Abstract

This paper presents the implementation of a simulation of a robotic arm whose task is to collect different objects in a virtual environment. To develop this task, the control of the robotic arm is done through 10 different hand gestures, which are recognized by a CNN with a structure type DAG Network (or DAG-CNN), reaching an accuracy of 84.5% in the recognition of gestures. Likewise, real-time tests are carried out on the already trained network, where the user is in a semicontrolled environment indicating the different actions for the robot to perform, where the correct operation of the trained network was verified, obtaining a high precision in the recognition of the commands made, that is, without errors in the control actions followed by the robot.

Keywords: Convolutional Neural Network, DAG Network, Hand Gesture Recognition, Virtual Environment, Robotic Arm Control, Inception architecture

1 Introduction

In recent years, a variety of techniques have been implemented to perform the remote control of different agents or devices for the execution of tasks, such as the control of wheelchairs using basic elements such as sensors located in gloves [1], the control of a robot specialized in surgery using haptic feedback techniques [2] or
the trajectory control of a remotely operated vehicle using neural network techniques [3]. Advancing to more complex models, there is the Deep Learning, from which base have been developed robust recognition techniques that are currently beginning to be used for the control of agents, being them mainly used in static applications. One of these techniques are the so-called Convolutional Neural Networks (CNN) [4], which are employed in the recognition of patterns in images, achieving very high precision in the recognition of characters made by hand or in document analysis [5], or even managing to discriminate up to 1000 different objects under a deep architecture [6]. Thanks to their great performance, these networks have begun to be used to interact with robots or in applications with moving elements, as can be seen in [7], where a mobile agent interacts with gestures of the hand, however, requires a very controlled environment and that the user wears a glove with a specific color, which causes variations in the agent's control performance if changes are made in the environment where it interacts.

Taking into account this, to achieve high accuracy with elements to recognize that they differ very little from each other and that, additionally, there are variations in the environment, such as lighting or noise depending on the device that captures the image, it is needed a greater depth in the network to learn more features belonging to each category. To compensate for an increase in linear depth in the general architecture of the network, there are developments such as the Inception architecture [8], which establishes a Directed Acyclic Graph (DAG) Network structure [9]. This type of architecture allows having a greater number of layers or depth in the network, without having to make it longer, improving the processing time and increasing the amount of features that the network can learn by allowing the input image to maintain a larger size through more layers of convolution.

The novelty of this work is found in the application of CNN with a DAG network structure (called in this work DAG-CNN) in the control of a mobile robotic arm in a virtual environment, which has as its task to collect several elements. Although CNNs have been used in interactions with mobile robots by means of hand gestures, as shown in [7], in this work it will be used to directly control the manipulator and, at the same time, show the degree of improvement using a DAG Network type architecture for the recognition of the 10 gestures to be used for control.

The paper is divided into 4 parts, where section 2 describes the virtual environment to be used, the proposed DAG-CNN architecture along with its training and validation, and the developed interface. Section 3 presents the results obtained with real-time tests. Finally, section 4 shows the conclusions obtained.

2 Methods and Materials

The implemented development was divided into 3 stages that allow the manipulation of the robotic agent using Deep Learning techniques. In the first place, a virtual environment is adapted in which it is going to work. Then, the training of
a DAG-CNN is performed, that, once trained, allows the user to control the manipulator independently, to make the grip of the element that the robot has reached. Finally, an interface is created by joining each of the aforementioned items to execute the operation of object collection and control of the mobile manipulator by the user, where it has an option called "Auto" to perform tests of the mobile agent and the environment for proper operation, and an option called "manual", allowing the user to have manual control over the manipulator. Each stage is described below.

**Virtual Environment**

The simulation environment to be used was developed under the VRML programming platform of MATLAB®. This consists of a 3D environment within which is the mobile robotic arm (see Figure 1b) of anthropomorphic type that will be controlled by the user, in the same way, there are 3 types of tools distributed on the floor of different tonality each, which are scissors (yellow), scalpels (cyan) and screwdrivers (red). These objects are recognized and located by a Faster R-CNN [10] so that the robot can move towards each one. To make use of the Faster R-CNN, a capture of the global camera of the surroundings is taken, obtaining an image of 700x525 pixels, which is entered into the network, obtaining as output boxes that enclose each element, allowing to know its position and what type of object it is, as shown in Figure 1c. Additionally, there are 3 boxes located on the left side (see Figure 1a) to locate each object.

**CNN Architecture**

For the implementation of the manual control of the robotic manipulator, a CNN is trained with 10 different hand gestures that can be seen in Figure 2. The gestures Art1_L and Art2_R allow control of rotation to the left and right, respectively, of the joint 1 of the arm. Art2_Down and Art2_Up, make the articulation 2 rotate downwards or upwards, and in the same way the Art3_Down and Art3_Up but with the articulation 3. The commands Grip_Cl, Grip_Op and Grip_Rot control the closing, opening and rotation of the end effector, respectively. Finally, the Stop gesture indicates the termination of the grip, i.e. when the user makes the gesture for 3 seconds, the arm will no longer be controlled by the user, and will take the object to its respective box. To perform the training, a database of 200 images per category is created, in other words, a total of 2000 images, where 90% is used for training and 10% for validation.
The neural network to be used corresponds to a DAG-CNN, i.e., a network that is subdivided into several paths, for the present case 2 paths, which do not incur a cycle but reach a final point. This type of network has the great advantage of increasing its depth without drastically increasing its computational cost by making it more "wide", helping not only to increase the details possible to learn, but also its accuracy. Taking this into account, the architecture shown in Figure 3 is set.

This architecture, as a standard CNN (with a configuration of 5 convolutions in line with maxpooling), is trained with the elaborated database. When comparing the two networks, the standard CNN obtained 71% accuracy with 700 epochs of training, while the DAG-CNN improved by 4% the accuracy with respect to the other network, obtaining 75% accuracy with 200 training epochs, which in terms of recognition, this difference marks a significant improvement, as well as in the computational cost required for their training, using fewer training epochs, than for this case, they were 500 fewer epochs. However, since the application requires more precision, the percentage obtained is not enough.
Based on this, a Data Augmentation algorithm of the database is applied based on the improvement of the toolbox presented in [11], using 2 kinds of filters: Increase and reduction of illumination (see Figure 4), so that, although most of the training images were made on a white background, do not depend on abrupt changes of lighting. In essence, it does not require a totally controlled environment, but can be independent in the light that it has. In this way, it is possible to obtain a database of 2000 images per category, obtaining a total of 20,000 images, of which 18,000 are used for training and 2,000 for validation.
Figure 4: Samples of variation in the brightness, where the middle image is the original one.

As a result of using this new database, the network increased more than 9% in its accuracy using the same number of epochs to train, even having 10 times more validation images, as evidenced in Figure 5, where the overall percentage was reduced by the high misclassification of the Stop gesture. This is mainly because the category Art2_Up has a great resemblance to the Stop gesture, with only one difference in that the first has a raised finger, which led to erroneously classify a quarter of the Stop gesture images within this category, especially in images where the gesture is made at a distance far from the camera.

Figure 5: Confusion Matrix obtained from the CNN trained, where category 1 = Art1_L / 2 = Art1_R / 3 = Art2_Down / 4 = Art2_Up / 5 = Art3_Down / 6 = Art3_Up / 7 = Grip_Cl / 8 = Grip_Op / 9 = Grip_Rot / 10 = Stop.

To better understand the behavior of this type of network, in Figure 6 it can be seen the activations that each convolution has in the two paths of the architecture. In this
way, it is verified that in each path the convolution layers are able to learn different characteristics of the gesture, for example, in the layer 1 of the left path, the focus is on the texture of the hand, while that of the right path focuses mainly on the lower contour of the hand and the index finger.

![Activations Left Path](image1)
![Activations Right Path](image2)

**Figure 6:** Activations of each path in the CNN type DAG. The activations are organized from left to right starting with the input image, followed by the convolution 1 to 5.

**User Interface**

The developed interface integrates the trained neural networks and the virtual environment, allowing the user to have a global view of the work area, a view of all the objects recognized in each snapshot, and three different views of the robot: Top, left side and right side, in such a way that the manual collection of the tool can be carried out in a simple way. Additionally, it can be selected between the "Auto" option to perform mobile agent operation tests and the "Manual" option to control the manipulator, causing the camera to be activated, which is shown in the box at the bottom called "Cam", to send the instructions to the robot by means of hand gestures. Also, it shows which object is closest to the robot, being the one that will be collected. The complete interface can be seen in Figure 7.

**3 Results and Discussions**

Control tests of the manipulator are made in real time, so that the user indicates the actions to be performed by the arm and verify that it responds according to the desired. On the other hand, the execution times of a standard architecture CNN and a CNN with type DAG network architecture will be compared.

To use it in "manual" mode, the user must have a webcam that allows him to send gestures made in real time, so that the robot executes the command once the gesture is recognized, the time of the movement being the same as the user making the gesture. The start of the process is done autonomously, that is, the algorithm locates
each of the objects and then the robot moves to said location, in order of proximity to the starting point of the mobile agent. Once the object to be collected is reached, the user begins to make control over the manipulator part. Figure 8 shows different actions performed by the user to grasp the element, where Figure 8a the manipulator is been positioning in the direction of the object, once it has been located, it is ready to make the grip by closing the clamp, as shown in Figure 8b. Once this is finished, the user keeps the "Stop" signal for 3 seconds (see Figure 8c), indicating that the process has finished and the robot takes the tool to the corresponding box. To complete the task, all the recognized elements must be collected, where the user will control the manipulator to make a correct grip of each one.

In this way, the correct functioning in real time of the trained network and of the virtual environment implemented is verified, where it is achieved the collection and sorting of the objects in a specific workspace being the manipulator controlled by a user.

To verify the time that the algorithm takes to recognize the gesture, the average times of execution of 20 recognition tests are taken, not only from the DAG-CNN, but also from the standard CNN trained, to observe the efficiency in terms of processing speed. Table 1 shows the times obtained in the two neural networks, where the standard CNN was approximately 2.8 times faster than that configured with the DAG architecture, however, the times are relatively insignificant in terms of the application used, since the generated delay was not perceptible during the tests carried out on the interface.
Table 1. Execution time of each CNN configuration

<table>
<thead>
<tr>
<th>DAG-CNN (5 Conv Layers x2 paths)</th>
<th>Standard CNN (5 Conv Layers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 ms</td>
<td>8.9 ms</td>
</tr>
</tbody>
</table>

Figure 8: Robotic arm control performed by the user
4 Conclusions

This work presents the design and simulation of a user interface, which contains a virtual environment for tool collection, which allows to execute different actions of a robotic arm to perform the grip of 3 types of these tools. In the development of the application, a DAG-CNN architecture was elaborated and trained, being used as control of the robotic arm by means of hand gestures, reaching 84.5% accuracy, on the other hand, during the real-time tests, each gesture made by the user was recognized without mistakes, allowing to demonstrate the accuracy and excellent performance of this type of neural network for the control of robots.

When making efficiency comparisons between a CNN with a standard configuration and one with a DAG architecture, the latter, although it had a higher execution time, its accuracy, being trained with the same database as the standard, obtained a 4% improvement, using 500 less training epochs, in other words, the CNN configured with a DAG architecture helps reduce the computational cost of their training and greatly improves the recognition of objects. Likewise, the use of data augmentation proved to be a crucial factor in improving the accuracy of the network, where using the same images, only by varying its illumination, an improvement of more than 9% was achieved, reaching to a sufficiently precise network for the application of grip robotic control in a virtual environment.

The high performance in the operation of the control of the manipulator by means of the gestures allows to extend the field of telecontrol of robots by another type of method, as well as the communication with autonomous agents or systems of virtual environments. Bearing this in mind, this work demonstrates the possibility of creating new configurations of convolutional neural network architectures, even improving the efficiency of these, which gives way to increase the complexity of the real applications in which they are used, improving more and more the accuracy of recognition of elements that have very similar features or characteristics.

Acknowledgments. The authors are grateful to the Nueva Granada Military University, which, through its Vice chancellor for research, finances the present project with code IMP-ING-2290 (2017-2018) and titled "Prototype of robot assistance for surgery", from which the present work is derived.

References


https://doi.org/10.1109/iros.2014.6943165


Received: February 28, 2018; Published: March 28, 2018