Journal of Engineering and Applied Sciences 13 (10): 3479-3492, 2018
ISSN: 1816-949X

Manipulation of Tools by Means of a Robotic Arm Using Artificial Intelligence

M. Paula Catalina Useche, Ruben D. Hernandez Beleno and Robinson Jimenez Moreno
Facultad de Ingenieria, Universidad Militar Nueva Granada, Bogota, Colombia

Abstract: The following study presents the development of an algorithm of recognition, grip detection and trajectory planning for a robot of three degrees of freedom where objects are recognized by Convolutional Neural Networks (CNN) and gripping detection by geometric analysis of the object. The algorithm works on a non-controlled environment where it receives the images through a webcam, segments all the objects that are found in them, classifies them into one of three categories of tools (scalpel, scissors, screwdriver) trained on the CNN and searches for the tool desired by the user on which a feasible gripping point is selected and a path is executed that allows the manipulator to take the found object and move it to another point. Finally, functional tests are presented for the trained categories and the results are analyzed to determine grip accuracy in the real environment.

Key words: Deep convolutional neural network, hand gesture recognition, layer activations, human-robot interaction, tool, real environment

INTRODUCTION

Nowadays, manipulator robots have been able to imitate human movement for the execution of different types of tasks whether repetitive, dangerous or high precision which in turn often require specialized systems for the execution of these tasks, however in spite of their high working capacities, these robots must be able to interact with their surroundings when it is not structured and for this it is necessary to imitate in the same way, the visual capacities of the human being as proposed by Pardo (2009).

To try to imitate the sense of sight, the robot must have the ability to differentiate and identify objects within its environment which involves performing processes of segmentation of images and individual detection of elements as mentioned (Pardo, 2009; Gil *et al*., 2004), so that, the robot can select the desired object and execute a sequence of movements to interact with it. This sequence is known as the trajectory of the manipulator which seeks to control the position and orientation of the object held either with a robot arm or with two as presented by Garvin *et al*. (1997).

The process of segmentation of objects is done by capturing an image of the environment where, either based on the RGB and HSV color information of the objects with respect to the environment as proposed by Gil *et al*. (2004) or by the contours of the elements as presented by Xu *et al*. (2007), it is a matter of differentiating each object from the background or the surface that supports it. For the case discussed by Stander *et al*. (1999), an additional phase is added to the segmentation process to reduce the influence of shadows in the detection of objects in order to avoid generating a wrong classification of a moving object and by Leibe *et al*. (2008) representations of previously learned object shapes are used with a probabilistic extension of the generalized hough transformation in conjunction with the segmentation in order to increase the probability of detection of the objects.

After solving the problem of segmentation, all objects found in the resaerch area must be recognized, so that, the manipulator is able to find the element on which to perform the action. Facedwith this problem, work has been carried out by Liang and Hu (2015), Yao and Miller (2015) with convolutional neural networks (Krizhevsky *et al*., 2012) to detect different types of objects using large databases of images for the training of networks. On the other hand by Young *et al*. (1997) a multilayer Hopfield neural network is used (Adler *et al*., 2017) where each layer learns a particular feature of the object and all together make the classification. While by Guevara *et al*. (2008), Haar cascade classifiers are used for the detection of faces and by Fergus *et al*. (2003) object recognition is done from disordered and non-segmented images where the detector must learn the invariant features of the image that do not depend on a scale factor such as shapes or geometries of parts of the image.

By Malpartida (2003), a particular case of an object recognition and manipulation algorithm is presented with a SCORBOT ER-manipulator where the processes

Corresponding Author: M. Paula Catalina Useche, Facultad de Ingenieria, Universidad Militar Nueva Granada, Bogota, Colombia

described above are developed to allow the manipulator to take a particular object such as a nut and move it from one point to another.

A tool recognition and grip algorithm is presented where it is used an object recognition method by means of convolutional neural networks and segmentation and geometric analysis techniques for calculating the grip based on the friction surface between the end effector and the element and the shortest possible gripping distance with respect to the centroid of the tool. Trajectory planning was performed at each point of the run for a manipulator of three degrees of freedom and tests were performed with several tools to verify the accuracy of the robotin the actual grip.

The algorithm developed offers a solution to the problem of detecting and gripping objects in an unsupervised environment where no element is identified by some type of marker, nor do they have in common basic geometries whose grip can be generalized in families of objects which makes the recognition of each element and the calculation of the grip unique for each of the detected tools and allows it to be used in an application with any type of object that has been previously trained in the convolutional neural network.

This study is divided as follows: first, the structure of the algorithm is presented, explaining the architecture and the basic operation of the algorithm, subsequently, it is analyzed the tests where the code is run for an non-controlled environment in order to observe the degree of accuracy in the grip, then the results are presented and analyzed and finally the conclusions.

MATERIALS AND METHODS

Algorithm structure: The developed algorithm allows to recognize a certain tool and transfer it from one point to another using a manipulator of three degrees of freedom. Segmentation and binarization techniques are used for the detection of each tool located in the work area and for the calculation of the position and inclination required in the gripper to make the final grip on the tool. It is used convolutional neural networks for the recognition of each of the tools, and trajectory planning for the control of the manipulator in the whole route.

The algorithm was divided into a series of successive steps that give as final result the position in millimeters and rotation in degrees of the gripper of the manipulator, to do the physical grip of the object selected by the user. Next, the five steps that make up the algorithm and their respective tasks are presented.

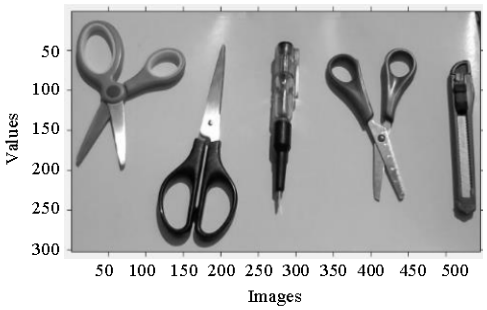


Fig. 1: Grayscale test image

Step 1; Tool segmentation: In the first place, the algorithm determines the quantity and location of each object in the work area to later identify each one. This first step is performed on a clear surface of a single hue, following the procedure described by Guacaneme (2012) where the surface average is obtained, subtracted with the original image and the result is binarized to perform the detection of each element.

On the grayscale image, each of the elements of the work area is segmented, defining as the background of the image all that soft change that presents the surface of the same and defining as object or tool all the sudden changes on said surface. As seen in Fig. 1 and 2, the surface on which the tools are located is of a single hue, while each element is of a darker color and generates a sudden change with respect to the background, causing peaks of intensity as shown in Fig. 2.

In Fig. 2b, the intensity changes of the input image are plotted in a three-dimensional form where each intensity change represents a change in the height of the pixel in the image. Figure 2a shows the surface of the image from the top view where it can be differentiated each of the tools in the resaerch area thanks to the change of tonality they have with respect to the table and Fig. 2b shows the side view of the surface where the background presents smooth changes throughout the image and the tools generate abrupt changes in the surface, causing lower peaks with darker colors than the bottom and higher peaks with the tool's metal shine.

From the subtraction of the background with the original grayscale image, a bottomless image is obtained, with the tools discriminated in white tones as in Fig. 3a. After differentiating the tools from the background, the resulting image is binarized and a final detection of objects is generated, resulting in the image of Fig. 3b, where each identified tool is delineated by a white line and filled with a certain color. In the case of Fig. 3b, 3 of

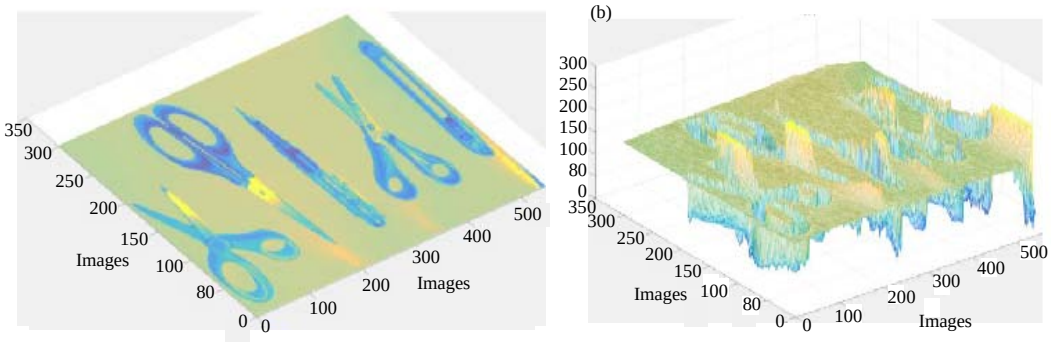


Fig. 2: Surface of the capture image

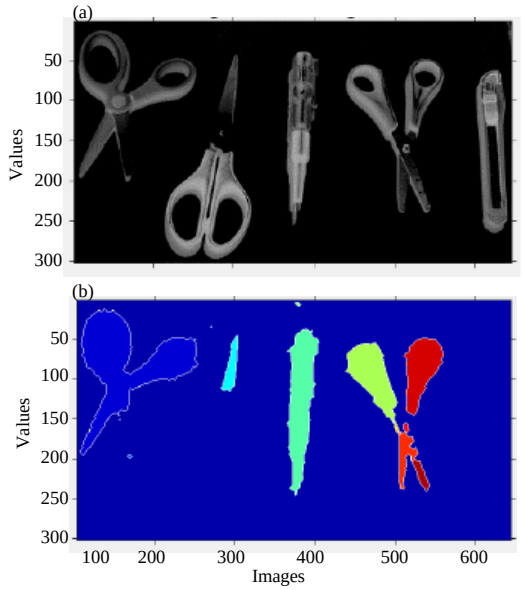


Fig. 3: Daetection of objects: a) Image without backgroud and b) Tools

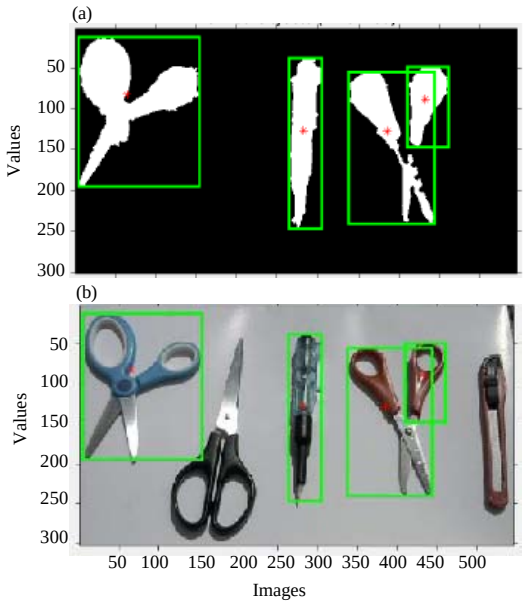


Fig. 4: Centroids and rectangles of each tool: a) Defined objects (Binarized) and b) Defined objects (Binarized)

the 5 tools were detected where the eye rings of the closed scissors were removed just like the scalpel, since, the algorithm by Guacaneme (2012) eliminates all elements that have contact with the edges.

On the other hand, the first scissors from left to right were almost completely recognized, losing part of the tool due to the gloss of the metal. The screwdriver was completely detected and the right scissors were detected in parts and not as a single object as the shifts caused by the shine of the metal generated that part of the image was considered as background.

Step 2; Centroids and selection of objects: In the second step of the algorithm, the image resulting from object detection is used and binarized in order to be able to operate the pixels of the image more easily. On it, the centroids of each tool are searched and marked with a rectangle each of the objects found as shown in Fig. 4b.

The rectangle completely contains the tool and covers all those white pixels that are in contact. With each of the rectangles it is possible to know the total number of tools recognized and their location in the image.

In Fig. 4a, the centroids of all the areas formed by white pixels are searched and the boxes that cover each of the areas are drawn. In Fig. 4b, the same boxes are placed on the original image to verify that the tools on the table that were found with the detection algorithm are indeed recognized.

The boxes are used to evaluate each tool separately. The position in pixels of the upper left corner of the box (with coordinates X_{H0}, Y_{H0}) and the width and height measurements of the rectangle (W, H, respectively) are known in such a way that it is possible to know all the pixels that compose the area within the box which are

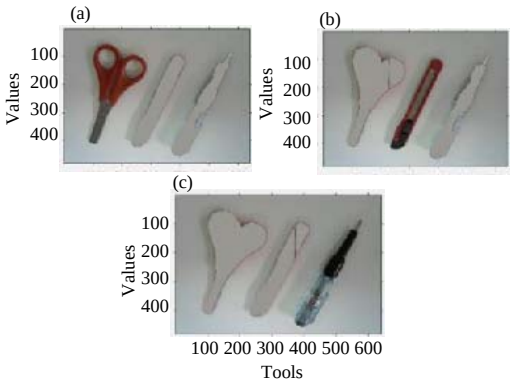


Fig. 5: Dectection of tools separately: a) Tool 1; b) Tool 2 and c) Tool 3

within the ranges defined by Eq. 1 where X_{re}, Y_{re} is the set of coordinates that make up the vertices of the box in which are found the coordinates X_{H0}, Y_{H0} of the box in the first column of the array and the coordinates of the following vertices, clockwise in the following columns:

$$\begin{bmatrix} X_{re} \\ Y_{re} \end{bmatrix} = \begin{bmatrix} X_{H0} & X_{H0}+W & X_{H0} \\ Y_{H0} & Y_{H0} & Y_{H0}+H & Y_{H0}+H \end{bmatrix} \quad (1)$$

The coordinate axis X is the horizontal axis of Fig. 5 and 6 positive to the right and the coordinate axis Y is the vertical axis and positive down.

Step 3; Tool recognition: The next step of the algorithm is to find the tool selected by the user where Convolutional Neural Networks (CNN) are used to recognize each of the objects detected in the work area and thus find the desired one.

Initially, all tools, except the one to be evaluated are "hidden" using three versions of the input image the original Image (Imo), a copy of the original Image (Imoo) and the binary image of Fig. 4a (Imb). The objective of "hiding" the objects is to try to make them as close as possible to the color of the surface that supports them, so that, only the object to be evaluated can be distinguished as shown in Fig. 5. With the binarized image the objects are searched to hide them in the copy of the original image the objects are hidden and with the original image all the tools are recovered to restart the recognition process with the next tool. In Fig. 5, each tool to be identified was separately plotted, hiding those that were not wanted to be observed. In case of tool 1, the scalpel and the screwdriver were hidden to make only the scissors visible, and in the cases of tools 2 and 3 only the scalpel and the screwdriver were visible, respectively.

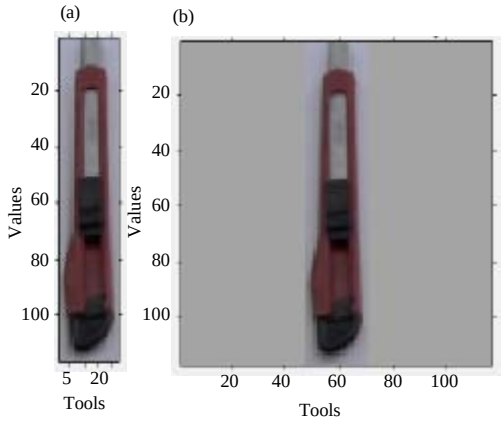


Fig. 6: Tool select for recognation with CNN

In order to enter the image of each tool to be classified by the network, the tool is cut out as shown in Fig. 6a and padding is added, so that, the image is square (Fig. 6b), since, the CNN input is size 128×128 due to the uniformity required for structuring the architecture of the network in the same way the padding prevents the image of the tool from being deformed when it is resized.

The structure of the Convolutional Neural Network is shown in Fig. 7 where the input image is in color and resized to 128×128 pixels. It consists of a deep architecture of 6 convolutional layers (CONV), 2 Dropout (DROP), 2 MaxPooling (POOL), 6 Rectified Linear Units (RELU), two fully connected and a softmax layer, allowing to identify one of the three categories with which it was trained: scalpel, scissors y screwdriver. Figure 7 shows the size of the Filters (F) of each convolutional layer, the Step (S), the Padding (P) and the number of filters (K). By Eq. 2 and 3, it is calculated the dimensions of the output image of each convolution layer where W_l and H_{l+1} correspond to the width and height of the input image and W_{l+1} and H_{l+1} are the width and height of the output image, respectively:

$$W_{l+1} = \frac{W_l - F + 2*P}{S} \quad (2)$$

$$H_{l+1} = \frac{H_l - F + 2*P}{S} \quad (3)$$

A fundamental part of the algorithm is to identify the desired tool which is the input parameter of the network. When the network recognizes one of the objects entered, compares the result with the name of the input tool desired by the user and upon detecting the first match,

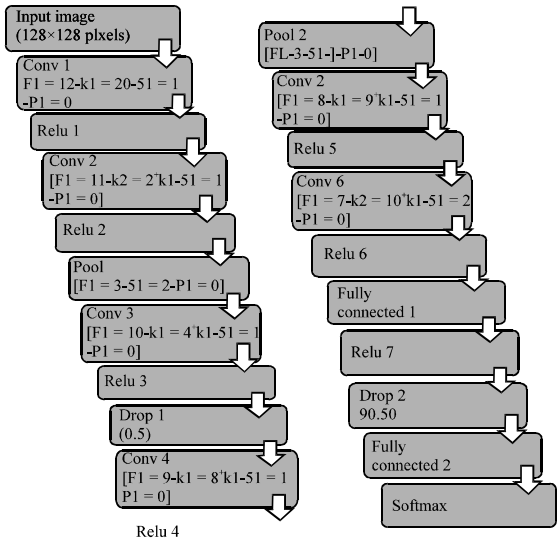


Fig. 7: CNN architecture

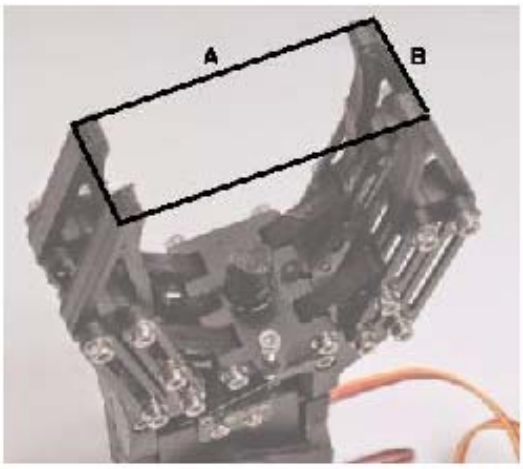


Fig. 8: Gripper dimensions

the recognition process is stopped, the image matching the desired tool is saved and the algorithm goes to the grip detection section.

Step 4; Selection of the grip zone: In the fourth step of the algorithm, the best grip position according to criteria of friction between the gripper and the selected tool is sought. The algorithm allows to enter the dimensions of the gripper corresponding to the maximum aperture thereof (dimension A of Fig. 8) and the width of the tips of the gripper (dimension B of Fig. 8), so that, it is possible to know the space occupied by the gripper in the working area and thereby define a grip where no clashes are generated between the tips of the gripper and other tools.

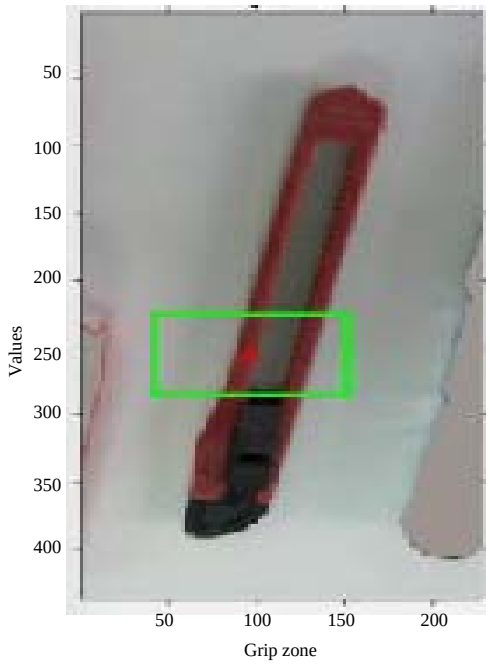


Fig. 9: Grip detection for an inclined scalpel

An example of the grip selection for an inclined scalpel is shown in Fig. 9 where the horizontal box represents the position and grip area of the gripper on the work area from which it can be obtained the degree of rotation that the gripper must have in relation to the tool and its coordinates in pixels to make the grip found.

The grip algorithm receives as input the image of the tool selected by the user but without adding padding as in the previous step, instead it increases the area of view of the tool when reading part of the elements that are near the object as shown in Fig. 9 where the vertical rectangle 1 is the box that covers the tool and that is obtained in the second step of the algorithm and the extra edges outside the box (area between boxes 1 and 2) correspond to the expansion of the viewing area where part of the original image is captured around the selected tool.

Unlike the CNN, in the gripping code the objects surrounding the tool must be known to have an idea of the free area that exists around the desired object, so that, the gripper can perform the grip without collisions. In Fig. 10a it is possible to observe more clearly the input image to the gripping code where it can be seen parts of the hidden tools such as the shadows and the edge thereof that become visible when looking for the grip.

For the calculation of the grip, the tool and some of hidden tools are segmented (Fig. 10a) in order to identify the geometry of the objects and focus the algorithm only

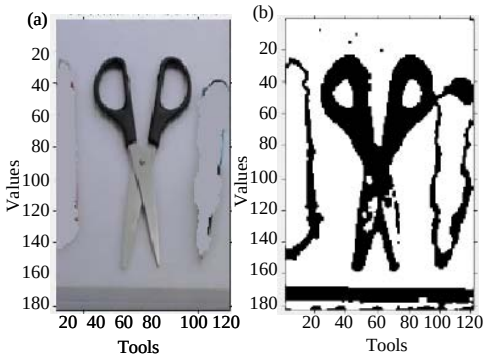


Fig. 10: Input image to the grip alghorithm: a) Selected tool: scissors and b) Binarized tool

on the segmented elements and not on the background. The image is binarized obtaining the result of Fig. 10b where the black pixels correspond to the objects and the whites in the background and a crop box is moved over it that is responsible for analyzing possible points of grip on the whole binarized image (box in the upper left corner in Fig. 10b). The frame is moved across the image and cuts out fragments of it to find a grip. Where the box is located, the image is croped below it and evaluated independently where the possible grips are those that comply with conditionals such as: degree of inclination of the gripping section with respect to the gripper tips, the area occupied by the tool section within the crop box and the free space between the gripping area and the area occupied by the gripper inside the frame to avoid collisions.

The dimensions of the frame that crosses the image are equalized to the dimensions of the gripper and when all possible grips on the tool are stored, the one with the highest grip area, the smallest inclination between the gripper section and the gripper, the more rectangular geometry and the smaller distance with the geometrical centroid of the tool is selected in order to look for the position and inclination of grip that ensures greater area of friction between the elements and that is closer to the centroid of the tool in order to look for greater stability of grip.

To determine the widest gripping surface, the number of pixels that compose the tool under the cropping box are added by Eq. 4 and select the highest value. The dimensions Dim 1 and 2 correspond to the width and height of the box im is the image captured under the box and, ..., the total sum of the pixels of the tool:

$$\text{Pix} \sum_{i=1}^{\text{Dim 1}} \sum_{j=1}^{\text{Dim 2}} \text{im}(i,j) \qquad (4)$$

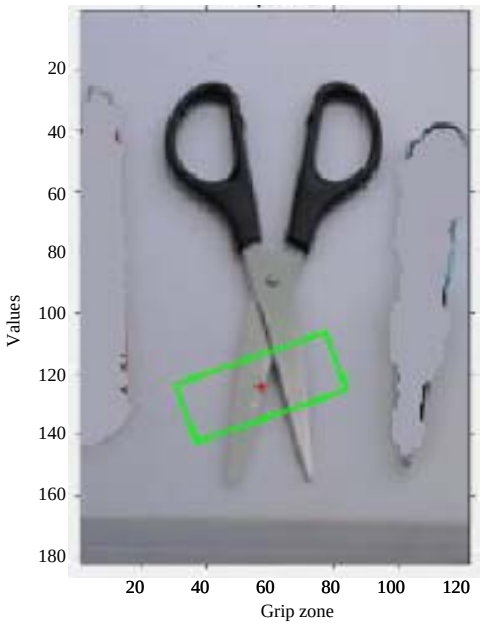


Fig. 11: Grip selection for scissors

In Fig. 11, a grip is selected for the tool of Fig. 10, where due to the dimensions of the gripper and the interference generated by the hidden tools around the scissors, a point of grip was chosen on the tips of the scissors whose stability is smaller than the area of the eye ring but it is in a region that is more accessible for the gripper.

The hidden tools as shown in Fig. 10b, generate some degree of interference in the search for the grip and in turn are narrow enough to prevent the gripping algorithm from considering them as part of the desired tool.

The gripping algorithm results in the X_c and Y_c position of the final gripping point, corresponding to the point in the center of the box of Fig. 11 and brings the inclination of the frame (Ang) which is evaluated between the horizontal of the original image and the edge of the frame dimension B, for example 110° of inclination for Fig. 11.

The coordinates X_c and Y_c correspond to the pixel position of the point of Fig. 11 with respect to the upper left corner of the image of Fig. 10a.

Step 5; Trajectory planning: Once the coordinates and the degree of inclination of the grip are obtained, the manipulator is controlled to hold the object and move it from one point to another where the end of the gripper performs a series of trajectories in a straight line to fulfill its objective. The first step is to obtain the coordinates of the gripping point with respect to the coordinates of the

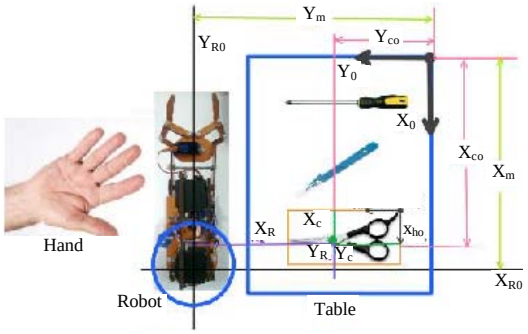
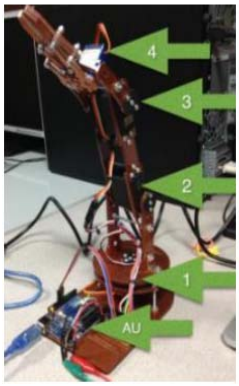


Fig. 12: Coordinates of the work area



Symbol	Small mainpulator
AU	Arduino one
1	Servomotor 1
2	Servomotor 2
3	Servomotor 3
4	Servomotor 4

Fig. 13: Mainpulator of 3° of freedom

manipulator, for which measurements must be made from the work area and conversions from pixels to millimeters until obtaining global coordinates. Figure 12 shows the general coordinates of the manipulator [X_{R0}, Y_{R0}], the position of the gripping point [X_C, Y_C] with respect to the general coordinates of the scissors box in Fig. 12 [X_{h0}, Y_{h0}], the general coordinates of the work area [X_0, Y_0], all in pixels and the working area measurements in millimeters corresponding to the distance between the X_{R0}, Y_0 axes and Y_{R0}, X_0 axes and the distance between the gripping point and the robot [X_R, Y_R] that determines the trajectory planning of the manipulator

The known distances are X_C, Y_C, corresponding to the final grip position generated in step four of the algorithm and the distance between the coordinate axis [X_0, Y_0] which corresponds to the position of the upper left corner of the tool box with respect to the upper left corner of the original image which is obtained in the second step of the algorithm with the coordinates X_{H0}, Y_{H0}. To know the distance X_{C0} and Y_{C0} the sum between the positions (X_c, Y_c) and (X_{H0}, Y_{H0}) is performed as shown in Eq. 5:

$$\begin{bmatrix} X_{CO} \\ Y_{CO} \end{bmatrix} = \begin{bmatrix} Xc \\ Yc \end{bmatrix} + \begin{bmatrix} X_{HO} \\ Y_{HO} \end{bmatrix} \quad (5)$$

Then the distance between the coordinate center [X_0, Y_0] and the center of the robot [X_{R0}, Y_{R0}] in millimeters is measured to obtain [X_m, Y_m] and the conversion of pixels to millimeters is performed for the distances [X_{C0}, Y_{C0}] in order to subtract both values as shown in Eq. 6 to obtain the final grip point coordinates [X_R, Y_R] with respect to the center of coordinates of the manipulator:

$$\begin{bmatrix} X_R \\ Y_R \end{bmatrix} = \begin{bmatrix} Y_m \\ X_m \end{bmatrix} + \begin{bmatrix} Y_{CO} \\ X_{CO} \end{bmatrix} \quad (6)$$

These coordinates allow to generate the trajectory of the manipulator from a certain starting point to the point of arrival where the desired object must be moved. For the tests, the camera was placed at a fixed height from the tool table and the distance [X_m, Y_m] in millimeters was measured. The conversion of pixels to millimeters also depends on the resolution of the camera where each millimeter contains a certain amount of pixels in a directly proportional relation as shown in Eq. 7 where the distance in pixels obtained for the grip [X_{C0}, Y_{C0}] by a constant B is multiplied in order to obtain its equivalence in millimeters and to use it in the subtraction (Eq. 6) to have the true grip position between the tool and the manipulator:

$$Milimeters = B * Pixels \quad (7)$$

Once the final gripping point is known, intermediate points are established for the trajectory where the end effector moves in a straight line between each of them, simultaneously moving its motors and passes through all until reaching the final position where it must leave the tool. The manipulator used which is shown in Fig. 13 has four servomotors that control the movement of the links and the opening and closing of the gripper and an Arduino Uno used to control each servomotor. It is the same manipulator that was used in a previous research exposed by Murillo *et al.* (2016) reason why its structure is known. The upper and lower views of the manipulator are shown in Fig. 14 whose inverse kinematics (Murillo and Moreno, 2016) are used to control the manipulator at each point in the trajectory, commanding the motors directly from MATLAB®.

To define each point of the trajectory, the distance between the start and end point for the components X-Z was calculated and the result was divided by a constant number of steps in order to obtain the intermediate positions of the trajectory and with them to calculate the value of angular rotation for each servomotor.

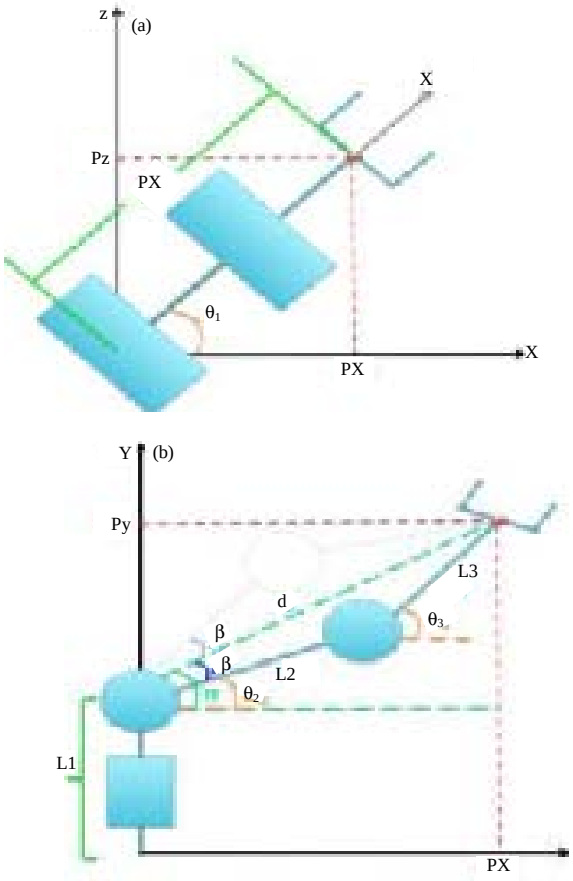


Fig. 14: Views of the manipulator for inverse kinematics

In Eq. 8 shows, the equation used to calculate each intermediate step of the trajectory is shown where $[X_f, Y_f, Z_f]^T$ are the coordinates of the end point $[X_i, Y_i, Z_i]^T$ are the coordinates of the initial point $[x_k, y_k, z_k]^T$ are the coordinates of the current position of the end effector, $[x_{k+1}, y_{k+1}, z_{k+1}]^T$ are the coordinates of the next end effector position and η is the number of intermediate positions to be taken by the manipulator to move from the initial to the final position where the calculation of Eq. 8 is repeated iteratively until the error between the current position and the final position is ≤0.05 mm:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \frac{1}{n}\left(\begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix}\right) - \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}\right) \quad (8)$$

Each intermediate position $[x_{k+1}, y_{k+1}, z_{k+1}]^T$ is entered into the inverse kinematics of the manipulator to calculate the angles of rotation of each servomotor and generate the motion in the actual manipulator, however, the coordinate plane from which the inverse kinematics was

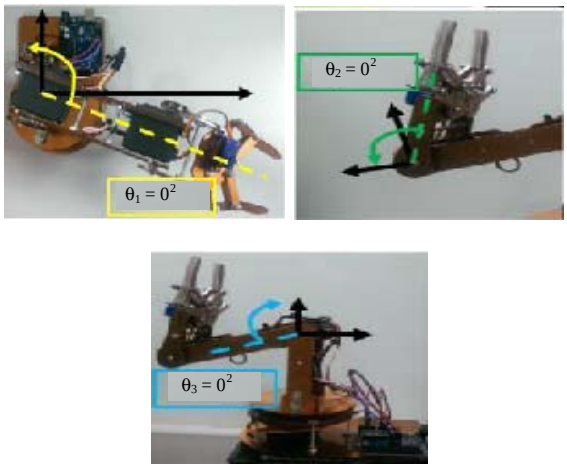


Fig. 15: Intial positions of the manipulator

calculated does not coincide with the directions of rotation of the manipulator motors or their initial positions, so, it was necessary to add or subtract an offset to each servomotor to consider these coordinated changes and to give the manipulator the real angles of each degree of freedom of movement.

Figure 15 shows the directions of increase of angular rotation for each servomotor and the initial position of each of them (dashed line) with respect to the coordinate axes that were taken in the inverse kinematics (arrows) and Eq. 9 shows the correction that was added to each angle where $[\theta_1, \theta_2, \theta_3]^T$ are the angles that are given to the manipulator and $[\theta_{1c}, \theta_{2c}\ \theta_{3c}]^T$ are the angles of inverse kinematics:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \theta_{1c} \\ \theta_{2c} \\ -\theta_{3c} \end{bmatrix} + \begin{bmatrix} -10° \\ -260 \\ -45° \end{bmatrix} \quad (9)$$

The correction angles indicated in Eq. 9 were experimentally determined by trial and error by executing a known trajectory where the offset angles for each servomotor were varied until the manipulator imitated the desired trajectory.

The displacement of the manipulator was divided into different linear trajectories where each inflection point is determined by the coordinates of Eq. 10 in which each column of the matrix represents a position in space with respect to the coordinate axis of the robot according to Fig. 12, the vector $[P_x, P_y, P_z]^T$ is the position of the end effector at each inflection point and X_R and Y_R are the positions of the final grip in millimeters. All measurements of the matrix are in millimeters:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} 128 & X_R & X_R & X_R & 0 & -100 \\ 5 & Y_R & Y_R & Y_R & 120 & 70 \\ 200 & 200 & 200 & 110 & 210 & 100 \end{bmatrix} \quad (10)$$

The first column of the matrix of Eq. 10 takes the manipulator to an initial position located 200 mm from the ground and with the end effector on the table, the next column takes it to the [X_R, Y_R] position of the final gripping point, then goes down to the table to hold the tool and finally climbs up again with thevcoordinates of column 4. After lifting the tool, it moves through the points of columns 5 and 6 to deliver the tool in the hand of the user located on the other side of the manipulator, achieving its final goal (Fig. 12).

In the next study, it is presented different tests carried out on the manipulator where is tried to determine the degree of precision in the grip with respect to the calculated coordinates for the three categories of tools trained in the CNN.

Test: A working area was set according to Fig. 12 where three tools, one of each category were located at different distances from the manipulator and the robot was fixed to the ground to ensure a fixed reference point. The camera and table were positioned as shown in Fig. 16 and polystyrene tools were used because the manipulator gripper does not have enough force to support the weight of real tools or generate enough friction to maintain the grip.

Figure 17 shows the input image captured by the camera (original image) where the three tools used for the tests can be distinguished. Binarization and segmentation of the input image (binarized image and defined objects), and the demarcation of each object found with a box (Defined Objects (Binarized)) are also shown. In the "Binarized image" all the objects recognized are set with black color and background with white in the "Defined objects" image, the noise of "Binarized image" is eliminated to make visible only the recognized objects, where the background is demarcated black.

After segmenting the work area, all objects found are recognized as shown in Fig. 18 and the user selected tool is exported as shown in Fig. 19 to calculate the grip where the "binarized tool" image is the segmentation and binarization of the scalpel entered and Fig. 20 shows the selected grip (for this example).

Similarly, with a more irregular and dark background than the tools, good results were obtained in the segmentation of objects as shown in Fig. 21 and 22, where both tools were differentiated from the surface and correctly identify through CNN thanks to the threshold changes made.

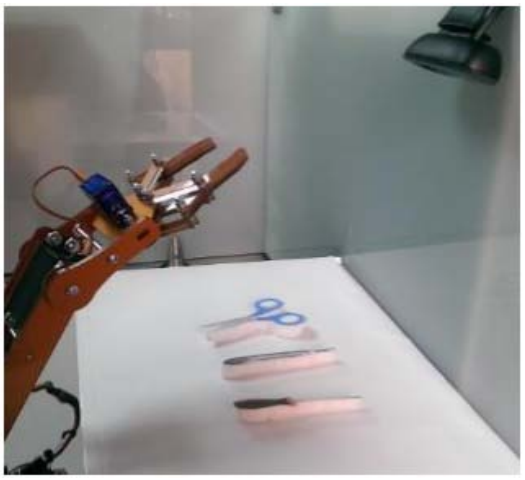


Fig. 16: General work areal

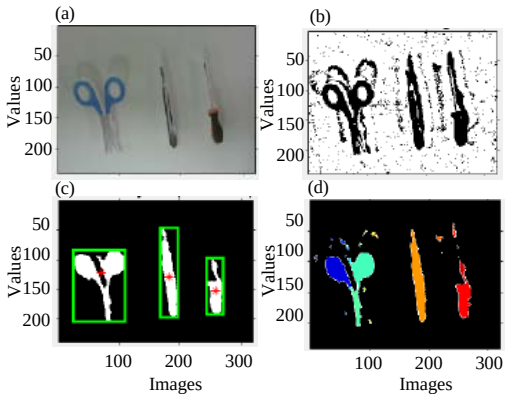


Fig. 17: Work area and segmation: a) Orignal image; b) Binarized image; c) Defined objects (Binarizes) and d) Defined objects (color)

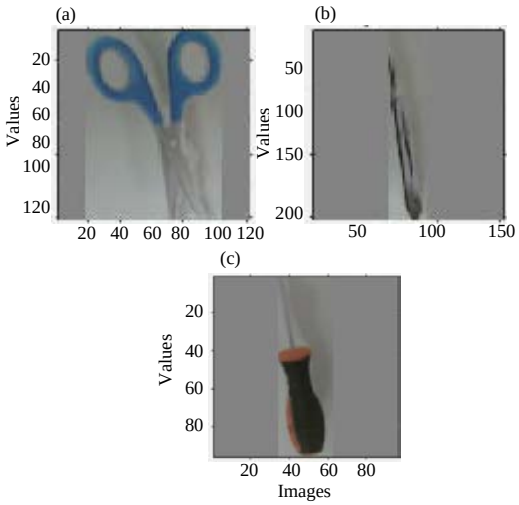


Fig. 18: Recognition of each tool: a) Scissors: 96.7253%; b) Scalpel: 99.8944% and c) Screwdriver: 99.6822%

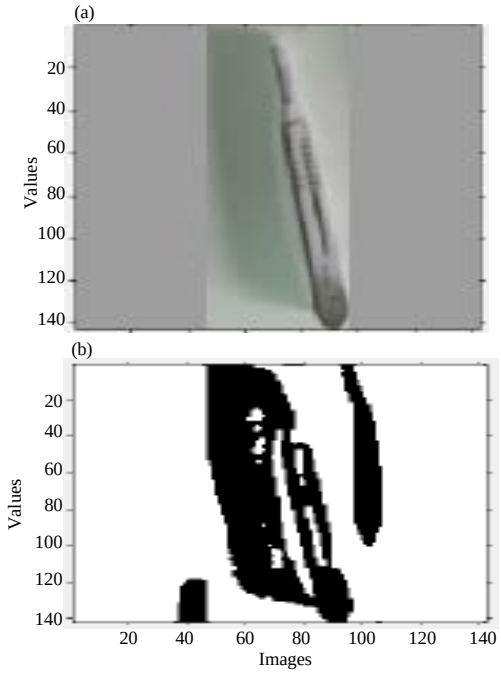


Fig. 19: User selected tool: a) Selected tool: scalpel and b) Binarized tool

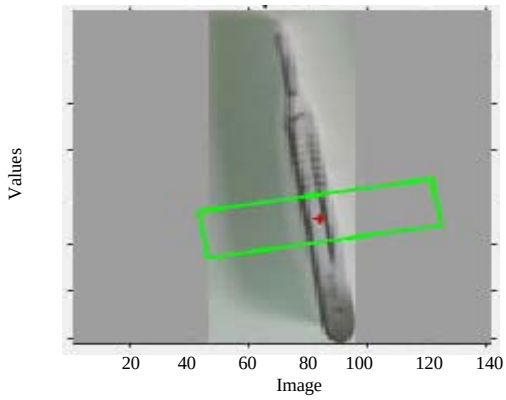


Fig. 20: Grip selection for a scalpel

Similarly, with a more irregular and dark background than the tools, good results were obtained in the segmentation of objects as shown in Fig. 22 where both tools were differentiated from the surface and correctly identify through CNN thanks to the threshold changes made.

On the other hand, Fig. 23 shows the activation areas of each object when is entered to the network, i.e., the parts of the object that most recognize the network where white sections represent the areas that most characterize the object and the black ones that are most irrelevant to CNN. It can be observed that, in the case of the screwdriver, the greatest area of activation is the handle, whereas in the case of the knife the entire body of the tool is distinguished. In the case of scissors, the eye rings generate the greatest activation of the tool. In contrast, the background is kept in dark tone which means thatthe network does not consider it at the time of the classification.

After selecting the tool and a feasible gripping point on it, the inverse kinematics of the manipulator is executed and the results of the physical grip are compared with the chosen one in the code, in order to try to establish the accuracy of the system. In Fig. 24, the grip selection for the scalpel can be compared with the grip made by the gripper, where it is possible to appreciate a slight shift of the end effector with respect to the final grip coordinates, however, the inclination and height of the gripper with respect to the tool body were found close to the specified gripping ranges with 98% accuracy.

For the scissors, a very close grip was achieved to the one selected by the code as shown in Fig. 25 where the adjustment of the gripper with the tool was performed almost on the desired point with an accuracy of 99%, achieving at the same time a quite stable grip due to its proximity to the geometrical centroid of the object.

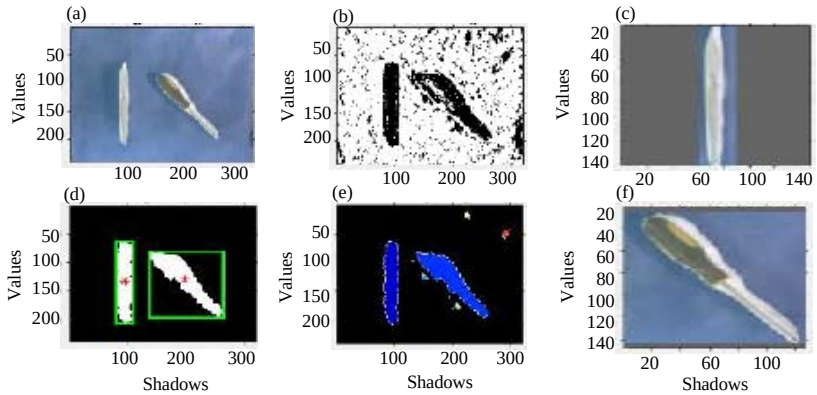


Fig. 21: Removel of the shadow: a) Orignal iamge; b) Binarized image; c) Scalpel: 99.9974%; d) Defined objects (Binarized); e)Defined objects (color) and f) Scresdriver

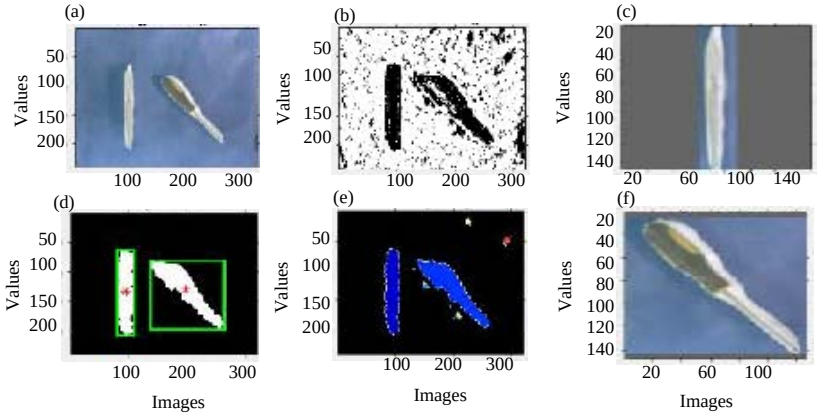


Fig. 22: Recognition of tools on dark irregular bacground: a) Orignal image; b) Binarized image; c) Scalpel: 99.9974%; d) Defined objects (Binarized); e) Defined objects (color) and f) Screwdriver

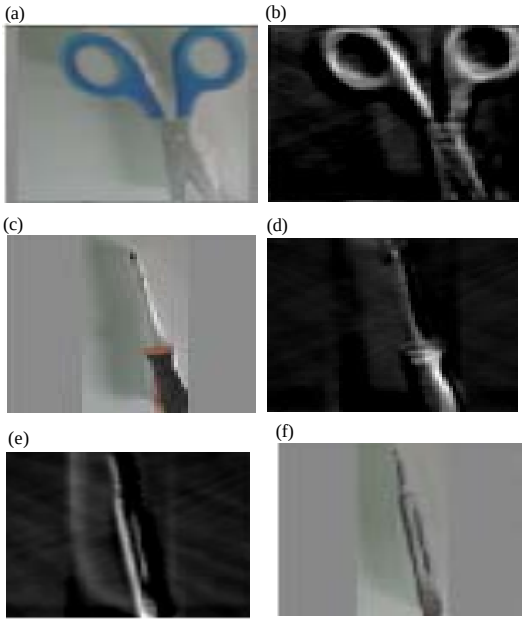


Fig. 23: Activations od each tool on the CNN

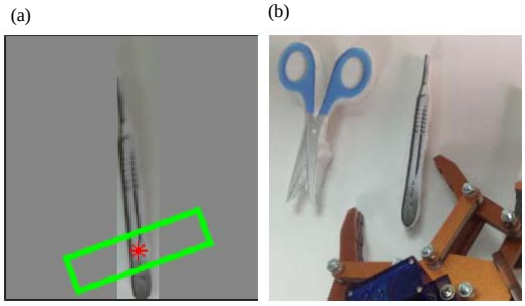


Fig. 24: Gripping on the scalpel

In the case of the screwdriver, a more noticeable displacement was observed than that of the scalpel where the physical grip reached a higher point than the desired

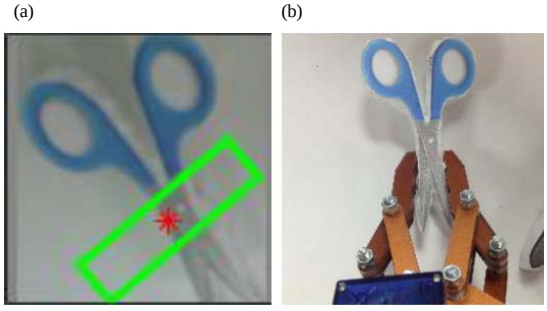


Fig. 25: Gripping on scissors

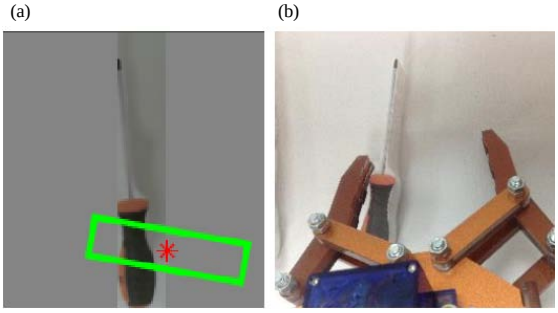


Fig. 26: Gripping on the screwdriver

with respect to the tool body as shown in Fig. 26, however, achieved the lateral grip shown in the box, where the center of the gripper is not just on the tool handle but shifted to the right with a grip accuracy of 78%.

RESULTS AND DISCUSSION

Four tests were performed on each tool placing them in different positions of the table and the distance between the final position of the gripper (physical or experimental grip) and the final grip point (theoretical grip

Table 1: Position errors

	Selected grip (mm)			Real grip (mm)			Relative error (%)		
Tool	X	Y	d	X	Y	d	X	Y	d
Scalpel	76.52	134.22	154.50	76	134.0	154.05	00.68	0.16	0.29
	77.71	144.33	163.92	75	144.0	162.36	03.49	0.23	0.95
	81.36	153.33	173.58	78	152.9	171.65	04.13	0.28	1.11
	62.36	143.36	156.34	60.8	143.0	155.39	02.50	0.25	0.61
Scissors	98.22	139.53	170.63	97.5	139.2	169.95	00.73	0.24	0.40
	106.89	142.38	178.04	106.5	142.0	177.50	00.36	0.27	0.30
	89.92	138.47	165.10	89.5	137.0	163.64	00.47	1.06	0.88
	93.51	138.73	167.30	92.8	137.2	165.64	00.76	1.10	1.00
Screwdriver	26.75	145.30	147.74	24	144.7	146.68	10.28	0.41	0.72
	33.73	146.82	150.64	32	146.0	149.47	05.13	0.56	0.78
	42.73	153.80	159.63	42.5	153.7	159.47	00.54	0.07	0.10
	25.91	147.59	149.85	23.4	147.0	148.85	09.69	0.40	0.66
Average error	-	-	-	-	-	-	03.23	0.42	0.65

or selected by the algorithm) was measured, obtaining the results of Table 1 where it is possible to visualize the relative error percentage for both the X and Y coordinates of the theoretical and experimental data and for the distance between the gripping point and the coordinate axis of the manipulator (d) that is obtained from the X and Y coordinates by Eq. 11 and an average error for X, Y and d at the bottom of Table 1:

$$d = \sqrt{X^2+Y^2} \quad (11)$$

From Table 1, it is possible to observe that the highest percentage of position error was obtained on the X axis of the manipulator with an average of 3.23% while in the Y axis there was an average error of 0.42% where the highest error in Y for all tests was 1.1 and 10.28% for X. In the case of distance, an average relative error of 0.65% was obtained where the highest error was found in the third test of the scalpel with 1.11%.

Based on Table 1, the data with respect to the distance (d) calculated by Eq. 11 were sorted in ascending order and the X and Y coordinates were plotted for each test with respect to distance d as shown in the graph of Fig. 27 where the horizontal axis represents the distance between the grip point and the coordinate center of the manipulator, the vertical is the distance in millimeters for each coordinate of the grip, the coordinates for the theoretical grip (X Grip, Y Grip) are marked with squares and asterisks and the coordinates for the experimental grip (Real X, Real Y) are marked with circles and diamonds, respectively.

As shown in the chart, the experimental grip imitates the behavior of the theoretical grip, however, its distance is smaller than desired by no more than 3 mm. This margin of error allows for cases such as the gripping of Fig. 25, that the position error does not greatly affect the actual holding of the given element, so that, the dimensions of

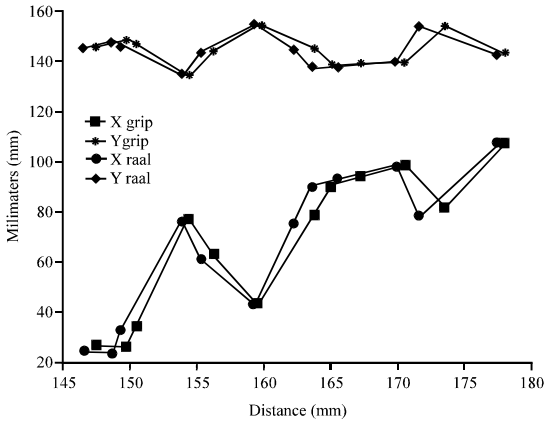


Fig. 27: Position error chart

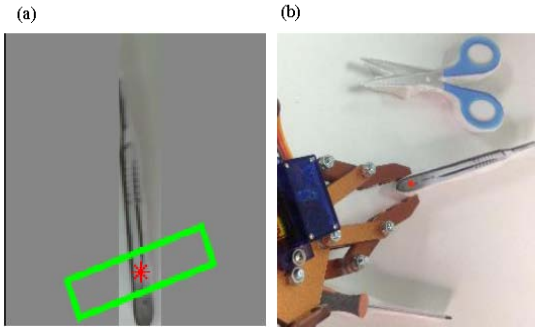


Fig. 28: Unstable grip

the object allow it to be held slightly lower or higher, however in a grip very close to the tool end as in the example of Fig. 28, may fail or be very unstable if it does not reach the desired position.

From the average relative error of the distance, it is determined that the grip accuracy has a margin of error of 0.65% which means that for grip points located at 150 mm and 200 mm distance from the robot, it can have tolerances of ±0.975 and ±1.3 mm, respectively.

CONCLUSION

Accuracy errors in the grip are influenced by the angle correction of the servomotors, since, it is not possible to take an actual measure of the number of degrees to be added or subtracted from the angles of the inverse kinematics to ensure that the position of the end effector exactly matches the calculated endpoint. They also depend on the resolution of the motors and the measures taken from the links which may differ by a few millimeters with the actual ones.

On the other hand, the location of the camera and work area in turn affect the coordinates of the manipulator, since, it must be ensured that the location of each element of the working environment is correctly positioned according to the measurements taken for the calculation of the actual grip positions. Any deviation from the camera, change of height of the table or change of position of the manipulator can cause that the coordinates change and the desired point is not reached. The tools, instead can freely change location and rotation as long as they can be captured by the camera.

In addition, the conversion of pixels to millimeters is not accurate which adds error to the calculation of the final position, especially, when the camera is not located on the center of the table but to the side which makes the capture of the image has perspective, i.e., the elements closer to the camera are larger while the more distant view is smaller, altering the actual measurements of each tool.

In spite of the various sources of error presented during the development of the algorithm, the robotic agent was able to perform an autonomous task through the algorithm when selecting and moving each type of tool as desired in the initial parameter, therefore, validating its functionality as a robotic assistance agent in the supply of tools.

The algorithm is able to recognize objects on backgrounds of different shades, however, very marked shadows as well as areas of the surface too illuminated, can affect the segmentation of the work area, even recognizing false objects, so, it is necessary to maintain a not very intense illumination to avoid the effect of shadows and shine.

ACKNOWLEDGEMENTS

The resaerchers are grateful to the Nueva Granada Military University which through its Vice chancellor for research, finances the present project with code IMP-ING-2290 and titled “Prototype of robot assistance for surgery”, from which the present work is derived.

REFERENCES

Adler, J., S.L.B. Ginting, B. Endar, B. Nurhandoko and P.D. Wardaya, 2017. Application artificial neural network-image processing to seismic wave propagation of carbonate rock. J. Eng. Appl. Sci., 12: 199-203.

Fergus, R., P. Perona and A. Zisserman, 2003. Object class recognition by unsupervised scale-invariant learning. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, June 18-20, 2003, IEEE Press, pp: II-264-II-271.

Garvin, G.J., M. Zefran, E.A. Henis and V. Kumar, 1997. Two-arm trajectory planning in a manipulation task. Biol. Cybern., 76: 53-62.

Gil, P., F. Torres and F.G.O. Zamora, 2004. [Detection of Objects by Combined Multilevel Segmentation of Color Spaces]. University of Alicante, Spain, ISBN:84-688-7460-4,.

Guacaneme, G., 2012. [Recognition of objects in images]. Electronica Inc., USA. (In Spanish) https://antiepoke.wordpress.com/2012/02/28/reconocimiento-de-objetos-en-imagenes/

Guevara, M.L., J.D. Echeverry and W.A. Uruena, 2008. [Detection of faces in digital images using cascade classifiers (In Spanish)]. Sci. Tech., 38: 1-6.

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. Imagenet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems, Leen, T.K, G.D. Thomas and T. Volker (Eds.). MIT Press, Cambridge, Massachusetts, ISBN:978-0-262-12241-3, pp: 1097-1105.

Leibe, B., A. Leonardis and B. Schiele, 2008. Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vision, 77: 259-289.

Liang, M. and X. Hu, 2015. Recurrent convolutional neural network for object recognition. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, IEEE, Boston, Massachusetts, ISBN:978-1-4673-6965-7, pp: 3367-3375.

Malpartida, E.A.S., 2003. Artificial vision system for the recognition and manipulation of objects using a robot arm. Master Thesis, Pontifical Catholic University of Peru, Lima, Peru.

Murillo, P.U. and R.J. Moreno, 2016. Multi user myographic characterization for robotic arm manipulation. Intl. J. Appl. Eng. Res., 11: 11299-11304.

Murillo, P.U., R.J. Moreno and O.F.A. Sanchez, 2016. Individual robotic arms manipulator control employing electromyographic signals acquired by Myo armbands. Int. J. Applied Eng. Res., 11: 11241-11249.

Pardo, A.E., 2009. [Object Recognition]. University of Barcelona, Barcelona, Spain, (In Spanish).

Stander, J., R. Mech and J. Ostermann, 1999. Detection of moving cast shadows for object segmentation. IEEE. Trans. Multimedia, 1: 65-76.

Xu, N., N. Ahuja and R. Bansal, 2007. Object segmentation using graph cuts based active contours. Comput. Vision Image Understanding, 107: 210-224.

Yao, L. and J. Miller, 2015. Tiny imagenet classification with convolutional neural networks. Master Thesis, Stanford University, Stanford, California.

Young, S.S., P.D. Scott and N.M. Nasrabadi, 1997. Object recognition using multilayer Hopfield neural network. IEEE. Trans. Image Process., 6: 357-372.