Iranian Journal of Electrical and Electronic Engineering. ADLEEE. (2024) 3331

IUST
Iran University of
Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Faster R-CNN and 3D reconstruction for handling tasks implementing a Scara robot

Julián Herrera-Benavidez*, Cesar G. Pachón-Suescún* and Robinson Jiménez-Moreno*(C.A.)

Abstract: This paper presents the design and results of using a deep learning algorithm for robotic manipulation in object handling tasks in a virtual industrial environment. The simulation tool used is V-REP and the environment corresponds to a production line based on a conveyor belt and a SCARA type robot manipulator. The main contribution of this work focuses on the integration of a depth camera located on the robot and the computation of the gripping coordinates by identifying and locating three different types of objects of interest with random locations on the conveyor belt, through a Faster R-CNN. The results show that the system manages to perform the indicated activities, obtaining a classification accuracy of 97.4% and a mean average precision of 0.93, which allowed a correct detection and manipulation of the objects.

Keywords: Faster R-CNN, Homogeneous Transformation Matrix, Point Cloud, V-REP.

1 Introduction

ADVANCES in artificial intelligence are making possible to address problems that involve the recognition of patterns [1]. Deep learning, through the use of convolution layers that abstract feature maps, has allowed creating networks that manage to develop a wide range of tasks, of which object recognition [2] and face detection [3] can highlight. Specifically, the convolutional neural network (CNN) [4] is widely used in classification tasks, while the faster R-CNN [5-7] addresses localization problems, which allows detecting different kinds of objects in the same image. Faster R-CNN architecture is of special interest since it has proven to be more efficient than its predecessors: R-CNN and fast R-CNN [8].

The scope that can be obtained through Deep Learning can be greater, for example, in [9], the use of a CNN with an encoder-decoder architecture is appreciated to find areas of interest in RGB-D images, for grasping objects in a given robot environment. The CNN-based architectures demonstrate to be able to approach segmentation problems [10], for example, in [11] a network called PoseCNN is developed, which estimates the position and orientation of the objects.

These investigations are important in fields such as robotics, since they allow the creation of developments that are applicable to object recognition and handling tasks [12], where an anthropomorphic robot and a Faster R-CNN are implemented to perform classification, where the creation of a point cloud in the environment is of special interest in this type of work. In [13], it can be seen a work that implements this tool in conjunction with a neural network based on a CNN to perform the control of a 6 DoF robot, the point cloud is found from an RGB-D camera, and through the depth image, it is possible to perform the inverse kinematics of the robot to hold the objects that have been pointed by a laser pointer. It is applicable in obstacle avoidance systems [14] and even in mapping tasks for mobile robots [15-17] and collaborative robot work [18][19].

As mentioned, the current state of the art is the use of depth cameras and deep learning algorithms for industrial applications in controlled environments with robot manipulators. However, when the positional reference of the object to be grasped by a robotic manipulator change, it is necessary to set the robot kinematics as a function of the camera coordinates in order to dynamically adjust the new motion angles for grasping. The integration of the coordinate equations and

Iranian Journal of Electrical & Electronic Engineering, 2024.
Paper first received 13 June 2024 and accepted 31 October 2024.
* The authors are with the Faculty of Engineering, Universidad Militar Nueva Granada, Bogotá, Colombia.
E-mail: robinson.jimenez@unimilitar.edu.co.
Corresponding Author: Robinson Jiménez-Moreno.

object localization using a faster RCNN with the adjustment of the robot position is the main contribution of this work, as it extends the contributions of the state of the art to semi-controlled environments and complements work such as that presented in [20]. For this work, a virtual environment (V-REP) and a RGB-D camera is used to create the point cloud used in the 3d reconstruction of the robot's working area. In addition, there is explained how the homogeneous transformation matrices are implemented in conjunction with the point cloud to calculate the inverse kinematics of the robot.

The first section of this paper presents the introduction. The second section presents the methodology with database and the architecture used for the neural network, as well as the model with which the camera is parameterized, and the point cloud of the environment is obtained. The third section presents the results and finally the conclusions.

2 Materials and Methods

In order to locate and arrange objects in a production line, using a Faster R-CNN and a depth camera, a virtual environment is created in V-REP and shown in Fig. 1. In this environment, it can be seen, the objects of interest, the Scara robot, the field of view of the camera and the image captured by it.

The network database comes from pictures taken through an RGB-D camera located in a link of the Scara robot. A generic depth camera provided by the simulator is used, of the Real sense type with a resolution capacity of 1280 X 720 and a deep field of view of 85.2 x 58. The objects of interest are in a conveyor belt that contains them in a disorderly manner, so that when the database is generated, random images are obtained.

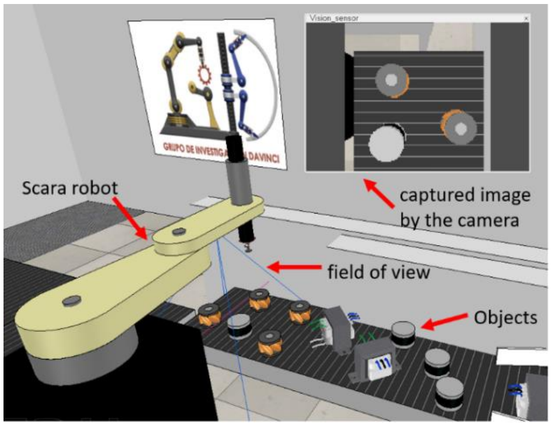


Fig. 1 Scene built in V-REP

The database consists of 1659 RGB images, an amount that is found experimentally, with a resolution of 128x128 pixels containing 3 classes of objects that have been manually labeled, which are transformer, sensor and wheel. The resolution chosen in the camera has the purpose of increasing the speed of communication between the V-REP environment and the software that indicates the orders, since the images must be exported. 90% of the images have been prepared for the network training phase and the remaining 10%, for the validation phase. A sample of the images labeled in the database is presented in Fig. 2.

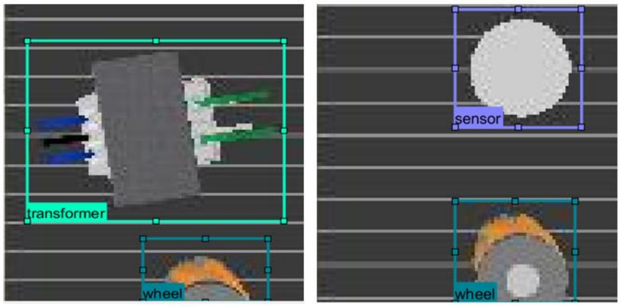


Fig. 2 Manual labeling of the images.

As mentioned above, the architecture used for classifying and locating objects is called Faster R-CNN. This architecture is explained in [21], where it is argued that its efficiency is since an external algorithm is not used to obtain the regions of interest; on the contrary, a network is trained that is totally focused on this task, called RPN, it shares the convolution layers with a Fast R-CNN, making the required processing less. Together the Fast R-CNN and the RPN form the Faster R-CNN. In Fig. 3, the architecture used can be seen.

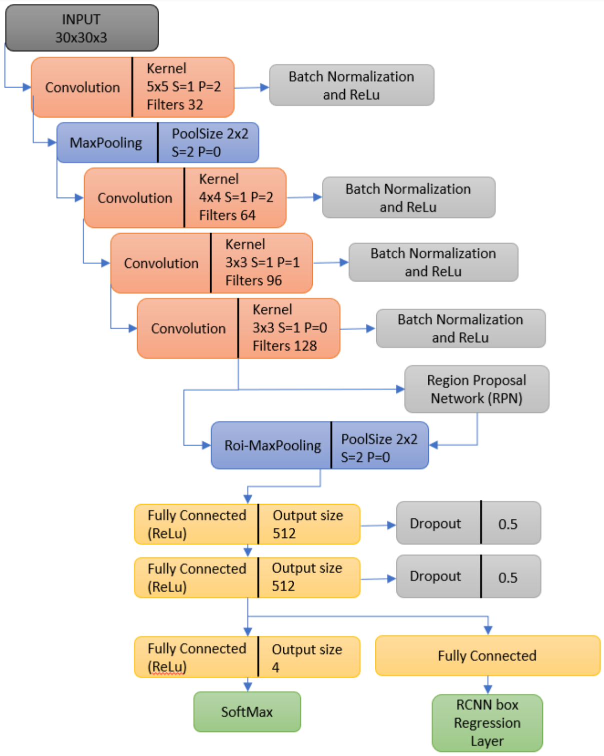
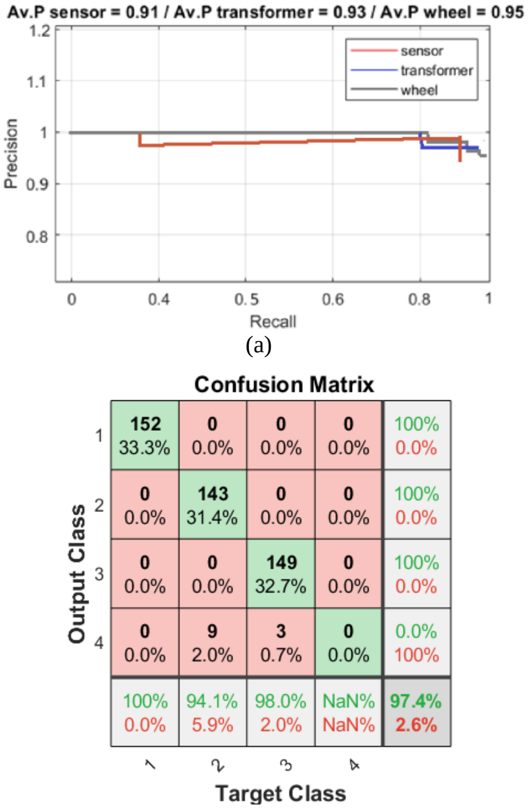

INPUT 30x30x3
Convolution
Kernel 5x5 S=1 P=2 Filters 32
Batch Normalization and ReLu
MaxPooling
PoolSize 2x2 S=2 P=0
Convolution
Kernel 4x4 S=1 P=2 Filters 64
Batch Normalization and ReLu
Convolution
Kernel 3x3 S=1 P=1 Filters 96
Batch Normalization and ReLu
Convolution
Kernel 3x3 S=1 P=0 Filters 128
Batch Normalization and ReLu
Region Proposal Network (RPN)
Roi-MaxPooling
PoolSize 2x2 S=2 P=0
Fully Connected (ReLu)
Output size 512
Dropout
0.5
Fully Connected (ReLu)
Output size 512
Dropout
0.5
Fully Connected (ReLu)
Output size 4
Fully Connected
SoftMax
RCNN box Regression Layer

Fig. 3 Faster R-CNN architecture.



(a)

Confusion Matrix

Output Class \ Target Class	1	2	3	4	
1	152 33.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	143 31.4%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	149 32.7%	0 0.0%	100% 0.0%
4	0 0.0%	9 2.0%	3 0.7%	0 0.0%	0.0% 100%
	100% 0.0%	94.1% 5.9%	98.0% 2.0%	NaN% NaN%	97.4% 2.6%

(b)

Fig. 4 (a) PR graph. (b) Confusion matrix

The inputs of the network are RGB images containing objects that must be within a region of not less than 30x30 pixels, a value that was determined after reviewing the smallest region of interest (RoI) in the database. This architecture is trained in four stages. The first one performs RPN training using a learning rate of 1x10^{-4}. The second stage uses the regions of interest produced by the RPN to train a Fast R-CNN with a learning rate of 1x10^{-4}. The third stage joins the convolution layers of the RPN and the Fast R-CNN to refine only the weights of the exclusive layers to the RPN with a learning rate of 1x10^{-5}. Finally, in the last stage the network is trained by refining only the exclusive layers to the Fast R-CNN with a learning rate of 1x10^{-5}, training was performed with 100 epochs, an iteration frequency of 1054 and an SDGM optimizer [21].

Once this is done, the performance of the network is evaluated using the PR (precision vs recall) graph and the confusion matrix that has been obtained with the test data (see Fig. 4). Fig. 4(a) shows that the accuracy approaches 1 at different recall levels, which indicates that the percentage of correct classifications made by the network has a good performance. The average accuracy obtained for each class is shown in the upper part of the graph. This value allows to see the ability of the network to find relevant objects and make correct classifications. Fig. 4(b) shows the confusion matrix, from which 97.4% of correct classifications are obtained for the test data base.

3 Point Cloud and Camera Model

The point cloud is a set of data that describes the position of a large number of points with respect to a coordinate system. These points allow the 3D reconstruction of the environment, so it is a means to know the position of the objects that have been detected by the Faster R-CNN. Depth information from the point cloud is used for camera calibration and spatial localization of the coordinates with respect to the robot. The network training is responsible for the two-dimensional localization and classification of the shape features of each object, working in conjunction with the depth for the robot's grasping and ordering tasks. In the network training for identification, the point cloud does not provide any information about the object type, except for the possible shape of the object, which by itself is not discriminative.

The objective of using this tool is to obtain the relative position of the elements with respect to the base coordinate system of the robot, and thus be able to calculate the inverse kinematics with which the gripper is moved to the target. To obtain this data, the intrinsic and extrinsic parameters of the RGB-D camera must be found, which describe how the objects are projected to form the captured photograph. In equations (1) to (3), this transformation is presented, which does not consider any distortion due to the lenses [22] [23].

$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \boldsymbol{K} \cdot \boldsymbol{H} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

$$\boldsymbol{K} = \begin{bmatrix} s_x & 0 & x_o \\ 0 & s_y & y_o \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_o & 0 \\ 0 & f_y & y_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{3}$$

In equation (1), the matrices involved in the transformation are shown. The matrix $\boldsymbol{K}$ is called intrinsic parameters of the camera, which depends on the distance of the focal axis f, scaling constants s_x and s_y that transform the units of distance to pixels, and translation constants x_0 and y_0 that measure the position of the optical axis of the camera with respect to the coordinate plane from which the pixels are measured. The matrix H is called the matrix of extrinsic parameters, this makes it possible to relate the coordinate axes of the camera with respect to a base plane, such as, for example, the base plane of the Scara robot. The homogeneous coordinates (*X*, *Y*, *Z*, *1*) locate a point with respect to the base plane of the robot, while the homogeneous coordinates (p_x, p_y,1) locate, in pixels, the projection of the point on the camera. A graphic explanation of this can be seen in Fig. 5.

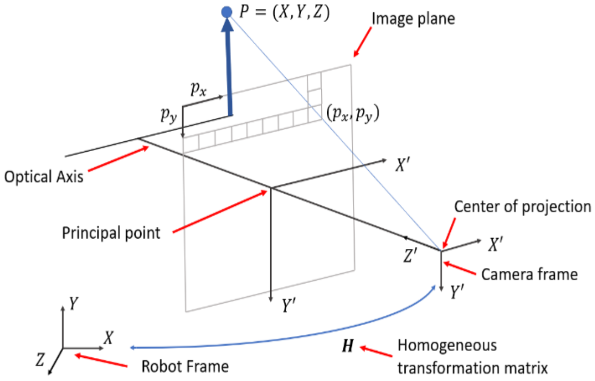


Fig. 5 Perspective projection according to the pinhole camera model.

The point *P*, which is located with respect to the coordinate system of the robot, is projected through a perspective projection towards the plane of the image. The projection results in pixels p_x and p_y, which store the captured RGB color. The depth channel considers the distance of said point from the plane of the camera. To find the coefficients that are in the intrinsic parameter matrix, the methods called "direct linear transformation" and "Zhang method" can be used [24], of which the last one is the most used due to its flexibility, since it only requires photographs where there should be a chessboard with known measurements. The images taken in V-REP for this purpose can be seen in Fig. 6. By this, it is possible to obtain the matrix of intrinsic parameters shown in Eq. (4).

$$\boldsymbol{K} = \begin{bmatrix} \frac{111.91pixel}{cm} & 0 & 64.5pixel & 0 \\ 0 & \frac{111.99pixel}{cm} & 64.6pixel & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (4)$$

The virtual robotic system for exercise monitoring allows the user to select the exercise to be performed, the number of repetitions and the rest time. The system tracks the sequence by counting both the repetitions and the rest time and only requires the use of a computer with a web camera where the designed virtual robot is installed. To calculate the point cloud with respect to the base plane of the camera (x', y', z'), Eq. (5) must be applied. This can be abstracted from Eq. (1), considering that it is treated with homogeneous coordinates.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_o \\ 0 & f_y & y_o \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} p_x \cdot Z' \\ p_y \cdot Z' \\ Z' \end{bmatrix} \qquad (5)$$

The value of Z' is given by the RGB-D camera in the depth channel and it must match the units of the f_x and f_y coefficients. It must be applied Eq. (5) for each pixel of the entire depth image and then graph the calculated points, which can have the color indicated by the RGB image. The environment in V-REP, the depth image and the point cloud can be seen in Fig. 7.

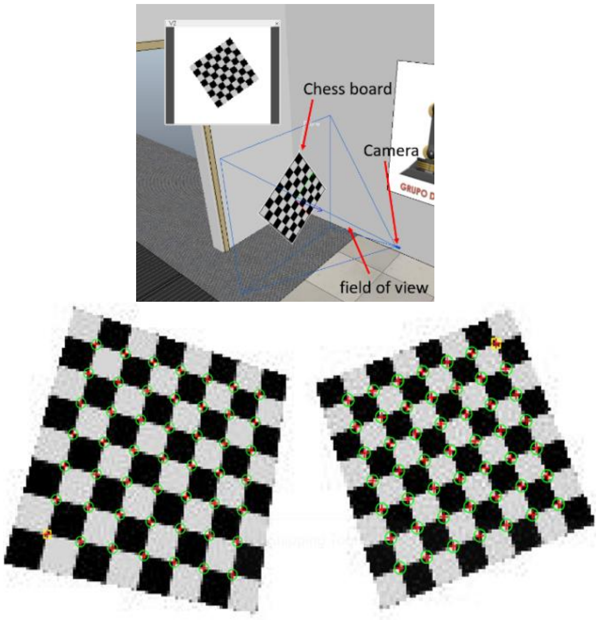
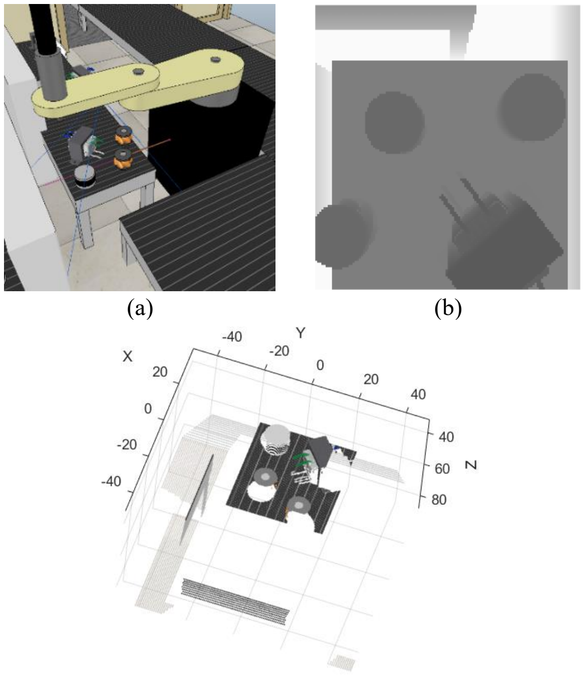


Fig. 6 50x50 cm chessboard used to calibrate the camera.


(a)

(b)

(c)

Fig. 7 (a) Environment in V-REP. (b) Depth image. (c) Point cloud.

With the information obtained by the point cloud and the Faster R-CNN, it is possible to obtain the position of the objects with respect to the camera. To find this information with respect to the plane of the robot from which the inverse kinematics is calculated, the homogenous transformation matrix ***H*** must be found, that is explained in [25]. The matrix ***H*** of Eq. (1) transforms the points of the plane (***x, y, z***) to the plane (***x', y', z'***), so in reality it is wanted to find the inverse of that matrix. Since the points with respect to the plane of the camera have already been obtained, they must be transformed to the plane of the robot. Fig. 8 shows the configuration of the robot and the location of the camera's plane.

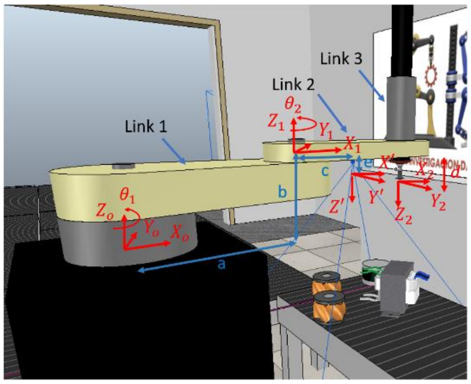


Fig. 8 Coordinated axes of the robot and the camera.

Fig. 8 allows to observe the coordinate axes of the 3 links that make up the robot. There are two rotational and one prismatic, so this robot is a Scara RRP. The number of generalized coordinates depends on the number of degrees of freedom, and these must be calculated to bring the gripper plane (x_2, y_2, z_2) to the detected object. Next, equations (6) to (9) present the generalized coordinates and the transformation matrices to obtain the ***H*** matrix.

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \\ d \end{bmatrix} \tag{6}$$

$$\boldsymbol{p_o} = \boldsymbol{T_1} \cdot \boldsymbol{T_2} \cdot \boldsymbol{p}_{(x',y',z')} = \boldsymbol{H^{-1}} \cdot \boldsymbol{p}_{(x',y',z')} \tag{7}$$

$$\boldsymbol{T_1} = \begin{bmatrix} c(\theta_1) & -s(\theta_1) & 0 & 0 \\ s(\theta_1) & c(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

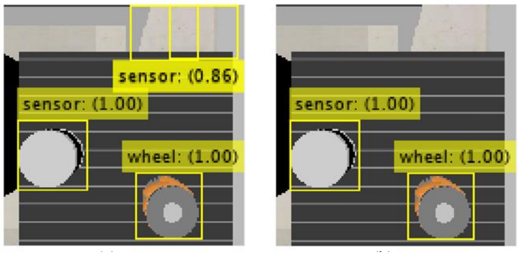
$$\boldsymbol{T_2} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -e \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\pi & -s\pi & 0 \\ 0 & s\pi & c\pi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Equation (6) contains the generalized coordinates that describe the robot. Equation (7) expresses how points of the camera plane (*x', y', z'*) are transformed to the base plane of the robot (*x, y, z*). Equation (8) contains the homogeneous matrices that relate the base plane of the robot with the base plane of the second link, which has a rotation around the axis and then a translation. Equation (9) expresses the homogeneous matrices that relate plane 1 to the plane of the camera, which uses a rotation around the axis, a translation and then a rotation of 180 degrees around the ***X*** axis.

4 Results and Discussion

Once the network is trained, its operation is validated, as it is seen in Fig. 9. The results demonstrate that no situation occurred in which there were errors in the classification and manipulation of objects, which demonstrates the robustness of the Faster R-CNN to solve problems related to robotics.

It can be seen in Fig. 9(a), that the network detects several regions of interest among which some are not on an object. This error is solved by selecting the most relevant results, i.e. the elements that have an accuracy in the classification lower than 90% are filtered and the remnant is considered as the result of the network, as seen in Fig. 9(b).



(a) (b)

Fig. 9 (a) Results of the Faster R-CNN. (b) Relevant results of the network.

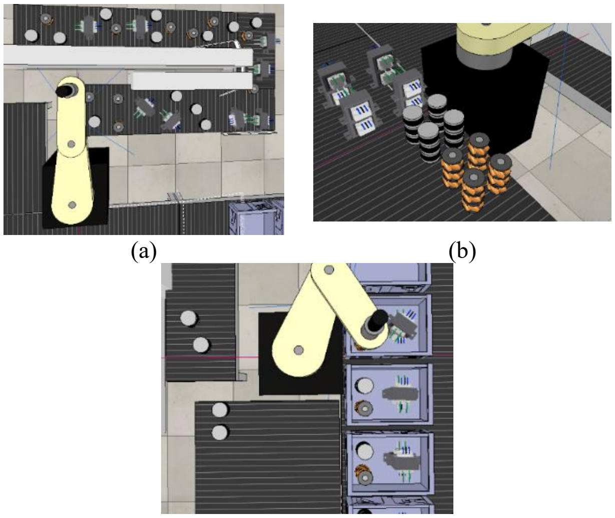
In Fig. 10, the objects in the main band with random positions can be seen; these are manipulated to perform 3 types of applications: the assembly of a structure (see Fig. 10a), the ordering of the products in a conveyor belt (see Fig. 10b) and the packaging of these (see Fig. 10c). The average time required to perform the classification and manipulation of the objects were 68 ms for the Faster R-CNN and 4.5s for gripping actions, using a computer with GPU Nvidia GTX 1050 with processor Inter® Core i7™ and 16 GB of RAM.

5 Conclusions

Applications that require locating and classifying objects within an RGB image can be perfectly addressed using a Faster R-CNN network. This can achieve high accuracies that make it reliable for jobs in the industry. In addition, its architecture allows to detect several objects in the same image with response times that are appropriate to include it in work applications in real time.

The data contained in the point cloud is ideal to work together with the results delivered by the faster R-CNN, this complements the information obtained regarding the location of the objects, allowing to perform manipulation work through fixed robots.

The use of generalized coordinates that describe the robot and the points of the camera plane, are necessary for the localization and grasping of objects in autonomous production systems, given the reference changes that the objects may undergo in relation to the camera position.



(a) (b)

(c)

Fig. 10 (a) Objects with random positions in the band. (b) Armed 3D structure. (c) Packaging of products.

Conflict of Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to "Iranian Journal of Electrical and Electronic Engineering".

Author Contributions

J. Herrera B: Conceptualization, Research and Investigation, Original Draft Preparation. **C. Pachón S:** Conceptualization, Research and Investigation, Original Draft Preparation. **R. Jimenez M**: Idea & Conceptualization, Revise and Editing.

Acknowledgment

The authors are grateful to Universidad Militar Nueva Granada for the funding of this project.

References

[1] V. Andrić, M. Gajić-Kvaščev, D. K. Crkvenjakov, M. Marić-Stojanović, S. Gadžurić, "Evaluation of pattern recognition techniques for the attribution of cultural heritage objects based on the qualitative XRF data", *Microchemical Journal*, Vol. 167, 2021, doi:10.1016/j.microc.2021.106267.

[2] V. N. Sichkar and A. V. Lyamin, "Design of Deep CNN Model for Effective Traffic Signs Recognition," *2021 International Russian Automation Conference* (RusAutoCon), pp. 367-373, 2021, doi:10.1109/RusAutoCon52004.2021.9537445.

[3] P. J. Lu and J. -H. Chuang, "Fusion of Multi-Intensity Image for Deep Learning-Based Human and Face Detection," *IEEE Access*, vol. 10, pp. 8816-8823, 2022, doi: 10.1109/ACCESS.2022.3143536.

[4] C. -W. Hsu, Y. -H. Huang and N. -F. Huang, "Real-time Dragonfruit's Ripeness Classification System with Edge Computing Based on Convolution Neural Network," *2022 International Conference on Information Networking (ICOIN)*, pp. 177-182, 2022, doi:10.1109/ICOIN53446.2022.9687292.

[5] R. Gonzales-Martínez, J. Machacuay, P. Rotta and C. Chinguel, "Hyperparameters Tuning of Faster R-CNN Deep Learning Transfer for Persistent Object Detection in Radar Images," I*EEE Latin America Transactions*, vol. 20, no. 4, pp. 677-685, April 2022, doi: 10.1109/TLA.2022.9675474.

[6] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015, doi: 10.1109/ICCV.2015.169.

[7] K. S. Htet and M. M. Sein, "Event Analysis for Vehicle Classification using Fast RCNN," *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, pp. 403-404, 2020, doi:10.1109/GCCE50665.2020.9291978

[8] H. Tahir, M. Shahbaz Khan and M. Owais Tariq, "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles," *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 587-594, 2021, doi:10.1109/ICCCIS51004.2021.9397079.

[9] A. Nguyen, D. Kanoulas, D. Caldwell, N. Tsagarakis, Detecting object affordances with convolutional neural networks, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2765-2770, 2016.

[10] C. Zhang, W. S. Jiang, Y. Zhang, W. Wang, Q. Zhao and C. J. Wang, "Transformer and CNN Hybrid Deep Neural Network for Semantic Segmentation of Very-high-resolution Remote Sensing Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp. 1-20, 2022, doi: 10.1109/TGRS.2022.3144894.

[11] X.Yu, T. Schmidt, V. Narayanan and D. Fox, Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes, 2018, doi:10.48550/arXiv.1711.00199.

[12] Y. Yu, Z. Cao, Z. Liu, W. Geng, J. Yu and W. Zhang, "A Two-Stream CNN With Simultaneous Detection and Segmentation for Robotic Grasping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 1167-1181, Feb. 2022, doi:10.1109/TSMC.2020.3018757.

[13] M. Zhong, Y. Zhang, X. Yang, Y. Yao, J. Guo, Y. Wang, and Y. Liu, Assistive Grasping Based on Laser-point Detection with Application to Wheelchair-mounted Robotic Arms, *Sensors*, Vol. 19, n° 2, 303, 2019, doi:10.3390/s19020303.

[14] F. Flacco, T. Kröger, A. De Luca, O. Khatib, A depth space approach to human-robot collision avoidance, *IEEE Robotics and Automation (ICRA)*, pp. 338-345, 2012, doi:10.1109/ICRA.2012.6225245.

[15] Z. Yu, U. L. S. Perera, H. Hauser, P. R. N. Childs and T. Nanayakkara, "A Tapered Whisker-Based Physical Reservoir Computing System for Mobile Robot Terrain Identification in Unstructured Environments," *IEEE Robotics and Automation Letters*, 2022. doi:10.1109/LRA.2022.3146602.

[16] C. Yang, Q. Chen, Y. Yang, J. Zhang, M. Wu and K. Mei, "SDF-SLAM: A Deep Learning based Highly Accurate SLAM using Monocular Camera aiming at Indoor Map Reconstruction with Semantic and Depth Fusion," *IEEE Access*, 2022. doi:10.1109/ACCESS.2022.3144845.

[17] P. Opaspilai, S. Vongbunyong and A. Dheeravongkit, "Robotic System for Depalletization of Pharmaceutical Products with 3D Camera," 2*5th International Computer Science and Engineering Conference (ICSEC)*, pp. 422-427, 2021, doi:10.1109/ICSEC53205.2021.9684575.

[18] M. Dhuheir, E. Baccour, A. Erbad, S. Sabeeh and M. Hamdi, "Efficient Real-Time Image Recognition Using Collaborative Swarm of UAVs and Convolutional Networks," 2*021 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1954-1959, 2021, doi:10.1109/IWCMC51323.2021.9498967.

[19] Y. Chen, W. Wang, V. Krovi and Y. Jia, "Enabling Robot to Assist Human in Collaborative Assembly using Convolutional Neural Networks," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11167-11172, 2020, doi:10.1109/IROS45743.2020.9340735.

[20] M. Abdalsalam Rasheed, W. M. Jasim, R. Nori Farhan, Enhancing robotic grasping with attention mechanism and advanced UNet architectures in generative grasping convolutional neural networks, *Alexandria Engineering Journal*, Vol. 102, pp. 149-158, 2024, doi:10.1016/j.aej.2024.05.082.

[21] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, n° 6, pp. 1137-1149, 2016, doi:10.1109/TPAMI.2016.2577031

[22] D. Shreiner, G. Sellers, J. Kessenich, B. Licea-Kane, OpenGL programming guide: The Official guide to learning OpenGL, (version 4.3. Addison-Wesley, 2013).

[23] G. Bradski, A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, (O'Reilly Media, Inc, 2008).

[24] Z. Zhang, A flexible new technique for camera calibration, *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, pp. 1330-1334, 2000.

[25] J. Craig, Introduction to robotics: mechanics and control, (USA:: Pearson/Prentice Hall, 2005, pp. 48-70).

Julian E. Herrera-Benavidez was born in Villavicencio, Colombia, in 1995. He received his degree in Mechatronics Engineering from the Pilot University of Colombia in 2018. He received his Master’s degree in Mechatronics Engineering from Nueva Granada Military.

Cesar G. Pachón-Suescún was born in Bogotá, Colombia, in 1996. He received his degree in Mechatronics Engineering from the Pilot University of Colombia in 2018. He received his Master’s degree in Mechatronics Engineering from Nueva Granada Military University.

Robinson Jiménez-Moreno is an Electronic Engineer graduated from Universidad Distrital Francisco José de Caldas in 2002. He received a M.Sc. in Engineering from Universidad Nacional de Colombia in 2012 and Ph.D. in Engineering at Universidad Distrital Francisco José de Caldas in 2018. His current working as associate professor of Universidad Militar Nueva Granada and research focuses on the use of convolutional neural networks for object recognition and image processing for robotic applications such as human-machine interaction.