Algorithm of detection, classification and gripping of occluded objects by CNN techniques and Haar classifiers

Paula Useche[1], Robinson Jimenez-Moreno[2], Javier Martínez Baquero[3]
[1,2]Department of Mechatronic Engineering, Militar Nueva Granada Military University, Colombia
[3]Department of Electronic Engineering, De los Llanos University, Colombia

Article Info

Article history:

Received Nov 4, 2019
Revised Mar 4, 2020
Accepted Mar 21, 2020

Keywords:

DAG-CNN
Fast R-CNN
Haar classifiers
Machine vision
Manipulator robot
Occluded objects
VRML

ABSTRACT

The following paper presents the development of an algorithm, in charge of detecting, classifying and grabbing occluded objects, using artificial intelligence techniques, machine vision for the recognition of the environment, an anthropomorphic manipulator for the manipulation of the elements. 5 types of tools were used for their detection and classification, where the user selects one of them, so that the program searches for it in the work environment and delivers it in a specific area, overcoming difficulties such as occlusions of up to 70%. These tools were classified using two CNN (convolutional neural network) type networks, a fast R-CNN (fast region-based CNN) for the detection and classification of occlusions, and a DAG-CNN (directed acyclic graph-CNN) for the classification tools. Furthermore, a Haar classifier was trained in order to compare its ability to recognize occlusions with respect to the fast R-CNN. Fast R-CNN and DAG-CNN achieved 70.9% and 96.2% accuracy, respectively, Haar classifiers with about 50% accuracy, and an accuracy of grip and delivery of occluded objects of 90% in the application, was achieved.

Corresponding Author:

Robinson Jimenez-Moreno,
Department of Mechatronic Engineering,
Militar Nueva Granada Military University,
Carrera 11 # 101-80, Bogotá, Colombia.
Email: robinson.jimenez@unimilitar.edu.co

1. INTRODUCTION

Working conditions that occur in an unknown physical environment have led robotics to the need to increase the independence of robots to perform tasks that overcome various difficulties not foreseen in the environment, such as the presence of occlusions during the track of a trajectory for a manipulator robot [1], or various obstacles that move around the environment, where the manipulator must rethink its trajectory when said obstacle is too close to its structure, to avoid a collision [2]. However, the presence of obstacles in the path of a manipulator is not the only unforeseen event that must be overcome, but occlusions in the elements to be fastened as mentioned in [3], where an RGB-D sensor was used in the gripper of the manipulator to generate a 3D voxel map of the environment, which allows to recognize the occluded objects and generate a trajectory for its subjection without collisions. Kinect sensors is an RGB-D camera used for control robots [4, 5]

In works such as those presented in [6] and [7], algorithms focused on the occlusion detection process have been developed, to use this information in the movement of manipulator and mobile robots, where [6] performed an algorithm of occlusion edge detection, using CNN with RGB, RGB-D, and RGB-D-UV input images and videos, where D represents the depth and UV horizontal and vertical components of the optical flow field, thus allowing differentiation between occlusion edges, of the appearance edges of the desired elements, while in [7], an algorithm was made for the detection of multiple people in a real-world environment, using thermal and depth information, to determine the position of each person from their thermal detection, and define the presence of occlusions according to the results of the depth information, to generate a free path of obstacles for a mobile robot.

CNN's are designed to perform recognition of desired patterns and characteristics in images, to classify them into a specific category, as mentioned in [8, 9]. The CNN is composed of convolutional filters whose parameters are trained with a database that has images or references of the patterns to be classified, which allow the network to learn and extract the most relevant characteristics of an image, whose information is used to generate a classification, as explained in [10], but not only images like is exposed in [11].

Some of the recently developed CNN applications are: 1) a 34-layer CNN led to the detection of a wide range of cardiac arrhythmias from electrocardiograms, whose performance exceeded the average results of Medical prediction, made by a group of 6 cardiologists [12]; 2) a trained cascade of CNNs to detect and classify faces in a real environment, overcoming difficulties such as pose changes, expression, and lighting, avoiding becoming computationally expensive [13]; 3) expanded use of CNNs towards a three-dimensional environment, where volumes of magnetic resonance voxel of the prostate were evaluated, generating a segmentation of the entire volume, using only a fraction of processing time compared to other methods previously used [14].

Besides CNN, other artificial intelligence methods have been developed for the classification of patterns, such as the fast R-CNN [15] and the DAG-CNN [16, 17], in the first case is former a stage of extraction of a Regions of Interest (ROIs) that is responsible for of detecting desired elements in the input image, extracting them and entering them into a CNN for classification, as explained in [18], while the last consists of a branched structure where each branch contains a sequence of convolutional layers whose filters vary of dimension, to extract characteristics of greater and smaller size of the input image, and in the end to unify the results to give a classification, as indicated in [19].

Some examples of applications for the fast R-CNN are reported in [20] and [21]. The first report applied the Faster R-CNN for face detection and classification, to achieve a higher processing speed concerning other methods of deep learning, and the second report a multi-class fruit detection using a robotic vision system based on Faster RCNN. On the other hand, the DAG-CNN has been used in applications such as those described in [22] and [23], where the first publication used the DAG-CNN for the estimation of age in people of different genders and ethnicities, taking advantage of the extraction of characteristics at multiple scales of said network, with an accuracy of around 80%, and the second publication used the DAG-CNN for the classification of heartbeats from electrocardiogram images (ECG), achieving the identification of 15 different signals with 97.15% accuracy.

Apart from the CNNs, there are other methods of recognizing elements in images, such as Haar classifiers, which train a series of weak cascading classifiers, which receive an input image and through a sliding window they discard those areas that do not contain the desired element until, in the end, the windows indicating the object of interest, are obtained, [24]. Examples of application of Haar classifiers are reported in [25] and [26], where the first one carried out a process of real-time detection of cow nipples to generate an automatic milking system, while the second developed a method of counting and detecting the number of books stacked on shelves, where 96% recognition was achieved when testing the system on a total of 20 shelves and 233 books.

For the following article, a system for detecting, classifying and grabbing occluded objects was designed using a manipulator robot in a physical and virtual environment, where artificial intelligence techniques were used for the process of detection and classification of elements in the work area, and a Kinect V1 was used as the machine vision system. This work generates a fundamental contribution for the increase of the automation of robotized processes within unknown environments, where a sequence of detection and elimination of occlusions was established since it is not possible to dodge them employing obstacle avoidance algorithms, but they must be removed to be able to hold the desired object.

The present article is divided into 4 main sections. In the first section, a brief introduction was made. In the second section, the operation of the algorithm proposed for the elimination of occlusions is explained. In the third section, carried out on the algorithm in a physical environment, to obtain the quality of recognition and delivery of desired elements, in the presence of partial occlusions. Finally, in the last section, a series of conclusions were established regarding the operation of the algorithm, according to the results obtained during the tests.

2. RESEARCH METHOD

The following is the basic operation of the program for detecting, classifying and grabbing occluded objects, using artificial intelligence techniques, and captured three-dimensional information by Kinect V1 (see [27]). For this, section 2 was divided into 2 subsections, the first raises the physical working conditions for the implementation of the application, the second describes the algorithm’s flow.

The developed algorithm captures an RGB image of the work environment, then acquires the three-dimensional information of the environment to detect the height of each element concerning the table, and according to the disposition of the found objects, a grip algorithm is generated that allows to remove the tools using a manipulator robot and relocate them in the delivery area. The sequence of operation of the algorithm is shown in the flowchart of Figure 1, where each step of the program was marked with numbers from 1 to 5, which are described in the following subsections. Next, the operation of the algorithm is explained, the steps in Figure 1 are described, and its application in the physical and virtual environment is shown.

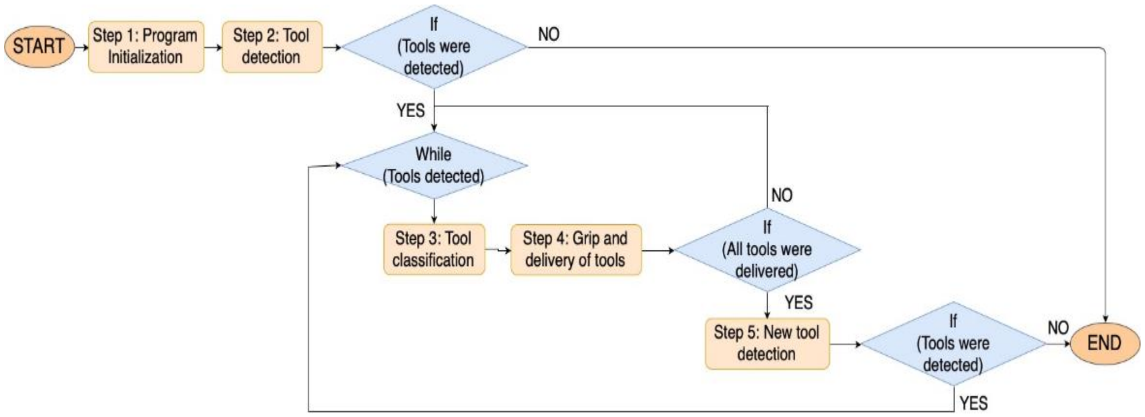

START
Step 1: Program Initialization
Step 2: Tool detection
If (Tools were detected)
NO
YES
While (Tools detected)
NO
Step 3: Tool classification
Step 4: Grip and delivery of tools
If (All tools were delivered)
YES
Step 5: New tool detection
If (Tools were detected)
NO
END
YES

Figure 1. Algorithm flow chart

2.1. Methods and materials

A physical work environment of 30x20cm area was established, at a height of 6cm from the ground, with a blue hue like the one shown in Figure 2(a). A Kinect V1 was used to capture the work environment and was located at a height of 98cm from the ground, in the center of the work area table or Grip Zone. A Delivery Zone was established, where two regions were demarcated to organize the tools, one for the Occlusions and the other for the Desired Objects, as shown in Figure 2(b). The robot used is shown in Figure 2(c) and has 3 DOF (Degrees of Freedom), all rotational. Its schematic representation is shown in Figure 2(d), where the angles of rotation were marked, the joints from 1 to 3 were listed, and the links from e1 to e3 were marked.

Five types of tools were selected for the application: Scalpel, Scissor, Screwdriver, Spanner, and Pliers, where each one was designed with a light and soft material (foam) that facilitates the adjustment of the gripper on the elements and reduces the influence of the weight of the objects on the dynamics of the manipulator. Each object was manufactured with 1cm thickness and maximum dimensions of 3.5x10 cm, and for each category, the shape and color of the tools were varied, in order to teach the program to recognize each element with tolerance for variations.

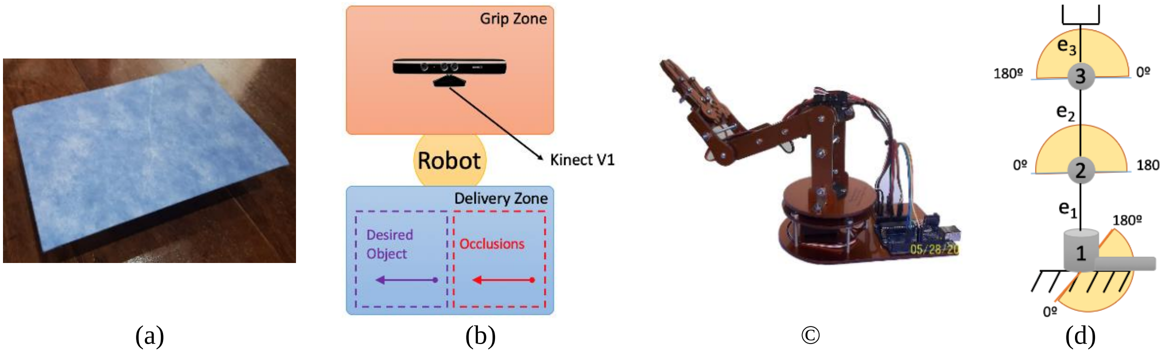


Figure 2. (a) Worktable, (b) grip and delivery zones of elements, (c) anthropomorphic robot, and (d) schematic representation of the robot

2.2. Algorithm flow

Step 1 program initialization: The first step consists of the initialization of the variables to be used in the course of the application, where imagD was defined as the vector that contains the number of pixels occupied by the Grip Zone in the RGB image of the Kinect ([128 174] pixels), and the PFH variable stores two matrices, each with 10 rows and 3 columns, where the first matrix saves the final positions for the desired objects, and the second saves the final positions for the unwanted elements (occlusions).

Step 2 tool detection: In this step, the first detection of tools in the environment is performed, where it is determined whether the program is executed or terminated according to what is identified in the Grip Zone as shown in Figure 1. Two methods of object detection, the fast R-CNN, and the Haar classifiers, were trained in order to compare the accuracy and precision of detection of each, and thus define which of the two is used in the algorithm. A database of 450 photos was established that were augmented with the Data Augmentation program developed in [28], reaching a total of 3150 images, where 2800 were used for training and 350 for testing. The detection boxes for both methods were defined as 55x28 pixels since that is the maximum value of the space occupied by the tools in the image.

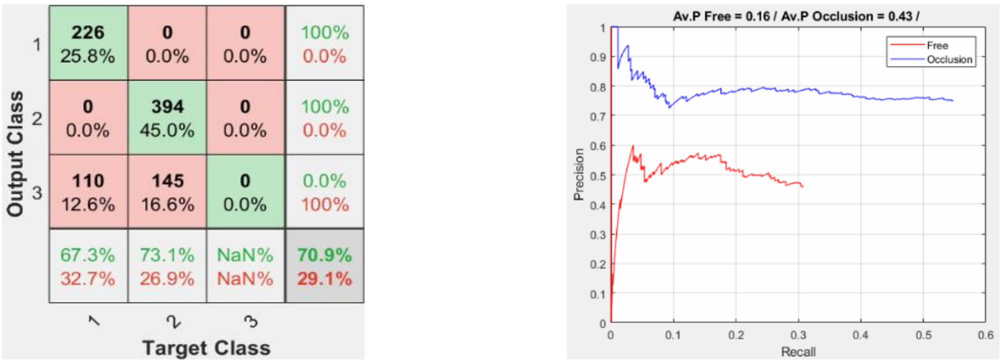
The fast R-CNN is a CNN-type neural network that has an ROI extraction stage designed for the detection of elements in images, as detailed in [18]. The architecture of this network is defined in the same way as that of a conventional CNN, using convolutional layers (CV), rectified linear units (RLU), maxpool (MP), average pool, batch normalization (B), fully connected (FC), dropout (DO), softmax (SOFT), among others, explained in [10]. The trained fast R-CNN has the architecture shown in Table 1, where it was observed that, when using rectangular filters, 4 convolution layers, 300 training times and a MiniBatchSize less than 30, the highest percentage of accuracy was achieved, receiving as input an image of image dimensions ([128 174] pixels).

Table 1. Architecture of the fast R-CNN for the detection and classification of occluded tools

Layers	CV+B+RLU	CV+B+RLU	MP	CV+B+RLU	CV+B+RLU	FC1+RLU+DO	FC2+RLU+DO
Filter Size	6x4	5x3	**2x3**	4x3	4x3	--	--
Stride	1	1	**2**	1	1	--	--
Number of Filters	16	128	--	256	512	--	--

The results of the trained fast R-CNN are shown in Figure 3, where Figure 3(a) shows the confusion matrix and Figure 3(b) shows the recall vs precision graph, with 1 being the free category, 2 being the Occlusion category, and 3 the Background. The results obtained in the fast R-CNN were not ideal, however, it was decided to use this network, since the structure proposed for the algorithm allows in step 5 to recognize the tools that were not captured in Step 2, reducing the influence of the low accuracy of the Fast R-CNN in the application, as described later in section 3.

For the training of the Haar classifiers, the values of a series of parameters that determine the basic training characteristics, explained in [29], were adjusted. A Haar classifier was trained for the recognition of the free category and another for the Occlusion category, whose accuracy and training parameters are shown in Table 2, where the highest accuracy obtained was 42% and 48%, respectively. As explained in [22], Haar classifiers need larger databases to obtain better detection results, however, the objective is to compare Haar classifiers with fast R-CNN, the reason why the database was not increased and was used in both cases.



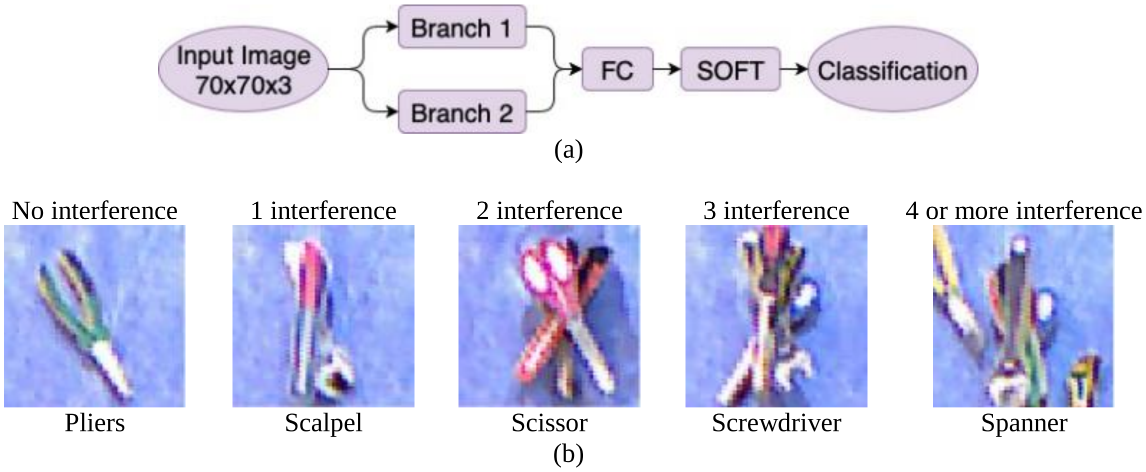
(a) (b)

Figure 3. Fast R-CNN: (a) confusion matrix and (b) recall vs precision

Table 2. Training parameters of Haar classifiers

Parameter	Accuracy	False Alarm Rate	Number Cascade Stages	Negative Samples Factor	True Positive Rate	Total Positives Samples	Total Negative Samples	Negative Samples	Number Positive Samples
Free	42%	0.01	15	3	60%	2800	5000	1272	424
Occ	48%	0.01	100	3	60%	2800	5000	204	68

Step 3 tool classification: After the tool detection process, the recognized elements were extracted from the image to classify them by means of a DAG-CNN in one of the 5 trained categories: Scalpel, Scissor, Screwdriver, Spanner, Scissor and the result was compared with the desired object by the user, in order to determine if it belongs to the group of desired objects or occlusions. A classification box of 70x70 pixels was defined, the dimensions of which were taken from the upper left corner of the detection box generated in Step 2 or Step 5, and 70 pixels to the right and 70 pixels down were captured. In Figure 4(a), the structure used for the DAG-CNN of the application is shown, where a division of 2 branches that receive the input image was generated and finally joined in a FC to classify, and in Figure 4(b) shows an example of the database used for network training.



(a)

(b)

Figure 4. (a) Structure of and (b) database for the DAG-CNN

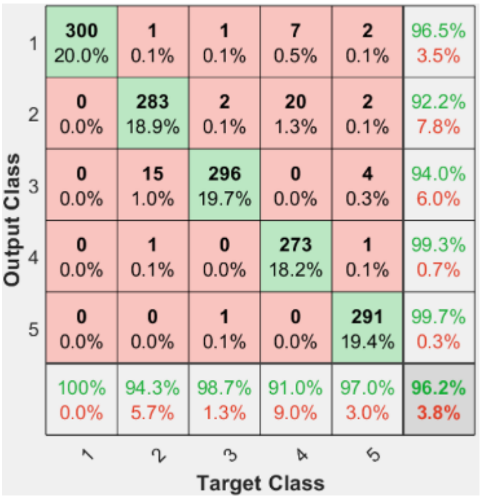
Table 3 reflected the architecture of the DAG-CNN, which was trained for 90 epochs, with a database of 3000 training images and 300 test images, per category. Figure 5 shows the confusion matrix of the DAG-CNN, which reached 96.2% accuracy, with a minimum of 91% accuracy per category. The numbers correspond to the classification categories, where 1 is pliers, 2 is scalpel, 3 is scissor, 4 is screwdriver, and 5 is spanner. For example, the classification of Scalpel was one of the categories that presented more difficulties in recognition, confusing itself with the Pliers category mainly.

Table 3. Architecture of the DAG-CNN (a) layers branch 1

	CV+B +RLU	MP	CV+B +RLU	MP	CV+B +RLU	MP	CV+B +RLU	FC1+RLU+DO	FC2+RLU+DO
Filter Size	4x4	**3x3**	3x3	**3x3**	3x3	**3x3**	3x3	--	--
Stride	1	**2**	1	**2**	1	**2**	1	--	--
Number of Filters	16	--	128	--	256	--	512	--	--

Table 3. Architecture of the DAG-CNN (b) layers branch 2

	CV+B +RLU	MP	CV+B +RLU	MP	CV+B +RLU	MP	FC1+RLU+DO	FC2+RLU+DO
Filter Size	6x6	**3x3**	4x4	**3x3**	3x3	**2x2**	--	--
Stride	1	**2**	1	**2**	1	**2**	--	--
Number of Filters	16	--	256	--	512	--	--	--



Output Class \ Target Class	1	2	3	4	5	
1	300 20.0%	1 0.1%	1 0.1%	7 0.5%	2 0.1%	96.5% 3.5%
2	0 0.0%	283 18.9%	2 0.1%	20 1.3%	2 0.1%	92.2% 7.8%
3	0 0.0%	15 1.0%	296 19.7%	0 0.0%	4 0.3%	94.0% 6.0%
4	0 0.0%	1 0.1%	0 0.0%	273 18.2%	1 0.1%	99.3% 0.7%
5	0 0.0%	0 0.0%	1 0.1%	0 0.0%	291 19.4%	99.7% 0.3%
	100% 0.0%	94.3% 5.7%	98.7% 1.3%	91.0% 9.0%	97.0% 3.0%	96.2% 3.8%

Figure 5. Confusion matrix for the DAG-CNN

Step 4 grip and delivery of tools: Once the algorithm defines whether the tool is a desired object or an occlusion, the movement of the manipulator robot for the transfer of the element to the delivery zone is executed. The center of the detection box of the tool classified as the holding point was selected, and the data of the PFH variable was used to determine the delivery coordinates of the element. To detect the clamping height of each tool concerning the table (Z coordinate) it was necessary to evaluate the three-dimensional information of each detection box and calculate its average. The algorithm first removes the elements classified as free, and then those defined as occlusion, changing the detection box only when all the elements of the tool stack have been removed, which means that the calculated average height corresponds to the height of the table.

Step 5 new tool detection: Before terminating the program, Step 5 was added, to verify the presence of tools in the Grip Zone. This step is responsible for repeating the process of detection of Step 2, through which the existence of new elements in the environment is defined, and according to the results obtained, Step 3 and Step 4 are executed, or the program is terminated, as shown in Figure 1.

3. RESULTS AND ANALYSIS

The first comparison between the methods of detection and classification of occlusions mentioned in section 2, where the percentages of accuracy, where the fast R-CNN exceeded the Haar classifiers by 20% accuracy. Additionally, a comparison was made between both methods applying them to the physical work environment, with different percentages of occlusion. The fast R-CNN managed to recognize a greater number of tools in the environment, and categorized them more accurately than the Haar classifiers, in addition to presenting greater accuracy in the detection boxes. For these reasons, it was decided to use the fast R-CNN in the algorithm.

The algorithm was applied in a previous simulation environment was tested by comparing the number of desired tools moved to the delivery zone, with respect to the total number of desired tools present in the Grip Zone, the results of which are shown in Table 4, where the column “Delivered tools” shows the amount of Desired objects delivered (numerator) with respect to the total number of desired objects present in the environment (denominator) for two fastening tests per tool, and the column “Percentage” shows the percentages of successful deliveries for each case according to the results of the column “Delivered tools”, in which the number of elements, their positions, the occlusion percentages, and the objects desired by the user was varied.

According to the results of Table 4, the algorithm reached 80% accuracy in the process of detecting, classifying and grabbing occluded objects, within a simulated environment, which means that at least 4 of 5 desired objects will be delivered correctly. During the tests, it was possible to observe two factors that affected the accuracy of the algorithm: the low accuracy of the detection boxes generated by the fast R-CNN, and the results of Step 5, which demonstrated inability to recognize any element in the simulated environment, since there are no variations in the input image that allow new elements to be recognized.

Table 4. Quality of delivery of desired objects of the simulated algorithm

Desired objects	Delivered tools		Percentage		Average
Screwdriver	3/4	3/3	75%	100%	**80%**
Scalpel	3/5	2/2	60%	100%	
Scissor	4/5	4/5	80%	80%	
Pliers	5/5	1/2	100%	50%	
Spanner	3/4	4/5	75%	80%	

3.1. Functioning in the physical environment

To determine the performance of the algorithm within a physical environment, two premises were established, the first focused on determining the percentage of tool stacks detected with respect to the total amount of tool towers present in the Grip Zone (Detection Quality), and the second defined the percentage of tools held correctly, with respect to the total amount of elements present in the environment (Holding quality). Table 5 reflected the results obtained for both premises, where the “Stacked Tools” row indicated the number of tools present in a stack of elements, and in the “Reinforcement Detection” row the percentage of recognized tool stacks was recorded in the environment after executing Step 5 once.

Tests were carried out with stacks of elements that have between 1 and 5 tools, where 10 tests were made for each quantity of stacked elements and varied from their positions to their orientations. The best results of detection and clamping of tools were obtained for piles of elements between 1 and 3 tools, where their average precision ranged between 80% and 90% while when increasing to 4 and 5 tools, the percentage dropped to almost a 60%. The detection failures were because, when the height of the stack of objects increases, they get too close to the Kinect’s camera, causing them to take up a lot of space in the image and the detection box fails to recognize them due to their dimensions, while the clamping faults were due to the fact that, when calculating the height of the element, the heights of the lower tools of the stack are detected, causing the average to be altered and less than the desired tool. On the other hand, the reinforcement detection proved to generate a 15.5% improvement in the detection quality of the program, reaching a 90% detection of elements with a single execution of Step 5, which demonstrates the algorithm's ability to recognize new elements after the first delivery of tools, and the ability to deliver them almost entirely.

Table 5. Quality of the algorithm applied to a physical environment

Stacked Tools	1	2	3	4	5	1 to 3 stacked objects	4 to 5 stacked objects	General average
Detection Quality (%)	71.43	100	78.57	64.29	60.31	**83.33**	**62.3**	**74.92**
Holding Quality (%)	100	79.17	85.19	88.89	77.78	**88.12**	**83.34**	**86.21**
Reinforcement Detection (%)	**90.48**							

4. CONCLUSION

The algorithm of detection, classification and tool grip proposed, successfully delivered 80% of the virtual tools, and 90% of tools in the physical environment, identifying the presence of occlusions and classifying up to 5 different types of elements. These results demonstrate the ability of the algorithm to hold and deliver the tools desired by the user overcoming difficulties such as occlusions of one and more elements. Step 5 allowed to improve the ability to recognize elements in the physical environment by more than 10% with a single execution, however, it did not generate any effect on the simulated environment, because the input image in Step 5 does not show large variations with respect to that of Step 2, which avoids a new detection of tools. The calculation of the height of each tool with respect to the table was affected by the presence of other elements in the detection box, resulting in a holding quality of 86%. Finally, a comparison between two detection methods was achieved, such as the Haar classifiers and the fast R-CNN, where the second presented a 20% improvement in the detection accuracy with respect to the first, in addition to demonstrating a better ability to recognize elements and classify them correctly as Free or Occlusion.

ACKNOWLEDGEMENTS

The authors are grateful to the Universidad Militar Nueva Granada, which through it’s Vice rectory for researchs finances the present project with code IMP-ING-2290.

REFERENCES

[1] M. Benzaoui, *et al.*, “Trajectory tracking with obstacle avoidance of redundant manipulator based on fuzzy inference systems,” *Neurocomputing*, vol. 196, pp. 23-30, 2016.

[2] D. Han, *et al.*, “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 98-104, 2018.

[3] G. Kahn, *et al.*, “Active exploration using trajectory optimization for robotic grasping in the presence of occlusions,” in *2015 IEEE International Conference on Robotics and Automation, ICRA*, 2015, pp. 4783-4790, doi: 10.1109/ICRA.2015.7139864.

[4] Q. Chang and Z. Xiong, “Vision-aware target recognition towards autonomous robot by Kinect sensors,” *Signal Processing: Image Communication*, 2020. Doi: 10.1016/j.image.2020.115810.

[5] X. Li, “Human–robot interaction based on gesture and movement recognition,” *Signal Processing: Image Communication*, vol. 81, 2020. Doi: 10.1016/j.image.2019.115686.

[6] S. Sarkar, *et al.*, “Deep learning for automated occlusion edge detection in RGB-D frames,” *Journal of Signal Processing Systems*, vol. 88, no. 2, pp. 205-217, 2017.

[7] H. S. Hadi, *et al.*, “Fusion of thermal and depth images for occlusion handling for human detection from mobile robot,” in *2015 10th Asian Control Conference (ASCC)*, pp. 1-5, 2015. Doi: 10.1109/ASCC.2015.7244722

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[9] D. Konstantinidis, *et al.*, “A modular CNN-based building detector for remote sensing images,” *Computer Networks*, vol. 168, 2020. Doi: 10.1016/j.comnet.2019.107034.

[10] C. C. Aggarwal, “Neural networks and deep learning,” Berlin, Germany: *Springer*, 2018.

[11] J. O. Pinzón-Arenas and R. Jiménez-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.

[12] P. Rajpurkar, *et al.*, “Cardiologist-level arrhythmia detection with convolutional neural networks,” *arXiv preprint* arXiv:1707.01836, 2017.

[13] H. Li, *et al.*, “A convolutional neural network cascade for face detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5325-5334, 2015.

[14] F. Milletari, N. Navab, and S. A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565-571, 2016. Doi: 10.1109/3DV.2016.79.

[15] J. O. Pinzón-Arenas, R. Jiménez-Moreno, and César G. Pachón-Suescún, “ResSeg: Residual encoder-decoder convolutional neural network for food segmentation,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 1017-1026, 2020.

[16] J. O. Pinzón-Arenas, R. Jiménez-Moreno, and César G Pachón-Suescún, “Offline signature verification using DAG-CNN,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 3314-3322, 2019.

[17] J. O. Pinzón-Arenas and R. Jiménez-Moreno, “Object sorting in an extended work area using collaborative robotics and DAG-CNN,” *ARPN Journal of Engineering and Applied Sciences*, vol. 15, no. 2, 2020.

[18] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE international conference on computer visión*, 2015, pp. 1440-1448.

[19] S. Yang and D. Ramanan, “Multi-scale recognition with DAG-CNNs,” in *Proceedings of the IEEE international conference on computer visión*, pp. 1215-1223, 2015.

[20] H. Jiang and E. Learned-Miller, “Face detection with the faster R-CNN,” in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 650-657, 2017. Doi: 10.1109/FG.2017.82

[21] S. Wan and S, Goudos, “Faster R-CNN for multi-class fruit detection using a robotic vision system,” *Computer Networks*, vol. 168, 2020. Doi: 10.1016/j.comnet.2019.107036.

[22] S. Taheri and Ö. Toygar, “On the use of DAG-CNN architecture for age estimation with multi-stage features fusion,” *Neurocomputing*, vol. 329, pp. 300-310, 2019. Doi: 10.1016/j.neucom.2018.10.071

[23] Z. Golrizkhatami, S. Taheri, and A. Acan, “Multi-scale features for heartbeat classification using directed acyclic graph CNN,” *Applied Artificial Intelligence*, vol. 32, no. 7-8, pp. 613-628, 2018.

[24] M. G. Krishna and A. Srinivasulu, “Face detection system on AdaBoost algorithm using Haar classifiers,” *International Journal of Modern Engineering Research*, vol. 2, no. 5, pp. 3556-3560, 2012.

[25] A. Rastogi, A. Pal, and B. S. Ryuh, “Real-time teat detection using haar cascade classifier in smart automatic milking system,” in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 74-79, 2017. Doi: 10.1109/ICCSCE.2017.8284383.

[26] A. B. Kanburoglu and F. B. Tek, “A Haar Classifier Based Call Number Detection and Counting Method for Library Books,” *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, pp. 504-508, 2018.

[27] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE multimedia*, vol. 19, no. 2, pp. 4-10, 2012.

[28] P. C. U. Murillo, J. O. P. Arenas, and R. J. Moreno, “Implementation of a Data Augmentation Algorithm Validated by Means of the Accuracy of a Convolutional Neural Network,” *Journal of Engineering and Applied Sciences*, vol. 12, no. 20, pp. 5323-5331, 2017.

[29] MathWorks, “trainCascadeObjectDetector,” 2020. [Online] Available on: https://la.mathworks.com/help/vision/ref/traincascadeobjectdetector.html.

BIOGRAPHIES OF AUTHORS

Paula Useche Murillo is a Mechatronics Engineer graduated with honors in 2017 from the Nueva Granada Military University in Bogotá, Colombia, where she currently studies an M.Sc. in Mechatronics Engineering and works as a research assistant in the Mechatronics Engineering program.

Robinson Jiménez-Moreno is an electronic engineer graduated from the Francisco José de Caldas District University in 2002. He received an M.Sc. in Engineering from the National University of Colombia in 2012 and Ph.D in Engineering at the Francisco José de Caldas District University in 2018. His current research focuses on the use of convolutional neural networks for object recognition and image processing for robotic applications such as human-machine interaction.

Javier Eduardo Martinez Baquero is an Electronic Engineer graduated from University of the Llanos in 2002. Posgraduated in Electronic Instrumentation from Santo Tomas University in 2004 and M.Sc. in Educative Technology at Autonoma of Bucaramanga University in 2013. His current research focuses on Instrumentation, Automation and Control.