International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 6
ISSN: 2088-5334

Tool Sorting Algorithm Using Faster R-CNN and Haar Classifiers

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

[a] *Mechatronics Engineering Department, Universidad Militar Nueva Granada, Bogotá D.C, 110111, Colombia*

*Corresponding author: *robinson.jimenez@unimilitar.edu.co*

***Abstract*— The following paper presents an algorithm for sorting up to 5 different tools based on deep learning and specifically in a convolutional neural network, according to the top in pattern recognition found in state of the art and compared by a Haar classifier in object recognition tasks. A Faster R-CNN is used to detect and classify tools located randomly on a table and a Haar classifier to detect other tools delivered by the user. The Faster R-CNN allows recognizing the existing tools on the table and where they are located in the physical space. The Haar classifier detects and tracks, in real-time, a tool delivered by the user's hand to sort it on the table, together with the other elements. Both the training of the convolutional network and the design of the Haar classifier are exposed. The algorithm detects and classifies the tools found on a table, then orders them side by side, and finally waits for the user to deliver some of the five missing tools on the table, take it from his hand, and locate it at the end of the row of objects. A Faster R-CNN was used with an accuracy of 70.8% and a Haar classifier with a 96% recognition, managing to order the five tools in a physical environment. The average time in comparison demonstrates that the Haar classifier presents a lower computational cost.**

***Keywords*— Collaborative robotics; faster R-CNN; Haar classifier; object recognition; ordering algorithm.**

Manuscript received 21 Jul. 2020; revised 28 Feb. 2021; accepted 24 Apr. 2021. Date of publication 31 Dec. 2021.

I. INTRODUCTION

Collaborative robotics has sought to increase the variety and quality of services a manipulator can deliver to the user [1]. It is the ordering of objects in boxes or containers, where the aim is to adapt the robot's ordering logic to the user's requirements by learning order preference patterns [2]. For applications like the Human-robot coexistence and interaction in open industrial cells [3]. Where robotic grasping actions are necessary, for example, learning to grasp an arbitrary object from visual input [4].

On the other hand, the task of organizing objects also involves the use of detection algorithms and element recognition to give the robot the ability to visualize and detect all the objects that must be sorted within a workspace [5], algorithms among which the convolutional neural networks (CNN) can be found. These networks use a series of filters of different dimensions trained from a database of images of the object or objects to be classified to later use said training as a basis for the classification of untrained images [6], [7].

Some of the multiple applications in which CNN can be implemented are the recognition and classification of faces [8],[9], recognition of traffic signals [10],[11], and visual analysis of written documents [12]. However, the CNN focuses on classifying the image within a single category, given that they were trained in that way, with a category by image, with multiples training for different databases such as MNIST, NORB, NIST SD 19 [13], [14]. Therefore, the architecture of a CNN is oriented to the classification of images based on the categories trained, but not to the detection of multiple objects to be classified within an image, for that reason, it is necessary to use object detection and classification algorithms, such as the Haar classifiers, the Region-Based Convolutional Neural Network (R-CNN) or Faster R-CNN.

Haar classifiers have been used for the detection of objects or parts of the face, such as the eyes, mouth, or the same face in general [15]. However, they can be adapted to perform detection and classification tasks [16], where several Haar classifiers were trained for the classification of up to 5 surgical instruments.

On the other hand, R-CNN and Faster R-CNN are convolutional neural networks that classify more than one object per image [17]. A comparison between handwritten word and speech records has been developed. A faster R-CNN is used to recognize and classify people, objects, and animals in different environments [18].

Faster R-CNN is one of the main architectures of CNN based on regions and used at the top of state of art. Its application is very diversified, like in medicine with heart localization target [19], likewise in applications on precision agriculture in pest detection [20]. Faster R-CNN is used in transmission images based on pseudo-color maps for application in the scattering of biological tissues [21]. These studies show the relevance of this Deep Learning technique, but its comparison with other classics techniques is understudied. For example, techniques like Haar classifiers that too use regions for detection are the focus presented here.

Next, a novel method of ordering is presented, based on the distance of each tool concerning the right side of the table and the distances between them. The artificial intelligence techniques such as the Haar classifiers and the Faster R-CNN are applied to obtain the necessary information of each object in the work environment. Thus, the algorithm can be executed using the distances and centers of each object to position the elements.

The present article is divided into four main sections: the first is the introduction, where a straightforward approach to collaborative robotics and algorithms of detecting and classifying objects by artificial intelligence is made. The second section describes the operation of the sorting algorithm, from the first recognition of the work area to the location of the last tool to be sorted. Later, in the third section, the results are presented, and an analysis is made of them to finally propose a series of conclusions regarding the work developed in the fourth section**.**

II. Materials and Methods

In the following work, Haar classifiers and Faster R-CNN are used to detect and classify up to 5 different tools to be sorted on a table. Faster R-CNN is used to detect and classify existing objects on the table and extract the locations and orientations of each of them, while Haar classifier is used to detect and track any of the tools, which are delivered directly by the user's hand, where a manipulator must follow it to hold it and sort it on the table next to the others, as shown in Fig. 1.

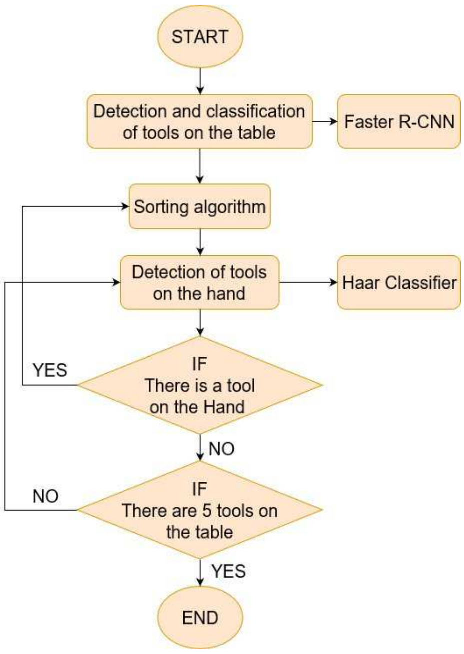


Fig. 1 Algorithm flowchart

In this sorting algorithm, up to 5 different tools are located in a row, both in a simulated environment and physically. There is no specific site for each type of tool, but they are organized one next to the other. This application automatically allows the ordering of multiple elements within a workspace, eliminating the need to employ human personnel for this task and ensuring constant distances between objects.

Next, four subsections are presented by means of which the functioning of the sorting algorithm is explained. The first section provides a basic description of the logic and general operation of the program. The second section presents the process of detection and classification of tools using a Faster R-CNN. In the third section, the logic of the ordering algorithm is explained, and in the fourth section, the process of detection of tools delivered by the user's hand, through Haar classifiers, is presented.

A. The Basic Operation of the Algorithm

The algorithm was designed to detect and classify up to 5 types of tools randomly located on a table, where their locations, orientations, and approximate dimensions of the space they occupy in the work environment are extracted, and a manipulator is programmed to perform the grip and sort of the detected tools, placing them next to each other with the same orientation.

Additionally, a function was designed to detect and grab one or more additional tools delivered by the user. The design can use a Haar classifier for tool detection to locate the object held by the user in hand and follow it continuously. When abrupt and sudden changes in its location are generated, it can reduce detection time that allows it to recognize changes every 5 ms approximately, as shown with the results of Table II. Once the manipulator is positioned on the object, the tool is grasped, and the sorting algorithm is used to place it next to the others, positioning it at the end of the row.

The first step is to detect and classify all the tools that are on the table using a Faster R-CNN. In this step, it is gotten the number of tools found, the position of each of them, their orientations, and the width and height dimensions of the space they occupy over the work area. The sorting tool algorithm is executed in the second step, which reorders all the tools found on the table, leaving them at a certain distance between them and the same height and orientation. To achieve this, the information extracted from the tools in the previous step is used, and they are ordered one by one, from right to left, leaving the first tool near the right margin of the table.

In the third step, it is looked for new tools to be sorted, where a Haar classifier is used to look for any of the five trained tools that the user is holding in his hand, and thus move the manipulator to follow the tool and grip it. Once grabbed, the ordering algorithm is again used to locate the new tool at the end of the row, at the previously determined distance. Where the cycle is repeated until the five tools are arranged on the table. The working space was organized as shown in Fig. 2, where for each camera used, an example of the image it captures within the environment was shown. The color of the table was alternated between blue and white to prevent the algorithm from memorizing it.

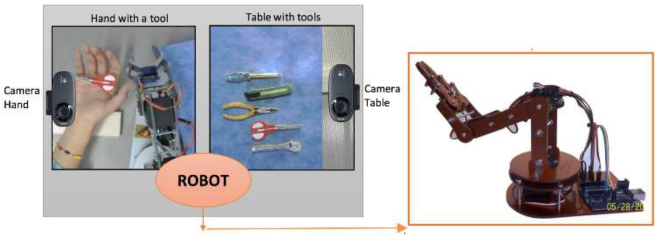


Fig. 2 Workspace

B. Step 1: Detection and classification of tools using Faster R-CNN

The algorithm starts with a preview of the workspace on the table (camera table of Fig. 2), where the user can locate the tools and make sure they are all within the camera's viewing range, then, the image is captured and sent to a trained Faster R-CNN with the five tools to classify: scalpel, screwdriver, spanner, scissor, and pincer. As shown in the example of Fig. 2, all the tools are between -15º and 15º of orientation concerning the camera’s vertical, to restrict the classification of objects to very small rotations, thereby facilitating their detection and correct classification.

The architecture of the network used, and its respective confusion matrix are shown in Fig. 3. The network was trained with a total of 716 images where each of them contains the five tools to be classified, and all were taken over the work area established for the application, where [22] explains the different layers of a convolutional neuronal network, which make up the Faster R-CNN [23], and [16] explains the confusion matrix presented below.

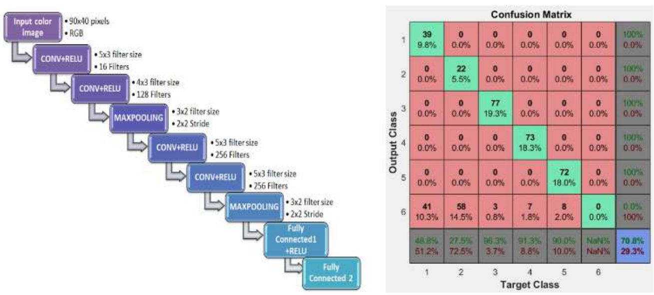


Fig. 3 Architecture and matrix confusion of the Faster R-CNN.

The four stages in which the training of the Faster R-CNN is divided were trained for 50, 100, 30, and 70 epochs, respectively. Stages 2 and 4 handled the training of the CNN that classifies the elements and the other two of the detection of them. Stage 3 and 4 are for fine-tuning. To test the network, 80 test images were used, each with the 5 tools, where the accuracy of 70.8% was reached, as shown in the confusion matrix of Fig. 3, where the numbers from 1 to 6 correspond to the scalpel, screwdriver, pliers, spanner, scissor and not found, respectively.

The confusion matrix shows that the most difficult tools to detect are scalpel and screwdriver, while the other three are detected quite easily. Additionally, the tools that were not correctly classified were stored in the not found category, which means that they were not classified due to difficulties presented in the detection mainly. Once the image is passed through the network, the classification of each of the tools found, their location, and the dimensions of the detection tables that cover them are given as output. Subsequently, the first activation of the network (output image of the first RELU) is used to segment the tool from the background and obtain its orientation, using an ellipse whose larger diagonal corresponds to the object's length.

C. Step 2: Sorting Algorithm

In step 2, from the regions obtained from the Faster R-CNN, all the positions and orientations of the tools are stored in arrays, and they are ordered from least to greatest distance with respect to the right margin of the table. The algorithm evaluates the matrix of positions and looks in it for the second tool closest to the right margin of the table, then takes the manipulator to that tool, grabs it, and takes it to the furthest point to the left to temporarily leave it there. Then, it is returned to the tool closest to the right margin and accommodates it near the edge of the table and parallel to it, then goes through the third tool of the ordered array, and so on until the only one that is not located is the one located at the far left.

To define the position of the next tool to be sorted and avoid overlapping with the previous one (already ordered), the width of the detection box that covers them was taken into account ("a1" for the previous tool, and "a2" for which it is going to be ordered), and a predetermined distance "d" in centimeters, defined by the user, which determines the distance between the detection box of the previous tool concerning the detection box of the tool to be sorted, as shown in Fig. 4.

Fig. 4 Distance between tools

After ensuring the distance between the boxes, the distance between the horizontal centers of each tool is calculated to define the global location of the object to be sorted (PosH) concerning the right margin of the table, obtaining said value with equation (1), where PosA is the distance between the previously ordered tool and the right side of the table. In Fig. 5, the distances PosA and PosH are shown for an application example.

$$PosH = PosA + d + \frac{a_1}{2} + \frac{a_2}{2} \quad (1)$$

Each time it finishes positioning a tool, PosA receives the value of PosH to define that distance as that of the last sorted tool. Hence, it can repeat the process for the following elements. Once all the tools on the table have been sorted, the manipulator retrieves the one left away from the work area and puts it at the end of the row. After that, the program continues with step 3 in search of additional tools to order.

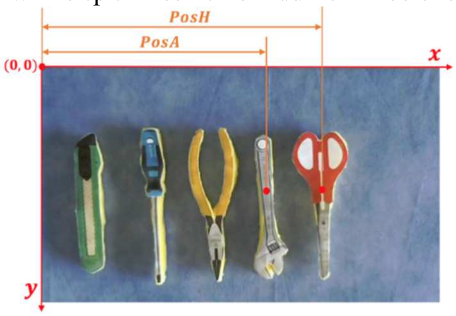


Fig. 5 Distances PosA and PosH for the tool sorting.

D. *Step 3: Detecting tools on the hand*

In step 3, a Haar classifier was used to detect tools delivered by the user. For this case, the database consisted of a series of images of users' hands holding any of the five tools and the manipulator trying to reach it. Partial occlusions were presented on some of the tool images, as shown in the left of Fig. 6. At the right, an example of detection of the five tools is shown using the Haar classifier, where it is possible to observe the behavior of said classifier facing different types of occlusions.

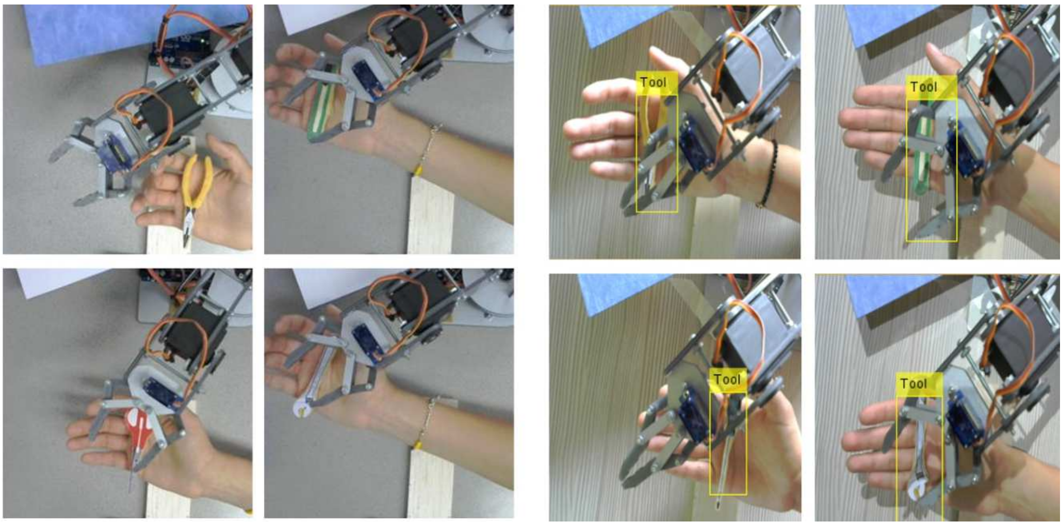


Fig. 6 Database and results for the Haar classifier.

The objective of the classifier is to detect any tool held by the user on the hand, even though the manipulator partially covers it during the gripping process. To do this, a Haar classifier was trained with the characteristics of Table 1, reaching 96% detection for the five tools, managing to overcome the occlusions presented by the robot. Once the classifier has found a tool, the manipulator moves to follow it until the end effector is on it, then the robot goes down, takes the object and takes it to the table to re-execute step 2, and put the new tool at the end of the row of elements already organized. The process is repeated until the five tools are arranged on the table, at which time the program ends.

TABLE I
CLASSIFICATION IN THE TEST DATASET

Variable	Value
Positive images	2500
Negative images	5620
Cascading classifiers	80
True Positive Rate	0.9705
Negative Samples Factor	0.7
Window size	[65 15]

III. RESULTS AND DISCUSSIONS

Initially, it was tried to train a Haar classifier for each tool, using the same database that was used for the classifier of step 3 but dividing it into five categories, one per tool. By training and testing each network independently, they were able to classify the five tools. However, the classifiers tended to confuse and classify the same tool in more than one category, as shown in the left of Fig. 7, or to detect tools where there are none, as in the example of the right of Fig. 7.

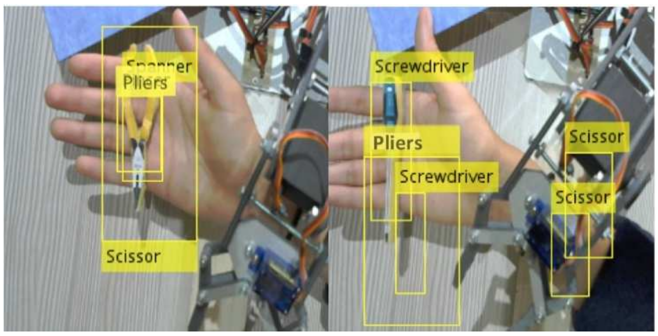


Fig. 7 Multiple categories for the same tool and detection of tools where there are not.

Due to the difficulties presented for the classification of tools using Haar classifiers and the fact that several of these classifiers are capable of detecting, to a certain extent, the same tool within the workspace despite classifying it erroneously, it was decided to train a single classifier with all the tools, and use it as a detector, achieving the results.

On the other hand, to avoid that, the classifier tended to confuse the robot with a tool. The labels of each category were made in such a way that only the most visible section of the tool was covered, as in Fig. 7 (left). It was visible despite the occlusions as in Fig. 7 (right). A certain degree of occlusion was accepted inside the label to encompass a greater area of vision of the tool. This type of label allowed reducing the number of detection boxes on the robot and increasing the tool recognition capacity, either totally or partially, as previously presented in Fig. 8 for pliers and spanners cases.

On the other hand, when applying the Haar classifier in the algorithm, it was observed that the detection of the tool, although accurate, was very variable because the detection box continuously changed in size. When the robot approached the tool, the box tended to move slightly towards the robot as if trying to cover it, as shown in Fig. 8. The robot could not easily find the grip point of the tool when it was close to it.

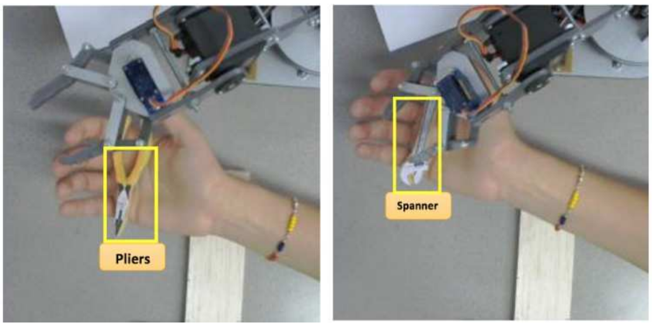


Fig. 8 Labels for the detection of tools with occlusions.

However, once the manipulator reached the center of the detection box, it stopped and went down to take the tool, and then locate it next to the others, as shown in Fig. 9.

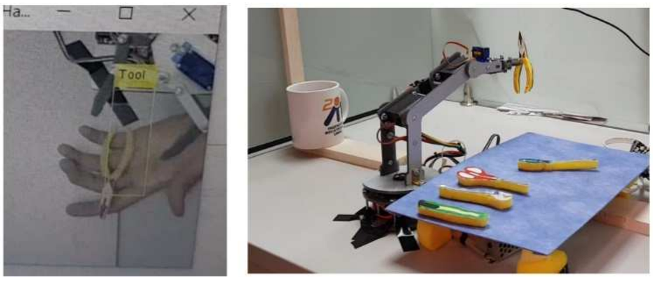


Fig. 9 Grip of the new tool delivered by the user.

For the detector and classifier by Faster R-CNN used in the recognition of tools on the table. It was possible to locate, with some difficulties, the objects to be sorted, reaching results such as those shown in Fig. 10. The classification of each element in the upper or lower part of the detection box is indicated. The degree of classification of each tool, from zero to one, i.e. with what confidence was made the detection and classification.

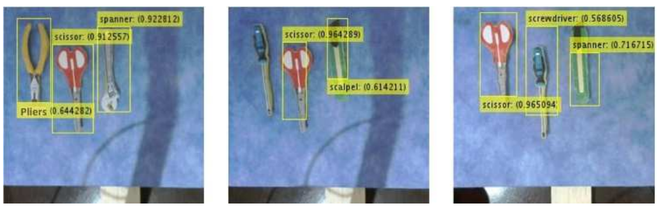


Fig. 10 Detection of tools on the table using Faster R-CNN.

As can be seen in Fig. 9 (left), it was possible to detect and correctly classify the tools, without the complications presented in the Haar classifiers, such as false positives or multiple classifications for the same tool. However, due to the low percentage of accuracy of the network, there were some complications for the scalpel and screwdriver tools detection, as shown in Fig. 9 (middle), where it was not possible to detect the screwdriver located on the table, and in Fig. 9 (right) where the scalpel was erroneously classified.

For the implementation in a real environment of the ordering application, it was necessary to condition the work area to the requirements of the simulation, that is, to establish certain distances between the cameras and the robot so that the distances obtained in simulation correspond to those in the physical environment. From this, the working environment was obtained, where the table on which the tools are arranged is on the right side of the robot, while the empty space on the left corresponds to the area where the new tool delivered by the user's hand is sought.

As can be seen, the manipulator does not have a degree of freedom that allows rotating the clamp to adjust the orientation of each tool before ordering it on the table. For that reason, all tools tilted, as shown in Fig. 11.

Fig. 11 Ordered tools probe.

To properly perform the sorting process, the manipulator was moved in such a way that each time it was to pick up or leave a tool on the table. It will be located first at the X, Y coordinates of the tool, at a height between the floor and the final effector of 15cm, to then descend in a straight line to the object. It can be taken and to go up in a straight line in order to avoid pushing the nearby tools with which are holding or sorting.

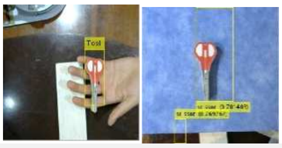
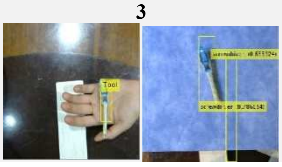
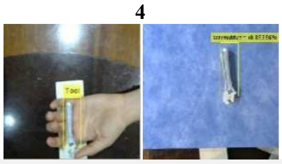
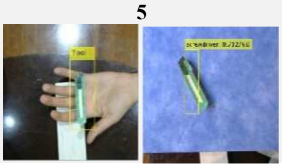
A comparison was made between the execution times of the two detection techniques used: Haar and Faster R-CNN classifiers, as shown in Table 2, in order to compare the results and determine which of them requires shorter processing time and, therefore, can be used in situations of detection of objects with sudden changes of position in short periods of time, in continuous video capture. Table 2 presents 5 different tests, one for each tool, and the time in milliseconds required by each method, where it can be seen a great difference between them, for any of the situations shown.

From the results of Table 2, it is possible to determine that the Haar classifiers generate an object detection much faster than the Faster R-CNN, using times less than 10ms, for each frame, while the Faster takes approximately 1s, which inhibits it from detecting sudden changes of the image during video shots, generated by sudden movements of the hand, which is why it was decided to use Haar classifiers for this section of the program, and the Faster only for the recognition of objects on the table. Starting from the 10º of inclination, the network begins to generate more than one detection frame on the tool, and from 30º, 3 boxes appear on the object, each with the same classification.

On the other hand, the behavior of each detection and classification method was evaluate facing situations such as inclination changes in the tool between 0º and 40º. Faster or different percentages of occlusion between 20% and 90%. Haar classifiers, starting from the 10º of inclination, the network begins to generate more than one detection frame on the tool, and from 30º, 3 boxes appear on the object, each with the same classification.

These results allow observing that, to obtain a total and not partial detection, tools with inclinations lower than 10º must be located to ensure that the dimensions of the box coincide with those of the object and its location generate the point of grip.

TABLE II
EXECUTION TIMES

Proof Number	Haar Time (ms)	Faster R-CNN Time (ms)
1 	5.174	1039.544
2 	5.096	1654.138
3 	4.704	986.699
4 	4.958	976.158
5 	4.967	987.120
AVERAGE	**4.980**	**1128.732**

The Haar classifier can support up to 50% occlusion on the tool during the tracking process, implying that the user must ensure that the robot does not cover the object above that percentage to ensure tracking. To obtain better results in the physical application of the algorithm, it is necessary to use a manipulator with an additional degree of freedom in the gripper, in charge of rotating the tool, to equalize the rotation of the physical tool to the simulated one.

This algorithm can be conditioned to other manipulators and workspaces. However, it would require modifying, inside the program, the dimensions of the manipulator, its inverse kinematics (only if these changes concerning the current manipulator), and the conversion of pixels from a distance, since the measurements would be completely different from those used in this work. However, the logic of operation would remain unchanged, allowing the program to function properly under the new working conditions.

Nevertheless, it was possible to generate a contribution to the automated processes of manipulation of multiple elements with the development of an algorithm that allows ordering and ensuring. In a precise way, the distance between different types of tools, and generates a constant ordering regardless of the number of elements that are added to the system or that already exist in the workspace. Therefore, the application can be extended to a greater number of tools according to the dimensions of each of them, the space available to order them, the dimensions of the manipulator, and the trained classifier.

IV. CONCLUSION

Faster RCNN is a recent technique in pattern recognition, and it was exposed that with an accuracy relatively low of 70.8%, it was possible to identify a group of five objects. Integrating this network with a Haar classifier can object recognition with occlusion of 50% in an effective way. The average processing time, including the Haar and the Faster R-CNN task, was around 1.7 seconds. That is enough for real-time applications with robots in human assistance o collaborative tasks.

Improve the Faster R-CNN network with a more extensive database does not improve the occlusion detection for the Haar classifier; instead, it can increase the processing time, affecting the real-time application. The capability of the Haar classifier to detect elements is robust enough to tolerate occlusions of up to 50% on the object. However, more than one classifier to detect and classify objects presents great drawbacks, such as the multiple classifications of categories for a single element or the detection of objects in areas of the image with only a background. Therefore, the best way to use this classifier is with its detection capacity, leaving a single classifier for the execution of the algorithm instead of five.

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-2290 and titled "Prototype of robot assistance for surgery".

REFERENCES

[1] C. S. Franklin, E. G. Dominguez, J. D. Fryman, M. L. Lewandowski, "Collaborative robotics: New era of human–robot cooperation in the workplace", *Journal of Safety Research*, Volume 74, ISSN 0022-4375, Pages 153-160, 2020, DOI: 10.1016/j.jsr.2020.06.013.

[2] N. Abdo, C. Stachniss, L. Spinello, W. Burgard. "Robots organize my shelves! Tidying up objects by predicting user preferences", *IEEE International Conference on Robotics and Automation (ICRA),* pp. 1557-1564 2015. IEEE. WA, USA.

[3] E. Magrini, F. Ferraguti, A. Jacopo, F. Pini, A. De Luca, F. Leali, "Human-robot coexistence and interaction in open industrial cells". *Robotics and Computer-Integrated Manufacturing*, Volume 61, pp 1-19, 2020.

[4] E.De Coninck, T.Verbelen, P.Van Molle, P.Simoens, B. Dhoed, "Learning robots to grasp by demonstration", *Robotics and Autonomous Systems*, Volume 127, 2020.

[5] H. Yang, R. Liu, S. Kumara, "Self-organizing network modelling of 3D objects", *CIRP Annals*, Volume 69, Issue 1, pp 409-412, ISSN 0007-8506, 2020. DOI: 10.1016/j.cirp.2020.04.099.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. *Advances in neural information processing systems*. pp. 1097-1105, 2012.
[7] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, J. Schmidhuber. “Flexible, high performance convolutional neural networks for image classification”. *In International Joint Conference on Artificial Intelligence (IJCAI)*. pp. 1237-1242. Catalonia, Spain. 2011.
[8] A. Mangal, H. Malik, G. Aggarwal, “An Efficient Convolutional Neural Network Approach for Facial Recognition”, *10th International Conference on Cloud Computing, Data Science & Engineering* (Confluence). pp. 817-822. Noida, India. 2020.
[9] M. Taskiran, N. Kahraman, C. E. Erdem, “Face recognition: Past, present and future (a review)”, *Digital Signal Processing*, Volume 106, 102809, ISSN 1051-2004, 2020, DOI: 10.1016/j.dsp.2020.102809.
[10] N. Hassan, K. W. Ming and C. K. Wah, "A Comparative Study on HSV-based and Deep Learning-based Object Detection Algorithms for Pedestrian Traffic Light Signal Recognition," *3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, Singapore, pp. 71-76, 2020.
[11] J. Lee, J. Chung, K. Sohn, “Reinforcement Learning for Joint Control of Traffic Signals in a Transportation Network”. *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1375-1387. 2020.
[12] J. Sharad, S. Suraj, K. Nitin. “First steps toward CNN based source classification of document images shared over messaging app”. *Signal Processing: Image Communication*. Volume 78. pp 32-412019.
[13] D. Ciregan, U. Meier, J. Schmidhuber. “Multi-column deep neural networks for image classification”, In *2012 IEEE conference on Computer vision and pattern recognition (CVPR)*, pp. 3642-3649. IEEE. Providence, Rhode Island, USA. 2012.
[14] C. G. Pachón-Suescún, J. O. Pinzón-Arenas, R. Jiménez-Moreno,"Fruit Identification and Quality Detection by Means of DAG-CNN," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 5, pp. 2183-2188, 2020. DOI: 10.18517/ijaseit.10.5.8684.
[15] I. Gangopadhyay, A. Chatterjee, I. Das., “Face detection and recognition using haar classifier and LBP histogram”, *International Journal of advanced research in computer science*, vol. 9, no 2, pp 592-598, 2018.
[16] P. C. Useche, R. Jimenez-Moreno, J. O. Pinzon. “Comparison between CNN and Haar classifiers for surgical instrumentation classification”. *Contemporary Engineering Sciences*, vol. 10, no. 28. pp. 1351-1363, 2017.
[17] J. O. Pinzon, R. Jimenez-Moreno, “Comparison between handwritten word and speech record in real-time using CNN architectures”. *International Journal of Electrical and Computer Engineering (IJECE)*. Vol 10, No 4 , pp. 4313-4321, 2020.
[18] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”. *Advances in neural information processing systems*, pp. 91-99, 2015.
[19] S. Kermani, M. G. Oghli, A. Mohammadzadeh, R. Kafieh, “NF-RCNN: Heart localization and right ventricle wall motion abnormality detection in cardiac MRI”. *Physica Medica*, Volume 70, pp 65-74, 2020. DOI: 10.1016/j.ejmp.2020.01.011.
[20] L. Jiao, S. Dong, S. Zhang, C. Xie, H. Wang. AF-“RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection”. *Computers and Electronics in Agriculture*, Volume 174, 105522, 2020. DOI: 10.1016/j.compag.2020.105522.
[21] F. Liu, G. Li, S. Yang, W. Yan, G. He, L. Lin, “Detection of heterogeneity on multi-spectral transmission image based on multiple types of pseudo-color maps”. *Infrared Physics & Technology*, Volume 106, 2020, DOI: 10.1016/j.infrared.2020.103285.
[22] M.D. Zeiler, R. Fergus, “Visualizing and Understanding Convolutional Networks”. *In: European Conference on Computer Vision. (ECCV). Lecture Notes in Computer Science*, vol 8689, pp 818-833. Zurich, Switzerland, 2014.
[23] S. Wan, S. Goudos, “Faster R-CNN for multi-class fruit detection using a robotic vision system”, *Computer Networks*, Volume 168, 107036, ISSN 1389-1286, 2020. DOI: 10.1016/j.comnet.2019.107036.