International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 6, December 2020, pp. 6300~6308
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i6.pp6300-6308

Tool delivery robot using convolutional neural network

Javier Pinzon-Arenas, Robinson Jiménez-Moreno
Faculty of Engineering, Militar Nueva Granada University, Colombia

Article Info

Article history:

Received Oct 30, 2019
Revised Jun 6, 2020
Accepted Jun 17, 2020

Keywords:

Convolutional neural network
Delivery status adjustment
Hand tracking
Human-robot interaction
Path planning

ABSTRACT

In the following article, it is presented a human-robot interaction system where algorithms were developed to control the movement of a manipulator in order to allow it to search and deliver, in the hand of the user, a desired tool with a certain orientation. A convolutional neural network (CNN) was used to detect and recognize the user's hand, geometric analysis for the adjustment of the delivery status of the tool from any position of the robot and any orientation of the gripper, and a trajectory planning algorithm for the movement of the manipulator. It was possible to use the activations of a CNN developed in previous works for the detection of the position and orientation of the hand in the workspace and thus track it in real time, both in a simulated environment and in a real environment.

Corresponding Author:

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

1. INTRODUCTION

The human-robot interaction requires a machine vision system that allows the robot to recognize the human movement and interpret it as an instruction for the execution of a certain task, as clarified in [1] and [2]. Some of these cooperative works lead to the need to develop algorithms that allow the interaction between both, making control over the force exerted by the manipulator at the point of contact with the human, as explained in [3], or using an intelligent support system in the execution of tasks performed by an operator, where human effort is minimized and the execution of the task is optimized, as shown in [4].

For the developed application exposed in this article, human intention is represented as the hand that expects to receive a tool, and the task of the robot is to go and deliver the tool in it, which can change position and orientation at any time. This type of applications requires a machine vision system that allows the manipulator to see and follow the hand continuously as well as a control system that moves the end effector of the manipulator to its destination, regardless of whether it is changing, and that in turn allows adjusting the orientation of the tool to be delivered, so that it can be held by the user.

Some previous studies related to human-robot interaction involves sensor for the task of interaction in the gripper of a 3 DOF robot, as a previous stage of use of machine vision, as explained in [5]. Additionally, it must have the ability to recognize the joints of the hand and its varied positions [6] for the development of complex tasks such as video games and human-computer interactions. Even taking jobs like wearable robots where the interaction allows reduce the human metabolic energy cost in physical movements [7]. All this set of activities involves, generally, the use of cameras or depth sensors that allow to extract the hand from the environment, together with its articulations, as shown in [8, 9], and that simultaneously, manage to track with techniques like artificial intelligence or the use of correlations filters, respectively.

Artificial intelligence technique based on deep learning is exposed in [10] for give smart attributes to a robot used as assistant, the convolutional neural network (CNN) was used to develop the algorithm to human robot interaction. Convolutional networks have many applications in the top of the state of art, example like biological image classification [11], handwritten character recognition [12] and detection of packed and unpacked malware [13], they are recent researches of different science field's that use this technique.

Based on it, this work is oriented to integrate the above concepts in the development of a new human-robot interaction system, in charge of facilitating different types of tools to a user through an anthropomorphic manipulator, whose displacement depends on the location of the hand, where the enviroment is captured by a camera and recognized by means of a CNN. For the final orientation of the gripper of the manipulator, a correction of the delivery status of the tool is generated, which allows the robot to leave the instrument in the hand of the user in such a way that it can generate a cylindrical grip on the element, as a complementary support of state of art in human-robot interactions. As a complement of previous works relates in the state of art like [14, 15].

The development of this application is fundamental for the programming of an assistive robot, for example, in the medical area, where it operates as an instrumentalist with the ability to provide the surgical instruments to the doctor [16], delivering them in the hand with the appropriate orientation for its support. In previous works, the recognition and classification part of surgical instrumentation has been solved to allow the manipulator to find the tool requested by the user, hold it and take it to a certain point, in addition to training a CNN for the classification of hand gestures (such as open or closed), as shown in [17]. In such a way that the next step is based on the delivery of the instrument in the palm of the user, which is exposed below, as a contribution to the state of the art.

The integration of a robotic agent with artificial intelligence systems is common [18], where CNNs allow this integration at different scales such as robotic control [19], cleaning robots [20] or recyclers [21]. This technique has demonstrated its versatility [22] and articulation capacity with other techniques [23, 24], even in virtual robotic work environments [25], which is why they are the technique used in this development. The paper is divided into 3 main sections, where the first explains in detail the methodology developed for the execution of the hand tracking algorithm by the manipulator, and the adjustment of the delivery status of the tool. The following presents the results and analyzes achieved, and in the last section the conclusions obtained from the application are presented.

2. RESEARCH METHOD

A robotic assistance algorithm was developed whose objective is to provide tools to a user by constantly monitoring his hand within a limited work area, and finalizing the delivery process with an adjustment of the orientation of the tool, rotating it until it is arranged in such a way that a cylindrical grip can be generated on it, as shown in Figure 1.

A CNN was trained, and segmentation algorithms were used for hand recognition during a video sequence and data augmentation techniques [26]. Subsequently, the inverse kinematics of the manipulator was calculated to execute its displacement, and geometric analysis was used to adjust the delivery status of the tool with respect to the position of the user's hand, at the moment when the robot stops on the palm of the hand. Figure 2 shows the sequence of basic operation of the algorithm, in which a total of 5 steps are demarcated, where the first takes the manipulator to the initial position shown in Figure 7a, with the tool held by the gripper, and the following three are responsible for executing the tracking of the hand and the adjustment of the delivery of the tool, repeating in a cycle that only ends when the gripper is on the hand and the tool has already been accommodated for the grip, then it is passed to the fifth step and the algorithm is finished.

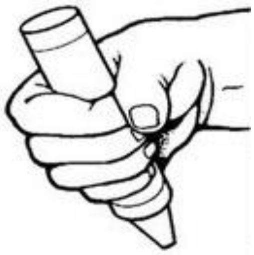

Figure 1. Cylindrical grip

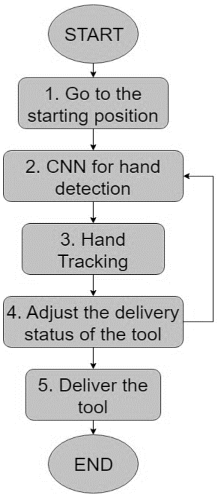
START
1. Go to the starting position
2. CNN for hand detection
3. Hand Tracking
4. Adjust the delivery status of the tool
5. Deliver the tool
END

Figure 2. Algorithm flowchart

2.1. Manipulator trajectory

An algorithm was developed that allows to set a curved path between the hand of the user and the manipulator, and recalculate it each time the position of the hand varies significantly. The trajectory is calculated in such a way that the speed of displacement of the end effector is controlled, ensuring a linear velocity equal to zero at the beginning and end of the trajectory, describing a parabolic behavior as shown in Figure 3, where the maximum velocity in the middle of the path is reached. For the displacement of the end effector, a trajectory with cubic behavior is used for the X axis, the Y axis and the Z axis, as shown in Figure 4, where the robot generates small displacements at the ends of the trajectory and great advances in the middle of it, thus allowing to ensure the desired speed.

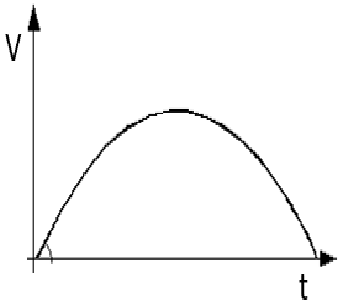

Figure 3. End effector velocity

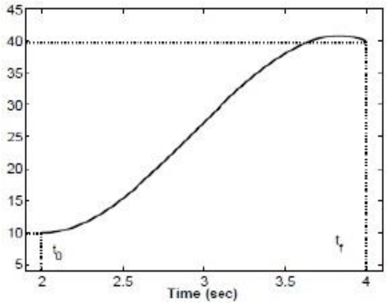

Figure 4. Displacement of the end effector

To calculate each of the points of the trajectory in the three coordinate axes, (1) was used, which describes the cubic behavior shown in Figure 4, where x(t) is the position function for any of the coordinate axes, a,b,c,d are constants, and t is the time, which is between 0, for the initial position, and the total time of the execution of the movement (Tt), defined by the user. For the calculation of each of the constants of (1), the velocity function of (2) was obtained, deriving equation (1) with respect to time, and it was established that the initial (v_i) and final velocity (v_f) should be equal to zero, as previously shown in Figure 3.

$$x(t) = at^3 + bt^2 + ct + d \tag{1}$$

$$v(t) = at^2 + bt + c \tag{2}$$

Then, the position and velocity functions were evaluated for a time t=0 and for t=Tt to find the constants a,b,c,d. Finally, the constants found in (1) were replaced, and t was replaced by different equispaced time instants to generate the complete trajectory as a set of 50 intermediate points between the initial (x(0)) and final (x(Tt)) position of the trajectory, and the manipulator and the inverse kinematics presented in [14] were used to execute the movement in the application.

2.2. CNN for the detection and orientation of the hand

The CNN trained in [13] was used for the recognition and classification of the hands, and a segmentation algorithm was developed in which an image of the working background is captured and the average of the tones of the whole image is obtained, in such a way that the object that is located under the camera and that has a different color to the background will be recognized as the user's hand, and will be cropped from the original image to be sent to the CNN. Data augmentation techniques have been used in training to improve the database used [27].

Once that image is entered in the CNN, the first activation of the network is extracted, i.e. the result of passing the image through the first ReLU (Rectified Linear Unit) layer of the network, and then it is binarized in order to obtain the overall shape of the hand, as shown in Figure 5. This form allows knowing the orientation of the hand, generating an ellipse on the largest number of white pixels of the binarized image, and taking the angle of inclination between the horizontal of the image and the major axis of the ellipse θ_m shown in Figure 5. Additionally, the final position of the hand is obtained when extracting the center of the box that covers it, as shown in Figure 6 ("+" sign). In this way, the manipulator seeks to bring the tool to the "+" symbol, and when placed on it, adjusts the orientation of the tool, and makes sure that when finished, it has not changed neither the location of the hand nor its orientation to finally deliver the tool.

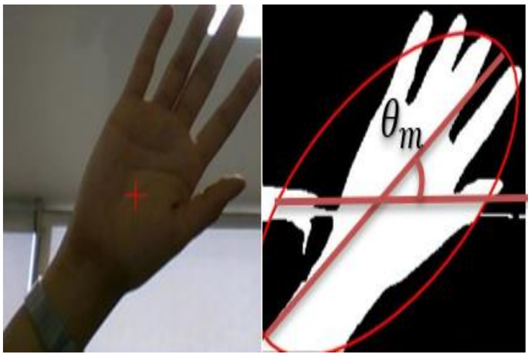

Figure 5. Orientation of the hand

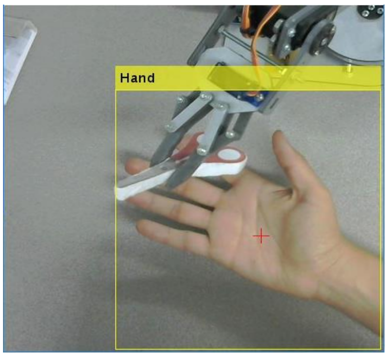

Figure 6. Tracking the hand in a real environment

2.3. Hand tracking algorithm

Once the manipulator has grasped the desired tool and moved it to the position shown in Figure 7(a), it is proceeded to calculate the first tracking path of the hand, generating 50 points between the current position of the final effector (initial position x(0)) and the current position of the hand (final position x(Tt)), which is obtained by means of the segmentation algorithm, which extracts the coordinates of this one from an image of the workspace, like the one shown in Figure 7(b).

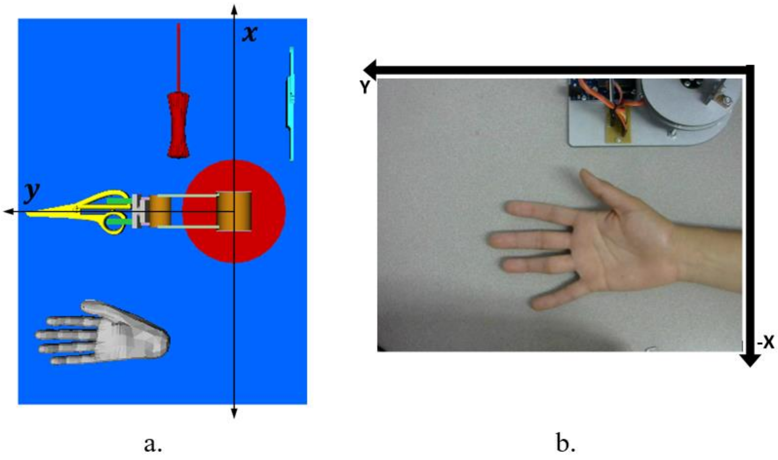

Figure 7. (a) Initial position for hand tracking and (b) example of workspace capture

Once the trajectory is generated, the manipulator moves point by point using the inverse kinematics, and in each step a photo of the workspace is captured again to search the hand again and thus detect changes of its position, in a constant way. Depending on how far the new position of the hand is located with respect to the first detected position, stored as x(Tt), the algorithm is passed to a series of conditionals that change the trajectory of the manipulator to bring it to a new final position. Figure 8 shows the trajectory changes generated by the variation in the location of the hand during the tracking process, where x(ta) is the current position of the hand, u1,u2,u3 are the distance thresholds that determine how close or far the hand is from the position x(Tt), P1%,P2%,P3% are percentages where P1%>P2%>P3%, Pa represents how much percentage of the trajectory has traveled the manipulator so far, and P is equal to the percentage P1%, P2% or P3% of the conditional to which the program has entered.

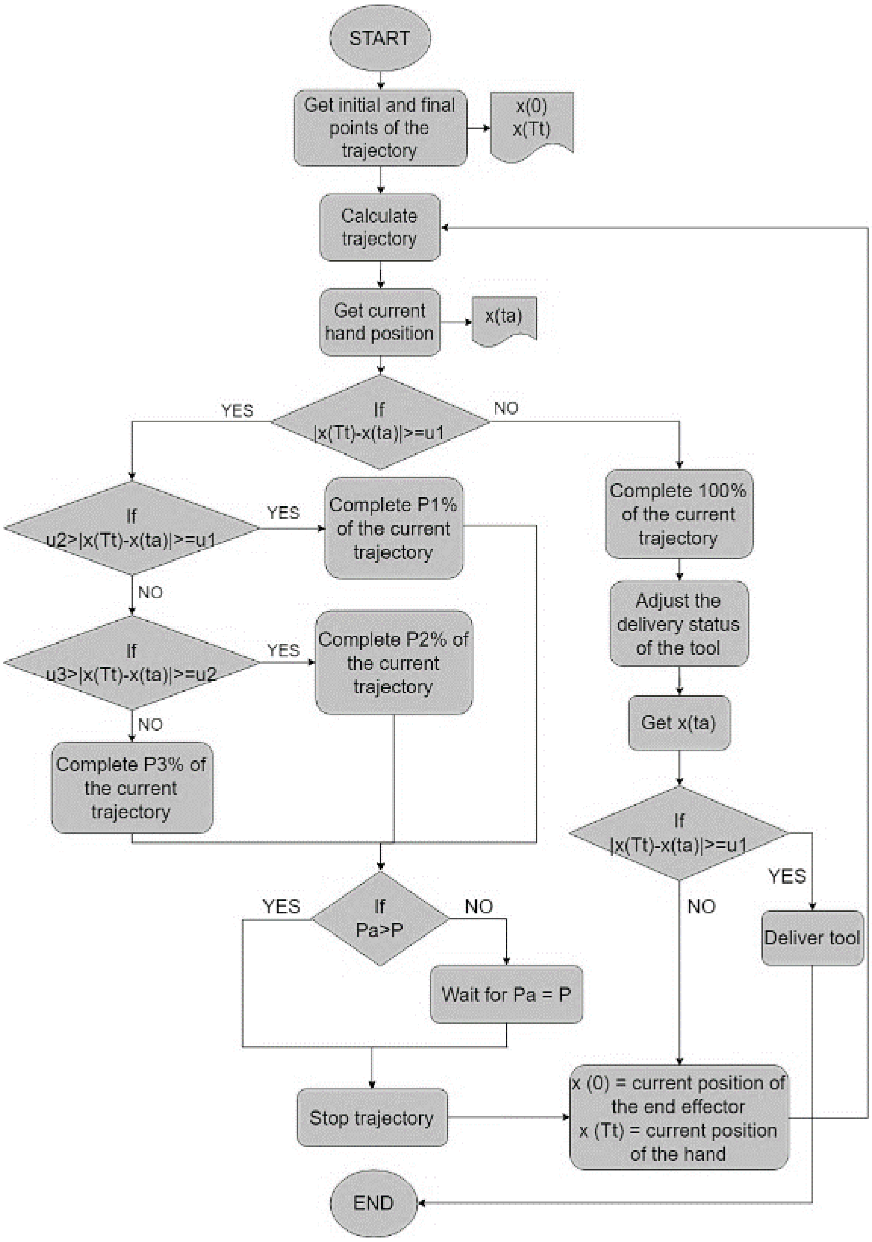
START
Get initial and final points of the trajectory
x(0)
x(Tt)
Calculate trajectory
Get current hand position
x(ta)
If |x(Tt)-x(ta)|>=u1
YES
NO
If u2>|x(Tt)-x(ta)|>=u1
YES
Complete P1% of the current trajectory
NO
If u3>|x(Tt)-x(ta)|>=u2
YES
Complete P2% of the current trajectory
NO
Complete P3% of the current trajectory
Complete 100% of the current trajectory
Adjust the delivery status of the tool
Get x(ta)
If |x(Tt)-x(ta)|>=u1
YES
Deliver tool
NO
If Pa>P
YES
NO
Wait for Pa = P
Stop trajectory
x (0) = current position of the end effector
x (Tt) = current position of the hand
END

Figure 8. Hand tracking conditionals

In the first instance, it is evaluated if the difference between x(Tt) and x(ta) is less than u1, i.e. if the hand is close enough to its previous position to consider that it has not moved, otherwise it is evaluated how far it is from its first position. For a distance between u1 and u2, it is said that the hand is not very far from x(Tt), so a percentage P1% close to 100% is established so that the robot almost ends its trajectory before correcting its direction in search of the hand. For a distance between u2 and u3 is said that the hand is fairly far away, so a P2% percentage close to 50% is set to avoid the end effector moving to a point not very close to its final objective. For a distance greater than u3, it is said that the hand is very far from its first position, so a P3% close to 0% is set to recalculate the trajectory quickly and correct the direction of advance.

Then, it is evaluated what percentage of the trajectory the manipulator has moved (Pa), and if it is greater than P, the trajectory is immediately stopped and the next one is recalculated with the new values of x(0) and x(Tt), otherwise, it is expected that the manipulator reaches the percentage P of advance before stopping the trajectory and calculating the next one. In case the difference between x(Tt) and x(ta) is less than u1, the established trajectory is completed, the delivery status of the tool is adjusted, the current position of the hand is again compared with x(Tt), and if it is still less than u1, the tool is delivered and the monitoring is completed, otherwise a new trajectory is recalculated with the new values of x(0) and x(Tt).

2.4. Delivery status adjustment

Each time the manipulator completes 100% of its trajectory, the degree of rotation required by the gripper is calculated to equalize its orientation to that of the hand and thus adjust the tool to a cylindrical grip. The program is responsible for equaling the angle of rotation of the gripper θ_P to the orientation of the hand θ_m, for which (3) to (6) are applied, where (3) calculates the angle of rotation θ_R between the point where the tool is and the horizontal axis of the system as shown in Figure 9, in (4) calculates the current orientation of the gripper θ_P adding the previous rotations θ_i to its initial orientation that is at -90° from θ_R. By (5), the difference between the current rotation of the hand θ_m and the current orientation of the gripper is calculated, in order to add this difference to θ_P to equal it to the orientation of the hand, leading by default the tool to be located perpendicular to the sagittal plane of the hand to allow the cylindrical grip. In (6) updates θ_i, adding the new rotation applied θ_{Rot} to the previous rotations.

$$\theta_R = 180° - \tan^{-1}\frac{W-y}{H-x} \tag{3}$$

$$\theta_P = \theta_R - 90° + \theta_i \tag{4}$$

$$\theta_{Rot} = \theta_m - \theta_P \tag{5}$$

$$\theta_i = \theta_{Rot} + \theta_i \tag{6}$$

Where W and H are the width and height of the image captured by the camera, respectively, and X, Y are the current coordinates of the end effector at the end of 100% of a trajectory. Figure 9 shows the angles and dimensions used for the adjustment of the delivery status, as well as an example of a first orientation already made and the new angles to be calculated, to generate the adjustment of the tool to a new position of the hand. This algorithm allows adjusting the delivery status of the tool from any orientation of the gripper to any orientation of the hand, useful for cases in which after having made a first adjustment, the hand has moved and the robot must follow it again and make a second adjustment.

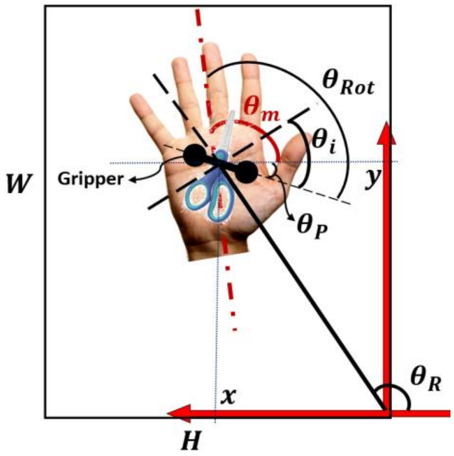

Figure 9. Delivery status adjustment

3. RESULTS AND DISCUSSION

Figure 10 shows the result obtained by the algorithm for a real work environment such as that posed in Figure 7(b), whose behavior follows the movements of the simulated robot in VRML of MATLAB®, performing the delivery of the tool in both environments, virtual and in real time, simultaneously. As can be seen in Figure 10, the simulation shows the tracking performed by the robot towards the hand and the adjustment of the delivery status of the tool, such as the position and orientation of the hand in the real environment. The only difference between both situations is that the real robot does not adjust the orientation of the tool because it does not have this degree of freedom. Ten tests were performed to determine the average time it takes for the manipulator to deliver the tool, from its initial position for tracking the hand see Figure 7(a) to the adjustment of the delivery status, making 5 different position changes for each test. Additionally, from each of the tests, the distance between the position of the end effector and the hand position, extracted by the algorithm, was measured to obtain a margin of error in the accuracy of the delivery. Table 1 shows the time used by the application for the delivery of the tool and the distance difference between the desired point and the final effector, taken in millimeters.

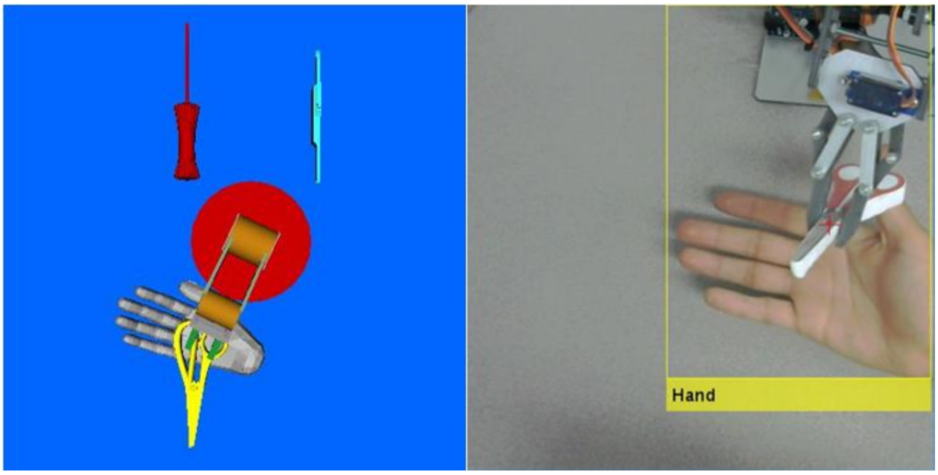

Figure 10. Delivery of the tool both in simulation and in a real environment

Table 1. Results obtained

Test	Time (s)	Distance error (mm)
1	15.14	1
2	19.57	5
3	10.25	17
4	15.45	8
5	13.73	3
6	16.04	2
7	15.01	1
8	16.30	2
9	16.76	3
10	15.32	9
Average	**15.357**	**5.1**

The application takes around 15s to track the hand, with changes in position during the application, and it has a margin of error of less than 10mm in most cases. As can be seen in Figure 10, one of the greatest difficulties of the application is the recognition of the hand when the manipulator is near or on top of it. Because the object detection algorithm includes all those whose difference of tones with the background is higher than a threshold, the algorithm tends to recognize the robot as part of the hand, which generates that the size of the box that covers it expands and covers the manipulator, displacing the real position of the hand, as shown in Figure 10. On the other hand, the presence of the manipulator on the hand causes deterioration or failure of the detection of the orientation of the hand, because in the first activation of the network the robot is recognized as part of the hand, causing the ellipse to encompass part of the robot and deflect the inclination.

4. CONCLUSION

This algorithm is the ideal complement for human-robot interaction works, such as the development of assistance robots, since it is capable of ensuring the delivery of the object desired by the user regardless of the location of his hand, its orientation or any change in delivery conditions that occurs at any point in the interaction process. The delivery of tools made by the robot allows the user to concentrate on the task he is developing, either a surgery or the repair of a vehicle, since the user do not have to spend part of the time looking for a tool among a large set of elements, but only ask for it and wait for it to be delivered.

It is proposed to train a CNN that recognizes the hand even when it has an obstruction on it, in order to obtain activations capable of focusing clearly on the hand, and not on the robot, and to ensure that the box covering the hand is more accurate and does not cover more than necessary. As a second option, a more robust object detection algorithm could be used that only extracts the hand from the work environment and enters it into the CNN already trained. Despite the difficulties presented when detecting the position and orientation of the hand, the algorithm has the ability to follow the desired end point, even if it moves and changes position constantly, and can also correct the orientation of the tool, the number of times it is necessary, until it detects that it really is in the position and orientation necessary to carry out the delivery.

ACKNOWLEDGEMENTS

The authors are grateful to the Militar Nueva Granada University, which through its Vice chancellor for research finances the present project with code IMP-ING-2290 and titled "Prototype of robot assistance for surgery". Likewise, acknowledgement is given to the participation of the engineer Paula Useche who, through the service provision order number 0721/2018, was hired for the development of the exposed algorithms.

REFERENCES

[1] Anna Henschel, Ruud Hortensius, Emily S. Cross, “Social Cognition in the Age of Human–Robot Interaction,” *Trends in Neurosciences,* vol. 43, no. 6, pp. 373-384, 2020.

[2] Angeliki Zacharaki, Ioannis Kostavelis, “Antonios Gasteratos, Ioannis Dokas, Safety bounds in human robot interaction: A survey,” *Safety Science,* vol. 127, 2020.

[3] Yueyue Liu, Zhijun Li, Huaping Liu, Zhen Kan, “Skill transfer learning for autonomous robots and human–robot cooperation: A survey,” *Robotics and Autonomous Systems,* vol. 128, 2020.

[4] S. Ehsan Hashemi-Petroodi, Simon Thevenin, Sergey Kovalev, Alexandre Dolgui, “Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey,” *Annual Reviews in Control,* 2020.

[5] Amin Hamed, Mehdi Tale Masouleh, Ahmad Kalhor, “Design and characterization of a bio-inspired 3-DOF Tactile/Force sensor and implementation on a 3-DOF decoupled parallel mechanism for human-robot interaction purposes,” *Mechatronics,* vol. 66, 2020.

[6] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, Sharp, T., et al, “Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences,” *ACM Transactions on Graphics (TOG),* vol. 35, no. 4, pp. 1-12, 2016.

[7] Jing Fang, Yuan Yuan, “Human-in-the-loop optimization of wearable robots to reduce the human metabolic energy cost in physical movements,” *Robotics and Autonomous Systems,* vol. 127, 2020.

[8] Yong Wang, Xian Wei, Hao Shen, Lu Ding, Jiuqing Wan, “Robust fusion for RGB-D tracking using CNN features,” *Applied Soft Computing,* vol. 92, 2020.

[9] Di Yuan, Wei Kang, Zhenyu He, “Robust visual tracking with correlation filters and metric learning,” *Knowledge-Based Systems,* vol. 195, 2020.

[10] Javier Orlando Pinzón-Arenas, César Giovany Pachón-Suescún and Robinson Jiménez-Moreno, “Assistant robot through deep learning,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 1, pp. 1053-1062, 2020.

[11] Jiaohua Qin, Wenyan Pan, Xuyu Xiang, Yun Tan, Guimin Hou, “A biological image classification method based on improved CNN,” *Ecological Informatics,* vol. 58, 2020.

[12] Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab, Debdatta Kandar, “Handwritten Character Recognition from Images using CNN-ECOC,” *Procedia Computer Science,* vol. 167, pp. 2403-2409, 2020.

[13] Danish Vasan, Mamoun Alazab, Sobia Wassan, Babak Safaei, Qin Zheng, “Image-Based malware classification using ensemble of CNN architectures (IMCEC),” *Computers & Security,* vol. 92, 2020.

[14] Paula Useche, Robinson Jiménez-Moreno and 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. 2. pp. 1308-1316, 2020.

[15] César Giovany Pachón-Suescún, Carlos J. Enciso-Aragon and Robinson Jiménez-Moreno, “Robotic navigation algorithm with machine vision,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 1, pp. 1329-1342, 2020.

[16] Rachelle Morgenstern, Edwin P. Su, "The gripper table mounted retraction system: A tireless surgical assistant," *Seminars in Arthroplasty,* vol. 29, no. 3, pp. 134-136, 2018.

[17] J. O. P. Arenas, P. C. U. Murillo and R. J. Moreno, "Convolutional neural network architecture for hand gesture recognition," *IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON),* pp. 1-4, 2017.

[18] Tresna Dewi, Siti Nurmaini, Pola Risma, Yurni Oktarina, Muhammad Roriz, "Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 2, pp. 1376-1386, 2020.

[19] Paula Catalina Useche M. and Robinson Jiménez-Moreno, "Detection and gripping of an occluded tool using DAG-CNN," *ARPN Journal of Engineering and Applied Sciences,* vol. 15, no. 9, pp. 1069-1078, 2020.

[20] Maryam Kouzehgar, Yokhesh Krishnasamy Tamilselvam, Manuel Vega Heredia, Mohan Rajesh Elara, "Self-reconfigurable façade-cleaning robot equipped with deep-learning-based crack detection based on convolutional neural networks," *Automation in Construction,* vol. 108, 2019.

[21] Zeli Wang, Heng Li, Xiaoling Zhang, "Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach," *Automation in Construction,* vol. 97, pp. 220-228, 2019.

[22] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[23] Enciso-Aragón, C. J., Pachón-Suescún, C. G. and Jimenez-Moreno, R., "Quality control system by means of CNN and fuzzy systems," *International Journal of Applied Engineering Research,* vol. 13, no. 16, pp. 12846-12853, 2018.

[24] Fengyi Zhang, Zhiyong Liu, Fangzhou Xiong, Jianhua Su, Hong Qiao, "WAGNN: A Weighted Aggregation Graph Neural Network for robot skill learning," *Robotics and Autonomous Systems,* vol. 130, 2020.

[25] Furrer, F., Burri, M., Achtelik, M. and Siegwart, R., "RotorS-A modular Gazebo MAV simulator framework," *Robot Operating System (ROS),* pp. 595-625, 2016.

[26] Yong Wang, Xian Wei, Xuan Tang, Hao Shen, Lu Ding, "CNN tracking based on data augmentation," *Knowledge-Based Systems,* vol. 194, 2020.

[27] Tianhao Hu, Tang Tang, Ronglai Lin, Ming Chen, Shufa Han, Jie Wu, "A simple data augmentation algorithm and a self-adaptive convolutional architecture for few-shot fault diagnosis under different working conditions," *Measurement,* vol. 156, 2020,

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 PhD 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