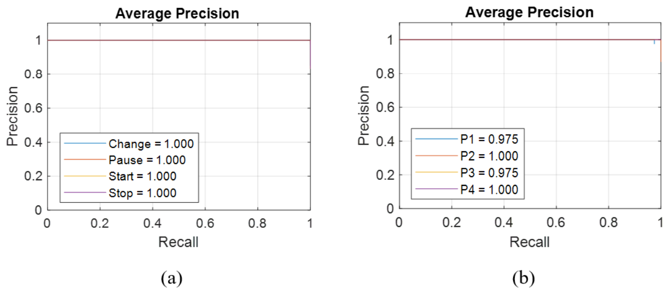

Fig. 7 Average accuracy in detecting commands for (a) Network 1 (control) and (b) Network 2 (selection).

With these validation results, it is possible to demonstrate that, despite using a reduced database for both cases, applying transfer learning allowed a high degree of performance to be achieved. This can also be seen in Fig. 8, where examples of detection of the two networks are shown. In this one, the networks react to drastic light changes in the environment, even almost mimicking the color of the hand with the wall and their capability to detect the gestures (yellow box) very close to the proposed ground truth (red box).

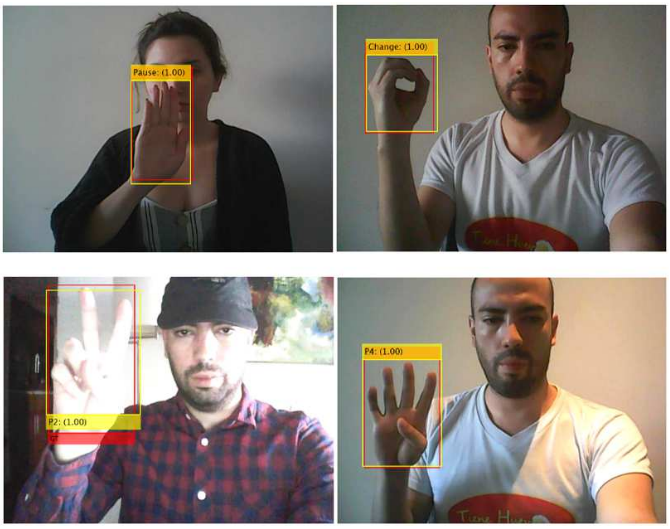

Fig. 8 Detection of the networks trained in the test dataset.

III. Results and Discussions

The trained networks are integrated through a robot control system to perform the interaction between the user and the virtual environment. The operation of the system starts with the location of the robot in a waiting position. Then, the recognition of the robot control commands begins. If the user makes the "Stop" gesture, the system will immediately terminate the feeding task. On the other hand, if the "Start" command is recognized, and it is the first time that the system is going to perform the task, it is proceeded to verify from which plate the user wants to receive the food. Once the plate selection command is recognized, the robot starts to execute the task, collecting the food, and taking it to a reception (or waiting) position. If the user makes the "Pause" gesture during the pickup process, the robot will stop until the user does the "Start" command again. When the robot reaches the waiting position, it checks if the user wants to continue receiving food from the same plate (Start command), change the plate (Change command), or stop feeding (Stop command). The system flowchart is shown in Fig. 9.

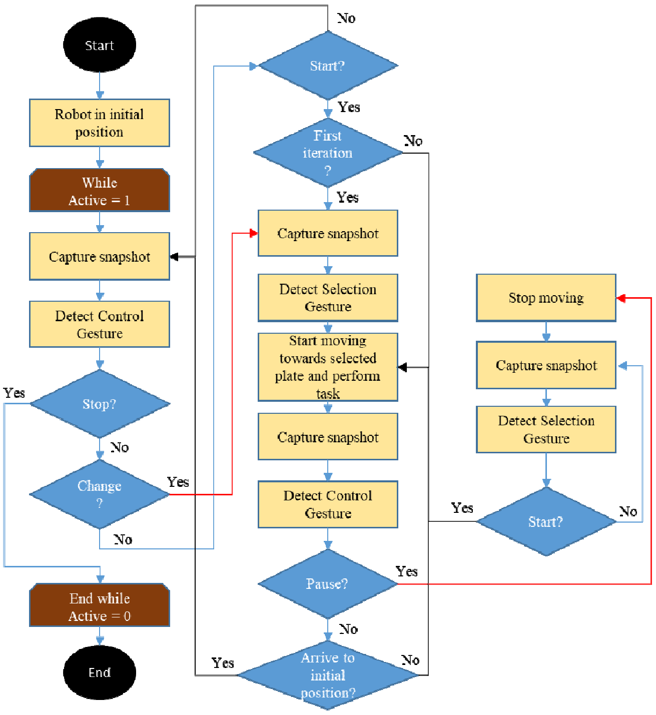

Fig. 9 Flowchart of the implemented system.

Five tests of the system are carried out in real-time, repeating the cycle of task execution four times, i.e., from the moment it indicates the beginning until it reaches the waiting position again, changing plates between cycles. This is done to observe the response time of the detection of the commands and verify the correct functioning of the implemented algorithm. The results of these tests can be viewed in Table 2 as follows:

- The percentage of control command detection (that it has detected the command corresponding to the one done by the user correctly).
- The percentage of selection command detection.
- The percentage of correctly completed cycles, i.e., that did not require rebooting the system, because it was not able to detect any of the commands within the conditional statements.
- The average execution times in detecting the commands and of the cycle without interruption (without using the "Pause" command, which was done in two cycles per test).

TABLE II
System Performance Tests.

	Test 1	Test 2	Test 3	Test 4	Test 5	Total
Detection of control commands	100%	100%	100%	100%	100%	100%
Detection of selection commands	100%	75%	100%	100%	100%	95%
Completed cycles	100%	100%	100%	100%	100%	100%
Average detection time	253 ms	269 ms	244 ms	245 ms	259 ms	254 ms
Average cycle time	21 s	23 s	21 s	20 s	20 s	21 s

Within the tests, the system correctly detected all the control commands indicated to it, even those pausing during the robot's movement. However, in the selection commands, in one of the cycles during test 2, the system recognized one of the gestures incorrectly. Nevertheless, the average number of correct detections was 95%. As for the execution of the cycles, they all ended correctly, without the need to reboot the system.

During the whole process, the command detection took an average time of 254 ms, which is fast enough for the correct execution of the task and control of the robot in real-time, especially when performing the "Pause" command, which must be read during the robot's movement. As for the execution time of the complete feeding task without interruption, it was 21 seconds. These tests were performed with an Nvidia GTX 1070 GPU.

In Fig. 10, it is possible to observe the system process, where the user indicates the robot to start with the feeding task. Then, when it is the first time it starts, the system asks the user to indicate the plate from which they want to receive the food, pointing to plate 2. In the upper part of the user's image, the control command selected and the plate where the robot will pick up the food are displayed, which in this case is plate 2. The robot performs the collection, and when it returns to the waiting position, the user asks to change plates, selecting plate 1. During the trajectory, the user pauses the process, making the robot stop until instructed to start again. Finally, the user ends the feeding process with the "Stop" command in the waiting position.

IV. CONCLUSION

In the present work, a versatile system of simulated assisted feeding was implemented through the interaction between the user and a virtual environment, being integrated by means of Deep Learning techniques. The interaction was performed by means of control commands made by the user's hands, which indicated the process of the assisted feeding that the robot was going to perform and which plate it should go. This allows simple communication so that the person receiving the food can control the robot and the continuity with which the robot gives the food.

For the detection of the commands, two Faster R-CNN were used to facilitate the learning of the gestures, which were divided into two categories (control and plate selection) to avoid possible confusion with similar gestures plate 4 and pause. With this, the neural networks achieved average accuracies higher than 97.5% in the detection of commands, coming to locate the correct gesture not only in the test images but in the execution in real-time with a high degree of performance.

On the other hand, during the execution of the tests in real-time, the network for the control commands managed to recognize all the gestures made by the users. The network for the selection commands reached 95% in the hits, making a single mistake during all the tests. Also, the processing times in the detection were less than 300 ms, allowing a fast response time enough for the user to control the robot properly, especially when the user wanted it to stop it during the process of executing the feeding task.

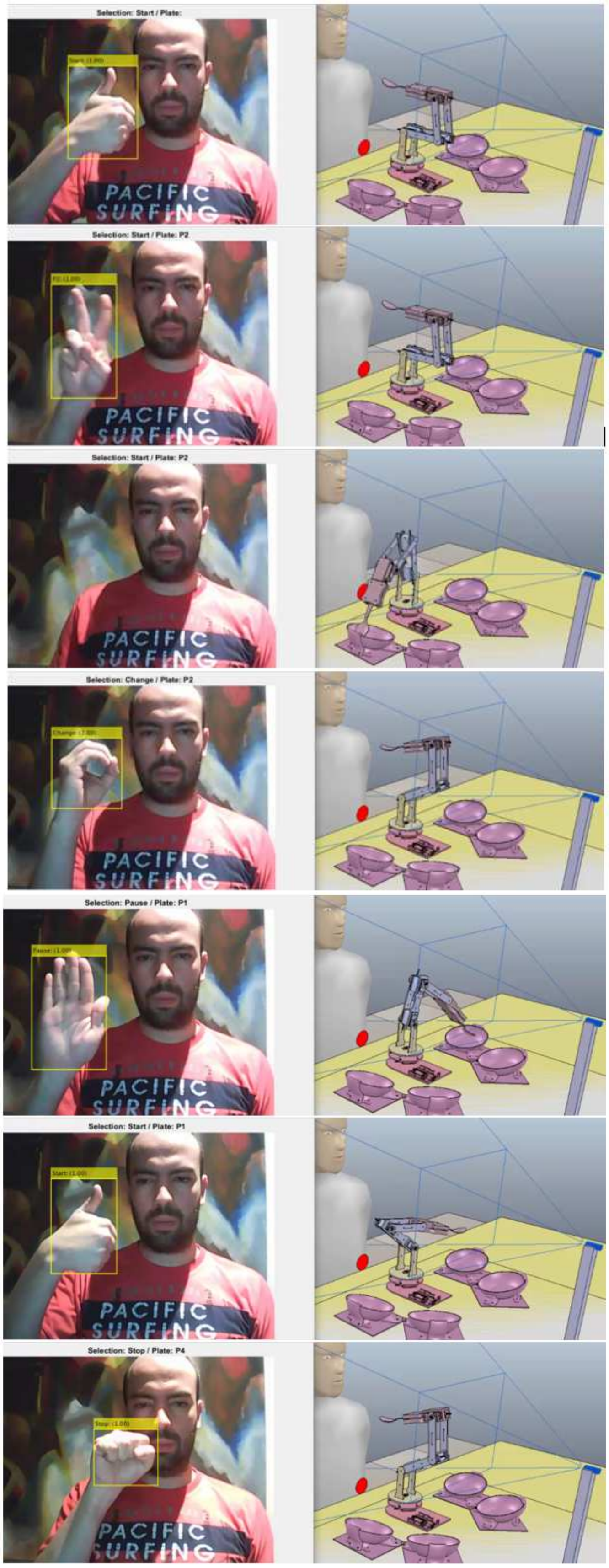
Selection: Start / Plate:
Start (1.00)
Selection: Start / Plate: P2
P2 (1.00)
Selection: Start / Plate: P2
Selection: Change / Plate: P2
Change (1.00)
Selection: Pause / Plate: P1
Pause (1.00)
Selection: Start / Plate: P1
Start (1.00)
Selection: Stop / Plate: P4
Stop (1.00)

Fig. 10 A graphic example of the feeding task and interaction of the user with the virtual environment.

ACKNOWLEDGMENT

The authors are grateful to the Universidad Militar Nueva Granada, which, through its Vice-chancellor for research, finances the present project with code IMP-ING-2935 (being in force 2019-2020) and titled " Prototipo robótico flexible para asistencia alimentaria ", from which the present work is derived.

REFERENCES

[1] Elly A. Konijn, Johan F. Hoorn, Robot tutor and pupils' educational ability: Teaching the times tables, Computers & Education, Volume 157, 2020, 103970, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2020.103970.

[2] Beiqun Zhao, Hannah M. Hollandsworth, Arielle M. Lee, Jenny Lam, Nicole E. Lopez, Benjamin Abbadessa, Samuel Eisenstein, Bard C. Cosman, Sonia L. Ramamoorthy, Lisa A. Parry, Making the Jump: A Qualitative Analysis on the Transition From Bedside Assistant to Console Surgeon in Robotic Surgery Training, Journal of Surgical Education, Volume 77, Issue 2, 2020, Pages 461-471, ISSN 1931-7204, https://doi.org/10.1016/j.jsurg.2019.09.015.

[3] Giancarlo Albo, Elisa De Lorenzis, Andrea Gallioli, Luca Boeri, Stefano P. Zanetti, Fabrizio Longo, Bernardo Rocco, Emanuele Montanari, Role of Bed Assistant During Robot-assisted Radical Prostatectomy: The Effect of Learning Curve on Perioperative Variables, European Urology Focus, Volume 6, Issue 2, 2020, Pages 397-403, ISSN 2405-4569, https://doi.org/10.1016/j.euf.2018.10.005.

[4] L. Bresler, M. Perez, J. Hubert, J.P. Henry, C. Perrenot, Residency training in robotic surgery: The role of simulation, Journal of Visceral Surgery, Volume 157, Issue 3, Supplement 2, 2020, Pages S123-S129, ISSN 1878-7886, https://doi.org/10.1016/j.jviscsurg.2020.03.006.

[5] Yoji Yamada, Yasuhiro Akiyama, Chapter 14 - Physical Assistant Robot Safety, Editor(s): Jacob Rosen, Peter Walker Ferguson, Wearable Robotics, Academic Press, 2020, Pages 275-299, ISBN 9780128146590, https://doi.org/10.1016/B978-0-12-814659-0.00014-X.

[6] Francesco Lanza, Valeria Seidita, Antonio Chella, Agents and robots for collaborating and supporting physicians in healthcare scenarios, Journal of Biomedical Informatics, Volume 108, 2020, 103483, ISSN 1532-0464, https://doi.org/10.1016/j.jbi.2020.103483.

[7] Jasmin Grischke, Lars Johannsmeier, Lukas Eich, Leif Griga, Sami Haddadin, Dentronics: Towards robotics and artificial intelligence in dentistry, Dental Materials, Volume 36, Issue 6, 2020, Pages 765-778, ISSN 0109-5641, https://doi.org/10.1016/j.dental.2020.03.021.

[8] Ayman A Abu Baker, Yazeed Ghadi, Mobile robot controller using novel hybrid system, International Journal of Electrical and Computer Engineering (IJECE), Vol 10, No 1: February 2020, p. 1027-1034.

[9] Pietro Perconti, Alessio Plebe, Deep learning and cognitive science, Cognition, Volume 203, 2020, 104365, ISSN 0010-0277, https://doi.org/10.1016/j.cognition.2020.104365.

[10] Qing Gao, Jinguo Liu, Zhaojie Ju, Robust real-time hand detection and localization for space human–robot interaction based on deep learning, Neurocomputing, Volume 390, 2020, Pages 198-206, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.02.066.

[11] Jiménez-Moreno Robinson, Pinzón-Arenas Javier Orlando, Pachón-Suescún César Giovany, Assistant robot through deep learning, International Journal of Electrical and Computer Engineering (IJECE), Vol 10, No 1: February 2020, p. 1053-1062.

[12] Ivet Rafegas, Maria Vanrell, Luís A. Alexandre, Guillem Arias, Understanding trained CNNs by indexing neuron selectivity, Pattern Recognition Letters, 2019, ISSN 0167-8655. https://doi.org/10.1016/j.patrec.2019.10.013.

[13] Paula Useche, Robinson Jimenez-Moreno, Javier Martinez Baquero, Algorithm of detection, classification and gripping of occluded objects by CNN techniques and Haar classifiers, International Journal of Electrical and Computer Engineering (IJECE), Vol 10, No 5: October 2020, p. 4712-4720.

[14] A-reum Lee, Yongwon Cho, Seongho Jin, Namkug Kim, Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room, Computer Methods and Programs in Biomedicine, Volume 190, 2020, 105385, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2020.105385.

[15] Safa Ameur, Anouar Ben Khalifa, Med Salim Bouhlel, A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with Leap Motion, Entertainment Computing, Volume 35, 2020, 100373, ISSN 1875-9521, https://doi.org/10.1016/j.entcom.2020.100373.

[16] Linpu Fang, Ningxin Liang, Wenxiong Kang, Zhiyong Wang, David Dagan Feng, Real-time hand posture recognition using hand geometric features and Fisher Vector, Signal Processing: Image Communication, Volume 82, 2020, 115729, ISSN 0923-5965, https://doi.org/10.1016/j.image.2019.115729.

[17] Yifan Zhang, Lei Shi, Yi Wu, Ke Cheng, Jian Cheng, Hanqing Lu, Gesture recognition based on deep deformable 3D convolutional neural networks, Pattern Recognition, Volume 107, 2020, 107416, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2020.107416.

[18] Adithya V., Rajesh R., A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition, Procedia Computer Science, Volume 171, 2020, Pages 2353-2361,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.255.

[19] Edwin Jonathan Escobedo Cardenas, Guillermo Camara Chavez, Multimodal hand gesture recognition combining temporal and pose information based on CNN descriptors and histogram of cumulative magnitudes,Journal of Visual Communication and Image Representation, Volume 71, 2020, 102772, ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2020.102772.

[20] A. Mariani, E. Pellegrini and E. De Momi, "Skill-oriented and Performance-driven Adaptive Curricula for Training in Robot-Assisted Surgery using Simulators: a Feasibility Study," in IEEE Transactions on Biomedical Engineering, doi: 10.1109/TBME.2020.3011867.

[21] Christian Brecher, Stephan Wein, Xiaomei Xu, Simon Storms, Werner Herfs, Simulation Framework for Virtual Robot Programming in Reconfigurable Production Systems, Procedia CIRP, Volume 86, 2019, Pages 98-103, ISSN 2212-8271, https://doi.org/10.1016/j.procir.2020.01.045.

[22] Handy Wicaksono, Claude Sammut, A cognitive robot equipped with autonomous tool innovation expertise, International Journal of Electrical and Computer Engineering (IJECE), Vol 10, No 2: April 2020, p. 2200-2207.

[23] Ren, Shaoqing, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems. 2015. p. 91-99.

[24] He, Kaiming, et al. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.