Trajectory tracking of a robot arm using images sequences

Aicha Belalia[1] , Samira Schouraqui [2], M'hammed Boussir [3]

[1] Computer science department, USTO, Oran, Algeria

[2] Computer science department, USTO, Oran, Algeria

[3]Physics department, USTO, Oran, Algeria

E-mail address: boumedbou@yahoo.com, s_chouraqui@yahoo.fr, bouademabdo@gmail.com

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: In this project, which belongs to the field of computer vision and robotics, we use the SCARA two -degree- of - freedom robotic arm to generate trajectories through a combination of two methods. The first part of our method is the CNN, which is used to detect and locate the object concerned by tracking a sequence of images acquired by the arm during its movement in a 2D plane. Spline3 is used to generate a trajectory using the polar coordinates found by the CNNs. because it generates a set of given points while minimizing unwanted oscillations and irregularities.

Simulations are acquired with a model of two degree of freedom (2-d.o.f.) of the manipulative arm SCARA has proven that the better the object is located, the better the trajectory tracking precision is.

The work carried out is evaluated using metrics: the OD part uses mAP, recall, and precision; the creation of the trajectory is evaluated by cosine similarity, MSE, and PSNR.

The primary objective of this work is to support surgeons during surgery by following their path and then using the SCARA arm to create an identical path. As an early effort, we tracked a simple object with a simple trajectory.

Keywords: CNN, Object detection,SCARA ,Spline3,Trajecory

APREVIATION LISTE

This *DL: deep learning.*
mAP: Mean Average precision.
TP: True positive
FP: False Positive
TN: True Negative.
FN: False Negative
MSE: Mean Squared Error
PSNR: Peak Signal-to-Noise Ratio.
AI: artificial intelligence.
R-CNN: Region-based Convolutional Neural Network
YOLO: You Only Look Once.
IOU : intersection on union distance.
DH: Denavit and Hetenberg .
2d.o.f : two degree of freedom.
CNN: convolutional neural network.
SCARA: Selective Compliance Assembly Robot Arm.
IGM: Inverse geometric model.
DGM: Direct geometric model
OD: Object detection.

1. INTRODUCTION

First, Areas attracting researchers' attention include robotics and trajectory generation. In this case, the robot cannot perform the task if it does not have the information it needs to perform the task. Object detection is the first step in this field.

Trajectory planning is to find a time serial of successive joint angles to advance a robot from an first configuration to a last configuration in order to realize a task.[05]. The trajectory generator is used to provide a temporal history of the variables that are important for a robotic manipulator, such as the intended locations, speeds, and accelerations in the state space, i.e.

The space task, also known as the articular space (Cartesian space). At a set interval, the generated trajectory signals are redirected to the motion control system, which is also referred to as trajectory tracking. The robot manipulator follows the plan thanks to the motion control system. [21], [23] In general, two separate subproblems fall under the umbrella of the trajectory generation problem. [24], [05]. The first, referred to as "path planning," is supplying the spatial point sequence required to finish the task in order to construct the path geometry. In the second, referred to as "trajectory planning and optimization," the temporal (dynamic) structure of the trajectory is designed in accordance with optimization criteria.

The second component addresses a wide range of topics and is mostly dependent on the particular requirements of the application. [05]. an important aspect of trajectory planning is object detection and tracking, which allows the robot to interact with the environment by identifying and manipulating objects. Object detection (OD) is used to locate objects in a specific area in a specific image, which is called object localization technology. [01]. OD can produce valuable information for the linguistic interpretation of pictures and videos. It can be applied to a variety of tasks, including autonomous driving, image classification, facial recognition, and human behavior analysis [02]. In recent years, deep learning DL technology, especially convolutional neural networks (CNN), has shown great potential in improving the accuracy and robustness of target detection algorithms.

Object detection and tracking involve analyzing images or video frames captured by cameras mounted on robotic manipulators. Traditional methods often rely on handcrafted features and classifiers, which require a significant amount of domain expertise and may not generalize well across different environments and object types. In contrast, deep learning approaches leverage large amounts of labelled data to automatically learn relevant features and classifiers.

In this work, the neural solution has been used to plan the trajectory of a SCARA manipulator arm. The Robot arm is equipped with a camera to take a series of images; these images will be processed by the CNN in order to locate the object concerned with the tracking. The results of the localization will be stored in a file. An algorithm for creating the trajectory of a SCARA manipulator arm will then use this file. In this step, the neural solution, spline interpolation method are also applied in order to obtain a better result.

The object detection and tracking by CNN used in this work are justified by statistical measurements such as mAP, TP, FP,TN, FN,F1 SCORE, RECALL, also the loss value (Loss) ,the average loss(Avg_Loss), Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) play a very important role in the evaluation of our work.

This is how the rest of the paper is structured. In Section 2, it is explained how to describe earlier research related to a problem. The approach employed is described in Section 3 and contains information regarding data pre-processing, CNN algorithm OD, and performance metrics for algorithm assessment. Trajectory generation of SCARA arm manipulator is introduced in section 4. Section 5 provides obtained results and evaluation. Section 6 outlines the conclusions from the present work.

A. *Related work*

Several works have been done in the area of trajectory planning and object detection. The following section explains a few searches in these two branches of artificial intelligence (AI).

a) The *generation of trajectories*

When you control a robot, it is in your interest to be able to individually control each articulation or axis to be well in control of the trajectory. [07]

The Optimal Trajectory Generation Method proposed by [03], based on the spline cubic interpolation approach. The best compatible approach to generate a smooth trajectory that passes through the points (x, y) is to use the cubic spline (spline3).

Cubic spline (spline3) is a commonly used interpolation method to generate smooth curves from a set of control points. This method adjusts a series of cubic polynomials between control points to create a smooth, continuous curve. The cubic spline is well suited to represent regular and continuous trajectories for the SCARA arm from discrete control points. [28]- [06]. Alia Kamel and All in [17]: have found that the neuronal method developed is effective and that it is a very good solution to the problem of trajectory planning. The trajectory obtained is optimal in the direction of the shortest path. The advantages of this method include ease of implementation and representation. They work on a grid and each cell is in 26-connectedness. That is to say, from one cell, we have only 26 possible displacement choices. On the other hand, this method has a flaw. In particular the fact that the kinematic constraints of the robot are not taken into account (acceleration, speed, maximum angular velocity etc.). The research group of Brock and Xiao focuses on real-time approaches for motion planning of (multi-)robots. One popular way to represent determined trajectories is with splines2. During algorithm execution, a motion-scheduling algorithm determines the appropriate sets of spline nodes and trajectory parameters. Yang and Brock portray currently planned paths using vertices ("milestones") and collision-free edges on a roadmap, another sort of representation. These milestones or nodes are produced by taking a broad picture of the robotic system and its surroundings. Including a form of motion planning from a global perspective [09].

The trajectory-planning problem of redundant manipulator arms is widely studied in the literature. Its resolution requires the consideration of a number of constraints according to [25], which are:

- The calculation of the different configurations through which the robot must pass;
- Obtaining smooth curves (positions, speeds, accelerations).

Considering these two constraints in the resolution process can be done in two different ways.

The first is to presuppose that various curves follow smooth trajectories (using polynomials or trigonometric functions). The purpose of analysis is to calculate the parameters of each curve. The second technique is to handle these two constraints separately.

Therefore, different configurations are calculated and then interpolated. In addition to these two limitations, we also need to address the issues of robot redundancy and functionality [18].A technique that functions in an N-dimensional configuration space, where N is the number of manipulator degrees of freedom, was developed by Lozano-Pérez [08]. Polyhedral volumes in the workspace model the robot and obstacles. The construction of the configuration space is based on the calculation of the joint angles in contact with the polyhedral volume. Once the obstacles in the configuration space are calculated, the non-obstacles represent the legal space for robot movement

b) Object detection by CNN

In deep learning techniques, algorithms analyze data to detect relevant features and then combine them for rapid learning. To recognize objects and people based on images obtained from videos, various deep learning techniques are used. [32]. Among deep learning methods, CNN is most suitable for image-based search, and it achieves the best recognition accuracy in various applications. Therefore, we consider CNN in this study.

Cnn-based od algorithms include:

R-CNN is an object detection architecture. R-CNN first extracts regions of interest in the image and then uses these regions as input data to the CNN. This region division makes it possible to detect multiple objects of different categories in the same image. This solution proposed by Girshick et al., 2013 , improves the accuracy of the recognition model [13].

Faster-CNN: The most popular object detection

method is: Faster R-CNN. It is part of R-CNN. Fast R-CNN improves upon on the R-CNN family . In 2014 , Ross Girshick and others developed R-CNN , resulting in faster R-CNN. A convolutional neural network (CNN) receives the input image to obtain a property map of the objects present in the image. Backbone is a Region Proposal Network (RPN), which then uses this feature map and anchors to generate region proposals. These regions are then filtered using NMS (non-maximal deletion).By combining the character maps and bounding boxes of connected objects that the CNN extracted, new feature maps are created by pooling regions of interest (ROIs). After that, a fully connected layer processes the grouped areas to forecast output classes and object region coordinates. The header network is the name of this component of the Faster R-CNN design.

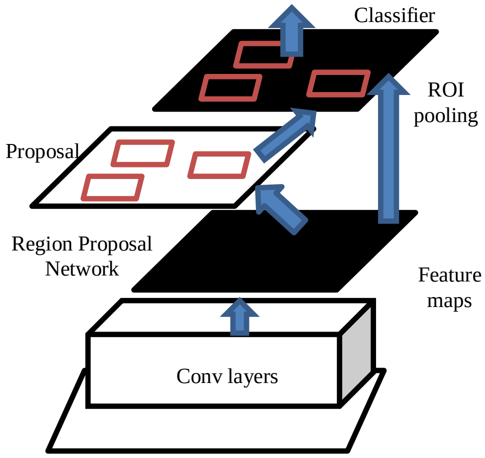

Fig 01: CNN architecture 01 [13]

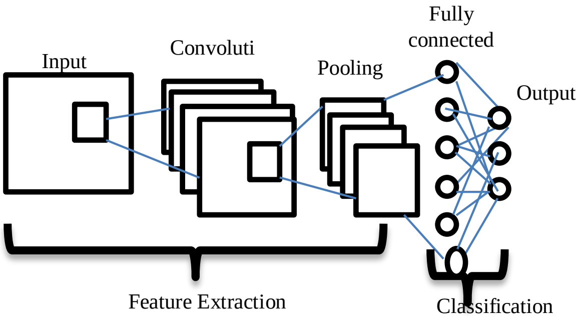

Fig 02: CNN architecture 02 [13] .

YOLO: The technique is among the frameworks that the majority of real-time object tracking and detection applications employ these days. [04]. YOLO, or "You Only Look Once," is a deep neural network that processes an image just once; this is in contrast to "region proposal methods," which are primarily employed by R-CNN-based models. There are seven iterations of YOLO (YOLOV1 ... YOLOV7).

METHOD

Our methodology is divided into two main parts, the first is the object detection of the image sequences acquired by the arm, using in this case one of the deep learning methods, which is CNN. The second part of our work is the trajectory generation using the result of the first part and the spline method.

Our method can therefore be represented by the following flowchart

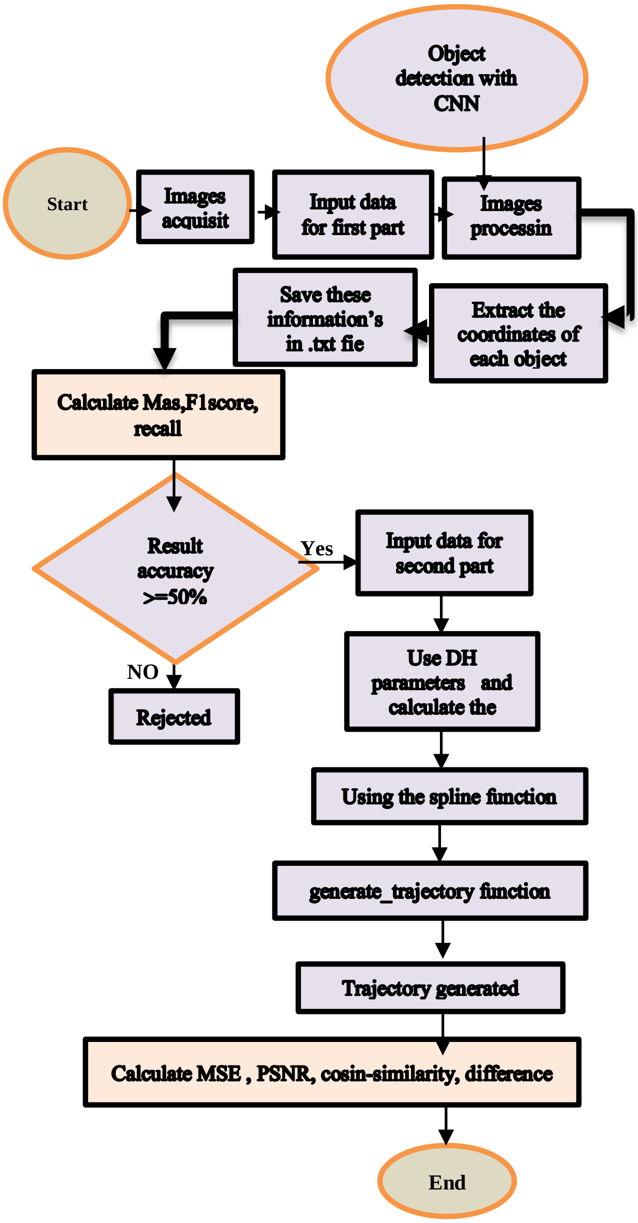
Object detection with CNN
Start
Images acquisit
Input data for first part
Images processin
Save these information's in .txt fie
Extract the coordinates of each object
Calculate Mas,F1score, recall
Result accuracy >=50%
Yes
NO
Input data for second part
Rejected
Use DH parameters and calculate the
Using the spline function
generate_trajectory function
Trajectory generated
Calculate MSE , PSNR, cosin-similarity, difference
End

Fig 03: Flowchart of the employed method.

As the preceding flowchart illustrates, we started by processing the consecutive photos that the SCARA arm had acquired. Additional details are displayed below.

A. *Object detection with CNN*

There are 172 photos in our database, all of various sizes. Among them 20 images will be used for the test and to extract the necessary information (position of gravity centers in all images, map, loss, precision, etc. . .).

Out of which 152 images will be used for learning. The pre -trained network Darknet 53 will be used to process the 152 images in the first step. The Darknet 53 first resizes each images to a size of 416*416, in color (3 channels) in order to extract its features. This method is based on a technique known as anchors, delimitation boxes or Bounding Box , which frames an object in a rectangular space.

Each image will be divided first on 32 then on 16 and finally on 8, resulting in images that have been divided into a grid of cells (13x13 , 52x52 , 26x26)

The object's center is located on in the cell that is in charge of object detection.

We obtain for each image (13x13 + 52x52 + 26x26) * 3 = 10647 bounding boxes. The algorithm will select one box from this group to identify our object.

Each image is processed by a set of layers: convolution, pooling, ReLU correction and fully connected

a) *Convolution:* Its objective is to identify whether a specific set of characteristics (or features) is present in the input images. To do this, convolutional filtering is employed. The basic concept is to "drag" a window that represents the filter over the image, and then calculate the convolutional layer by taking the convolutional product of the filter and each segment of the scanned image.

b) *The ReLU correction layer:*

The real nonlinear function given by ReLU(x)=max(0,x)ReLU(x)=max(0,x) is represented by the symbol ReLU (Rectified Linear Units).

Therefore, any negative values received as inputs are replaced with zeros by the ReLU correction layer. It serves as a mechanism for activation.[34]

c) *The pooling layer:*

It takes in several feature maps as input and performs the pooling procedure, which reduces the size of the image files without sacrificing quality. The image is split up into square cells and each cell's maximum value is preserved in order to achieve this. In practice, tiny square cells are often used to prevent too much information loss. The most common choices are 2 ×× 2 pixels that are near to each other but do not overlap or 3 ×× 3 pixels that are near to each other but are spaced apart by a step of 2 pixels (and so overlap).

The amount of feature maps in the output and the input are same, however they are significantly small.

d) *The fully-connected layer:*

The fully-connected layer is the final layer of a neural network, converting a vector into a new vector using linear combinations and activation functions. It classifies images using a vector of size N, with each element representing the probability of the input image belonging to a class.

The hyper parameters of our data base:

So all the images (152 images) are managed in the first place by our cfg file which contains all the hyper parameters of its 4 layers as follows:

e) *Feature extraction*

We developed the K-means algorithm and replaced the Euclidean distance it employs with intersection on union distance (IOU) in order to determine the distance between the delimitation boxes estimated by the algorithm and the predefined bounding box.

$$dist(box, centroid) = 1 - IoU(box, centroid). \ (1)$$

The closer the IOU is to one, the better the results.

The extraction of features from images will be saved in a file as weights. Its weights will later be used for object detection in other images.

The following list of formulas is displayed during the object detection phase:

Confusion Matrix: Four elements are required in order to create a confusion matrix:

TP: Based on the ground truth, the model accurately anticipated and matched the label.

TN: The model is not a component of the ground truth and does not forecast naming.

FP: The model predicts a label, but it does not correspond to the ground truth (Type I mistake).

FN: The label is a component of the ground truth, but the model does not forecast it. (Error type II).

$$Precesion = TP/(TP + FP) \ \ldots\ldots..(2)$$

$$Recall = TP/(TP + FN) \ldots\ldots\ldots\ldots\ldots(3)$$

mAP Finding the average accuracy (AP) for each class and averaging it over several classes yields mAP.

$$mAP = \frac{1}{N} N \sum_{i=1}^{N} AP_i \ldots\ldots.(4)$$

$$F1\ score \ = 2 * (precision * recall)/(precision + recall)..(5)$$

The subsequent section will make use of the findings obtained in this step.

B. *Trajectory Generation*

In order to generate the trajectory of our SCARA arm manipulator, we must go through several steps concerning the arm joints .We must define its direct geometric model and its inverse geometric model.

SCARA is a Selective Compliance Assembly Robot designed for handling pallets, boxes, cartridges and other objects that require horizontal or parallel grasping and movement.

a) *The SCARA 2d.o.f robot (RR)*

A SCARA robot defined in ISO standard 8373:1994, No. 3.15.6. SCARA robots use two or three parallel revolute joints to provide compliance in the horizontal plane against vertical loads. Characterized by high rigidity for vertical loads and flexibility for horizontal loads, making it ideal for vertical assembly tasks and small object handling,[30]. The first prototype of the SCARA robot was built in 1978. The second one built in 1980. In 1981, some industrial partners began to market their own versions of the SCARA [26].

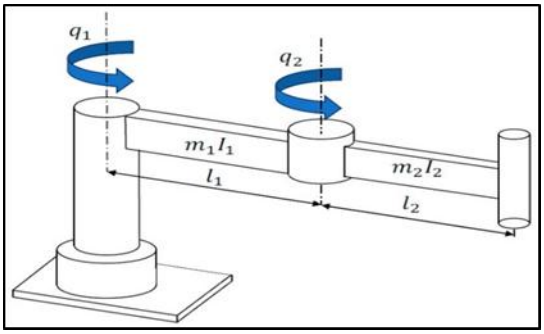

Fig 04: SCARA Arm Manipulator. [11]

b) *Denavit and Hetenberg parameter*

For the construction of the marks, it is necessary to follow the following procedure:

- Name the robot fields from i=0 to i=n starting at the robot base with i=0.

- Name the joints from I = 1 to n (1 for the first degree of freedom and n for the last).

For i = 0 up to i = n-1 fix the Zi axis on the articulation i + 1.

- The origin of the coordinate system R0 will be any point on the Z0 axis, so that X0 and Y0 form a direct orthonormal coordinate system.

- For i=1.2,..., n-1, the origin of the coordinate system is fixed at the intersection of the Zi axis with the perpendicular line common to Zi-1 and Zi. If the two axes intersect, the origin is the point of intersection.

- If, on the other hand, the axes are parallel the origin is the origin of the coordinate system of the articulation i + 1.

- The Xi axis is the perpendicular line common to Zi-1 and Zi, and the Yi axis is chosen so that the coordinate system is directly orthonormal. [19]-[27].

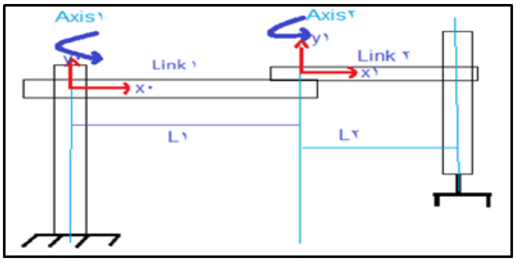

Fig 05 : SCARA arm structure.

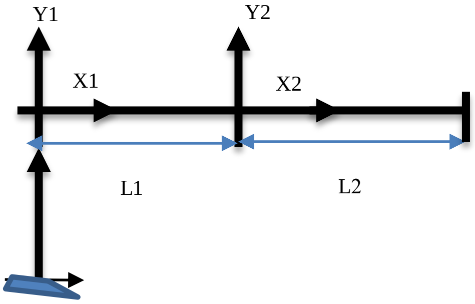

Fig 06: SCARA robot coordinat system

The DH method principle dictates that the SCARA Robot's structural parameters are categorized into four groups. The torsional angle αi, angle of articulation θi, joint length ai, and linkage length di make up these groups. The following are the definitions for these parameters:

(1) di is the translation, in the positive direction zi, between the xi-1 and xi axes.

(2) The translation between the axes of Zi-1 and Zi is ai-1, and it is positive along the xi-1 direction.

(3) The rotation of approximately xi-1 counterclockwise between the axes of zi-1 and zi is denoted by αi-1.

(4) The rotation around zi counterclockwise between the axes of xi and xi-1 is denoted by θi..

c) *Khalil's model*

In 1986, Khalil and Kleinfinger made improvements to the DH model due to ambiguities regarding robots with closed or tree structures. They also proposed a method for robots with simple open architectures. For robots with an open structure, the following convention is proposed:

- The coordinate system Rj is fixed with the link j,

- The axis of the articulation j is the axis zj and the axis xj is aligned with the perpendicular common to the current axes of articulation j and following j + 1.[33]

With this convention the equation of the transformation matrix from the coordinate system j-1 to the coordinate system j is:

$$T_{j-1}^{j} = Rmat(x_{j-1}, \alpha_j) * Tras(x_{j-1}, d_j) * Rmat(z_j, \theta_j) * Tras(z_j, \theta_j) \ldots\ldots(6)$$

Where: The angle formed by xj-1 and xj about the zj axis is denoted by θi.

The distance along zj between xj and xj-1 is denoted by rj.

The length of xj-1 is denoted by dj, while the angle formed by zj-1 and zj around xj-1 is represented by αj.

In relation to the coordinate system j-1, the coordinate system transformation matrix j is as follows: [20].

$$ {}^{j-1}_{j}T = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_J & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \ldots.(7)$$

d) *Two d.o.f SCARA arm geometric modelled on a 2D plane:*

The configuration of the robot is determined by the articular variables θ1 and θ2 (Figure 07) and The DH parameters of the SCARA robot (2d.o.f) are presented in Table 1: (With:l1=0.2m and l2=0.3m)

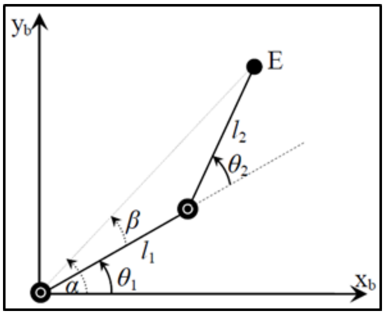

Fig 07: The SCARA arm configuration.[29]

TABLE 01: SCARA ARM PARAMETERS.

link	***ai***	***αi***	***di***	***θi***
1	L_1	0	0	θ_1
2	L_2	0	0	θ_2

To calculate the DGM of SCARA , we calculate the transformation matrix following this steps:

First in our case, we are going to work on a 2D plane because the CNN are mainly designed for 2D vision (images) and do not have a direct architecture to process depth information (z) from a single image.

In this case the translation matrix is

$$ {}^{0}_{2}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & L_1 + L_2 * \cos\theta_2 \\ \sin\theta_1 & \cos\theta_1 & L_2 * \sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix} ..(8)$$

e) *Inverse geometric model:*

The problem is to calculate the joint coordinate's θ of the robot from the operational coordinates X. The inverse geometric model is the inverse problem that makes it possible to know the articular varied according to the situation of the terminal organ, which can be represented by the relation: [10][27]

$\theta = g(X). \quad ..(9).$

Where X is the vector of the operational coordinates expressed in the reference frame R0, and q is the articular variance.

The matrix T, 0 n represents the position and orientation, expressed in the reference frame R0, of the robot's terminal member.

Three methods of calculating IGM can be distinguished [10]

- Paul's technique [22], which is appropriate for the majority of industrial robots and handles each unique instance separately.

– Pieper's solution [16] solves the problem for six-degree-of-freedom robots with three prismatic joints or three roasted joints of consecutive axes.

– The overall approach used by Raghavan and Roth [15], giving the general solution of six-joint robots from a polynomial of degree at most equal to 16.[12]

Let us be the following figures:

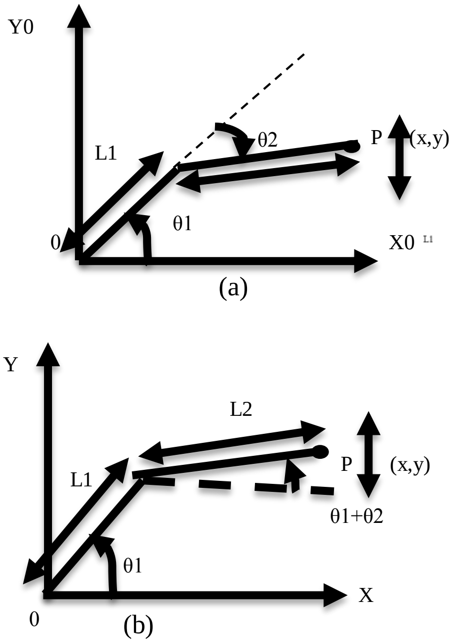

Fig 08: Arm parameters in a 2D plane.[27][31]

We have the following direct geometric model:

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2) \dots .(10)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2)$$

Let us complete the scheme of the direct geometric model by the Generalized Pythagorean theorem:

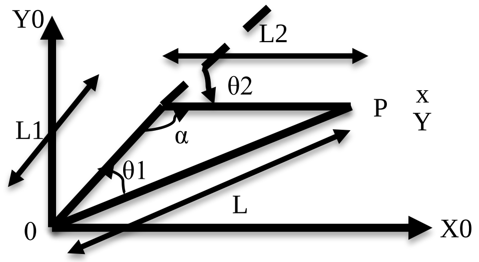

Fig 09: Pythagorean Theorem.

We have the following relationships:

$$L2 = x2 + y2 , \quad ..(7)$$
$$L2 = l12 + l22 - 2l1\, l2\, cos\,(\alpha) ,..(11)$$
$$\theta = \pi + \theta 2 \quad ..(9)$$

(Generalized Pythagorean Theorem),

Where from?

$$x2 + y2 = l12 + l22 - 2l1\, l2\, cos\,(\theta 2) \quad ..(12)$$

(we have : $cos(\pi + \theta) = -cos(\alpha)$) (13)

When $\theta 2$ is positive (resp., negative), the robot has a low elbow posture (resp., elbow high).

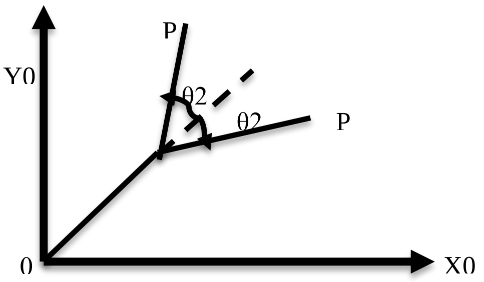

Fig 10: The calculates of θ1 and θ2.

Let be, via the spline method, one calculates

As a result the MGI is:

$$\theta_1 = Arctg\left(\frac{y(L_1 + L_2\cos(\theta_2)) - xL_2\sin(\theta_2)}{x(L_1 + L_2\cos(\theta_2))}\right) \dots (14)$$

$$\theta_2 = Arccos\left(\frac{x^2 + y^2 + {L_1}^2 + {l_2}^2}{2L_1 * L_2}\right) \dots (15)$$

C. The followed procedure and limitations for trajectory genartion :

The data from CNN object detection was stored in.txt files, providing coordinates for the object's study note and used as input for the arm manipulator system. Therefore, based on the following limitations, we design the trajectory of a SCARA manipulator arm.

- The robot's kinematic constraints, such as acceleration, speed, and maximum angular velocity, are not considered.

- Ignore collisions that occur on the work plane.

- Our work plan is divided into zones, each containing a single point with coordinates (x, y) representing the object's center of gravity detected in the image at time ti.

- The images were sorted based on their acquisition time, with images 01.png taken at time t1 and 02.png taken at time t2, and so on.

- There are six seconds worth of acquittals, each lasting 300 milliseconds.

- We have obtained 20 images, or discarded those with a detection accuracy below 50%, resulting in only 3 images in our case.

- The matrix containing time (ti) and coordinates (xi, yi) at time (ti) is used as input for the previously created model following articulation rules and respecting Direct Geometric Model (MGD) and Inverse Geometric Model (MGI).

D. The proposed trajectory generation procedure:

Joint 1 (for the abscissa): It controls the movement of the arm along the x-axis.

Joint 2 (for ordinates): It controls the movement of the arm along the y-axis.

The coordinates of the points we want to reach are given by (x, y).

The trajectory was created using the cubic spline interpolation method.

This cubic spline allows us to smoothly connect the key points and generate a continuous trajectory. Additionally, by adjusting the parameters of the cubic spline, we can control the smoothness and curvature of the trajectory to match specific requirements or constraints.

For each joint, there is a function.

Equation for articulation 1:

($\theta 1$): $\theta 1\ (t) = f1\ (t)$

Equation for articulation 2:

($\theta 2$): $\theta 2\ (t) = f2\ (t)$.

Consequently, we completed the following tasks in total:

-The first step is to create a matrix to store the (x, y) coordinates of the regions.

- Establish the durations needed to arrive at each point, which correspond to the quantity of images used for the test.

- Using the spline function, we obtain the cubic polynomials for each joint θ1 and θ2. For this process, we have converted the coordinates (x, y) utilized for this process to meters.

- By inputting the vector (xi), vector (yi), length L1, length L2, and time t into the function, we can obtain the corresponding angles θ1 and θ2 at that specific time. The function will return two vectors of the same length, where one vector contains the values of θ1 and the second vector the values of θ2.

- The generate_trajectory function takes the two vectors of angles θ1 and θ2 and combines them to create a single trajectory for the arm. This final trajectory represents the movement of the arm over time, allowing us to visualize its path and analyze its motion. Additionally, this function can be customized to include any necessary calculations or adjustments to optimize the arm's performance.

2. RESULTS AND DISCUSSION

As previously mentioned, there are two steps in our work. The first phase involves using CNN to recognize objects in image sequences; the second involves utilizing the result to create a trajectory that includes spline 3. All information regarding the outcome is arranged in accordance with the two steps in the section that follows.

A. Object detection :

After applying the CN networks, 2000 iterations on 152 base images. The results of extracting feature maps are explained below.

Object detection is illustrated by a rectangle containing the center of gravity coordinates of that object.

The detection's outcomes, represented as weights, will be kept in backup files: such as this form

(Next mAP calculation at 2000 iterations)

Last accuracy mAP@0.50 = 99.98 %, best = 99.98 %

In addition, as we notice the different values of this detection are given as follows:

detections_count = 223, unique_truth_count = 196

Mean average precision (mAP@0.50) = 0.999767, or 99.98 %

Therefore, among 223 detections, there are 196 true detections, which gives us a percentage of the average accuracy equal to 99.98%

Table 02 illustrates the matrix of positives and negative detections. The curve of evolution of mean average precision as a function of loss is given by the fig11.

TABLE 02 : CONFUSION MATRIX.

	Positive	
Négative	T (196)	F (4)
		0

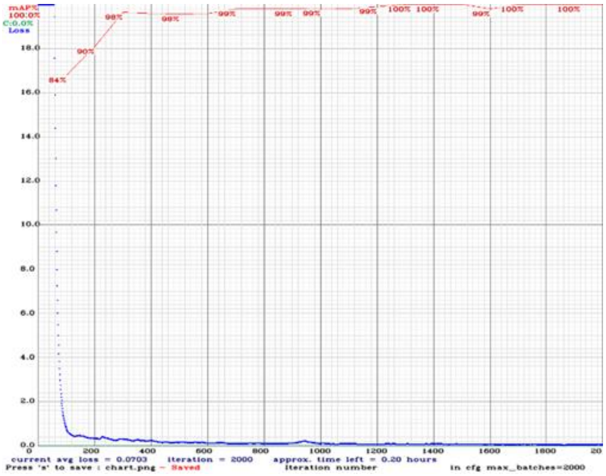

Fig11: Average precession versus loss curve.

Accuracy in our experiment = 0.98 with a recall = 1.00.

This indicates that there is an inverse relationship between the map (with red color) value and the loss (Loss with blue color), the more the value of the loss decreases, the more the precision value increases until it reaches its maximum peak at iteration 800. Whose accuracy is equal to 98% and the loss is monk of 0.1.

Subsequently, we did several tests on several images (from our database or external images). Here is an example showing the results of a test on an external image:

../content/drive/MyDrive/yolo20222/imagesballonsanstrajet/05.png: Predicted in 715.666000 milli-seconds.
ballon: 98%

The image (05.png) as illustrated in fig 12 contains a single object of type (ball) with an excellent percentage (98 %).

In this case, the detection accuracy can be displayed in text form or directly on the tested image, as shown in the following figures:

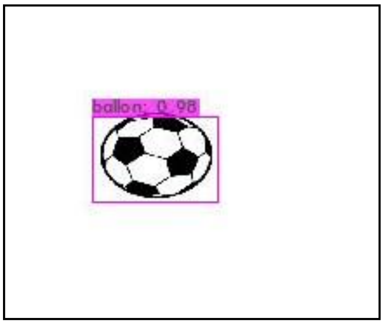

Fig 12: Object detection with CNN

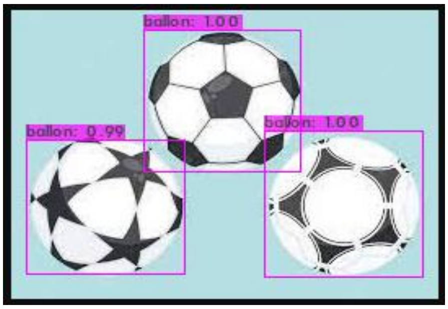

Fig 13: Detection of three objects with CNN.

The fig 13 presents another detection where there are three objects:

There are three ball-type objects, framed by rectangles and on each rectangle there is a percentage of detection (the object on the left 99%, in the middle and on the right with a percentage = 100%).

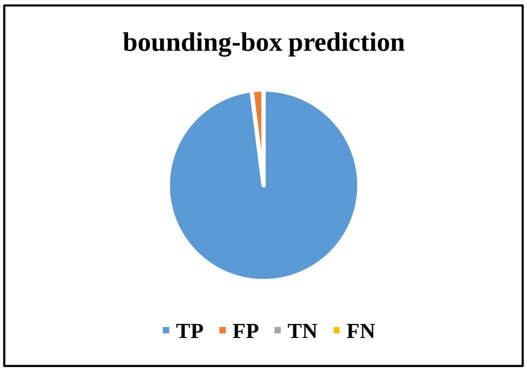

Fig 14: bounding-box prediction.

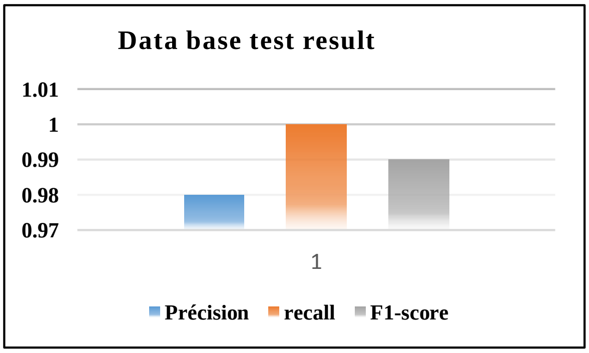

Fig 15: Data base test result.

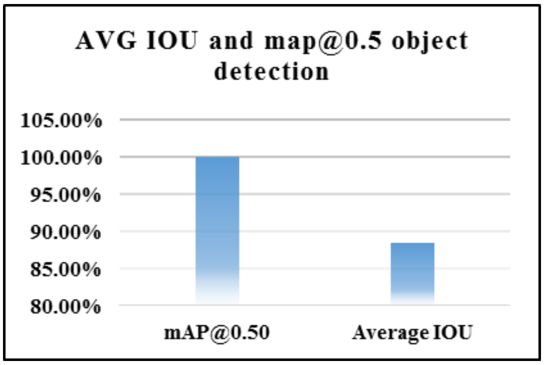

Fig 16 : AVG IOU and Map@0.5 object detection

The detection and location of the class "ball" in the 20 images named 01.png up to 20.png gives the results saved in the form of files (.txt), each file contains in addition to image names, and prediction time, the following information:

Ball: nn%: xx represents the percentage of detection (the percentage that this object is a ball), more than this number (nn) is more than 50 and close to 100, the better the result.

(left_x: xx top_y: yy width: ww height: hh), successively represent the Cartesian coordinates of the center of gravity of the detected object, as well as the length and height of the detection frame created around the detected object.

B. *Trajectory generation*

The next section describes how to create the SCARA arm trajectory using the spline cubic approach and the outcome from the first phase, which is saved in a (.txt) file. To simulate the trajectory generation, we created the model shown in Figure 17.

The following figure shows the SCARA arm and all the steps used to generate the trajectory

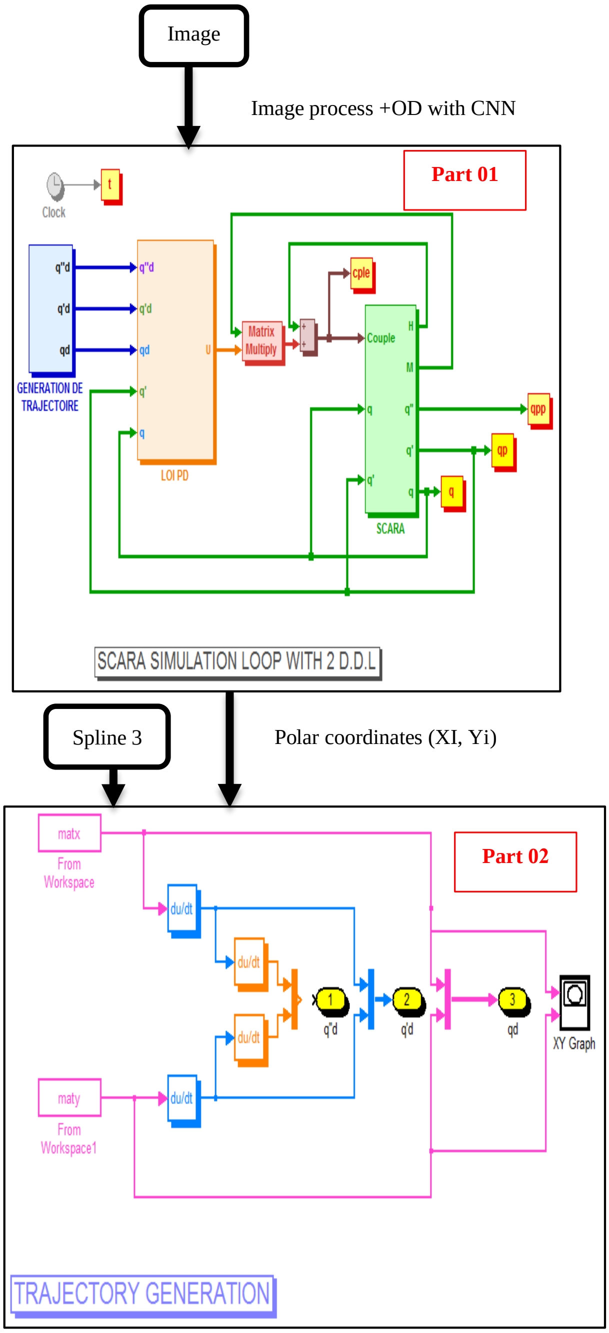

Fig 17: Trajectory generation with SCARA arm

The trajectory produced by the SCARA 2.d.o.f manipulator arm after the (xi,yi) coordinate polar and the cubic spline were entered is displayed in the accompanying figure (Fig. 18). and the (fig. 19) displays the actual trajectory that the monitored object (a ball) made while it was moving.

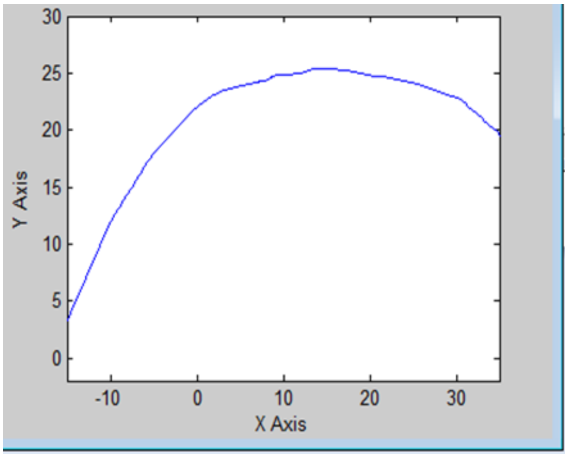

Fig 18: trajectory created by the SCARA

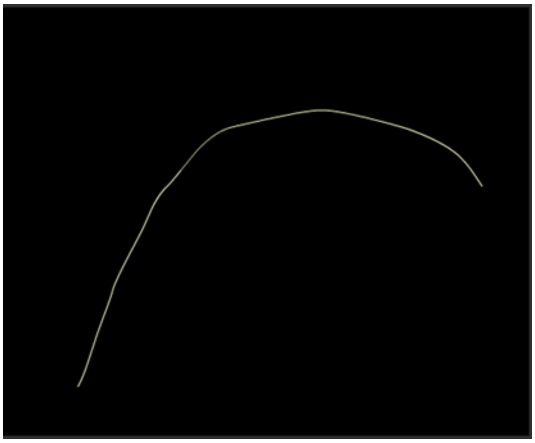

Fig 19: Real manipulator's arm trajectory

C. Evaluation:

We attempted to build the trajectory of the balloon as depicted in Figure 19 utilizing two tools that allowed us to create the trajectory from the same set of points indicated above in order to better evaluate our strategy. Following execution, the outcomes were as follows: The first application is a robust tool for complex data manipulation and advanced analysis, providing granular control over graphics customization. The fig 20 shows the obtaining trajectory.

The second application is a quick and simple method that uses the cited coordinates in columns to generate a static graph. For obtained trajectory, see fig 21.

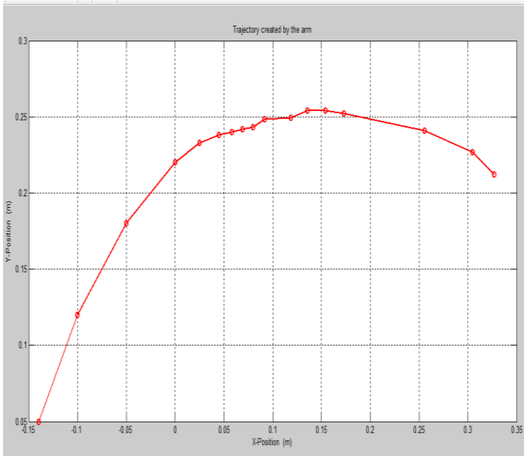

Fig20: Path created by application 1.

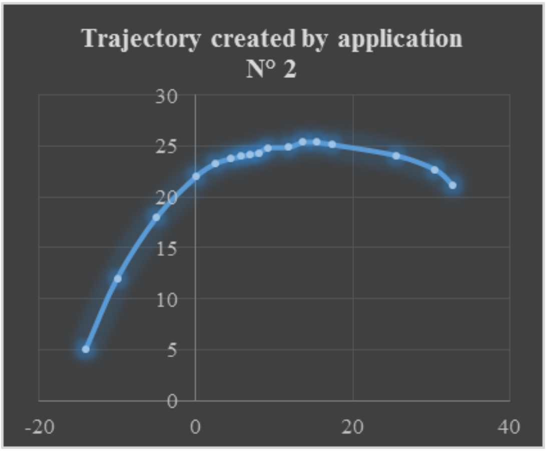

Fig 21: Create path by application 2.

D. Discussion:

To evaluate the quality of our work, we calculated different metrics of similarity between the trajectory generated by the SCARA arm and the real trajectory.

a) Difference between two pictures:

This figure (fig 22) illustrates the difference between the two images.

The graphic effectively illustrates the pixel-by-pixel fluctuations, reflecting the degree of object detection precision attained. We may further improve the trajectory to guarantee smoother and more accurate motions by adding the manipulator arm's kinematic constraints.

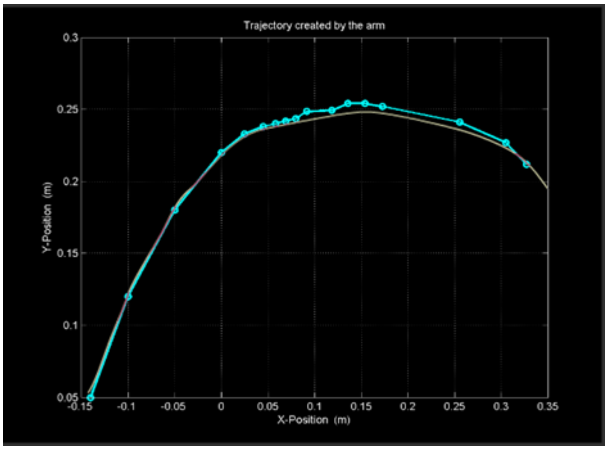

Fig 22: The difference between the two trajectories

b) The difference between two trajectories (two graphs):

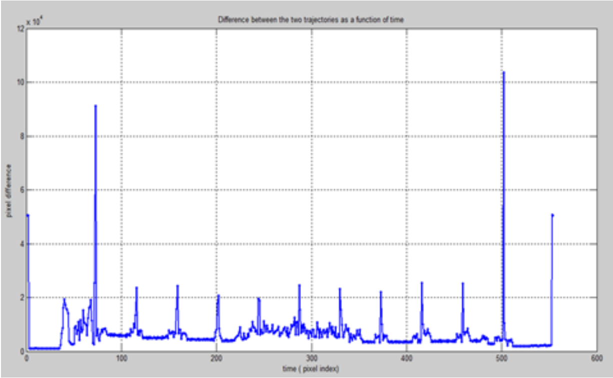

Fig 23: Difference between the two trajectories curve.

We calculated the difference between the pixels, according to each pixel.

We see that the graph contains two peaks, one at pixel level 150 and the second at pixel level 1100. This indicates that the pixels at these locations have a significant difference; they could indicate variations in brightness, color, texture or other characteristics. The other pixels between 0 and 1200 are all between the values 0 and 0.5, indicates that most of the pixels between the two graphs are very similar. The difference between these pixels is small, which means they look almost the same. .

c) Similarity metrics :

The calculation of similarity between two images using metrics provides a quantitative measure of how closely the detected objects match the ground truth, allowing for better detection optimization.

In this section, we used MSE, PSNR, and cosine similarity to evaluate the performance of our detection algorithm.

In order to calculate these metrics, we follow these steps:

First resized the two images containing the real trajectory and the trajectory created by the arm (image1, image2) at the same height and same width.

Convert images to matrices of doubles.

Convert images to grayscale.

Finally, we apply the following formulas:

THE (MEAN SQUARED ERROR) MSE:

$$squared\_difference = (double(image1) - double(image2)).2. \quad (16).$$

$$MSE = mean\,(squared\_difference(:)) \quad (17).$$

Its result is 509.943256. This rating suggests that while the images are still quite comparable, there are some obvious variances between them. Minor changes in the illumination, noise, or other elements may be the cause of differences.

THE PEAK SIGNAL-TO-NOISE RATIO OR PSNR

max_intensity = 255.

$$psnr = 10 * log10((max\_intensity\hat{}2) / mse). \quad (18)$$

Its result is 21.055585dB, which is moreover relatively high, indicating that the quality of image compression or reconstruction is excellent. A high PSNR is frequently linked to the same pictures

COSINE SIMILARITY:

$$vector1 = double(image1r(:)). \quad (19)$$

$$vector2 = double(image2r(:)). \quad (20)$$

$$cosine\_similarity = dot(vector1, vector2) / (norm(vector1) * norm(vector2)). \quad (21)$$

The result of cosine similarity is 0.993691. The two images are extremely similar to one another based on this value's proximity to one.

The figure below depicts the degree of similarity between the two graphs in relation to these three metrics

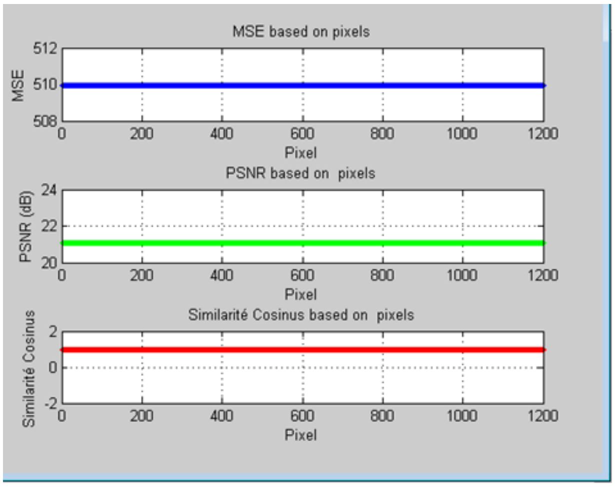

Fig 24: Image metrics comparison based on pixels

d) CONCLUSION

In this project, we applied a method to simulate and generate the trajectory of the SCARA robotic arm. The arm used in this simulation is the SCARA (2.d.o.f) arm. The method used in this project is to detect the object being tracked in a series of images captured by the arm over time then; the results obtained from OD by CNN are used as input to our SCARA robotic arm (2d.o.f), taking into account certain constraints, and following the Denavit - Hetenberg model modified by Khalil. This modified model takes into account the specific kinematic structure of the SCARA robotic arm, enabling accurate and efficient trajectory planning. We have generated a trajectory using the arm model we created previously, to which we will apply the Spline3 method. The accuracy of the obtained results reached 98.99%. The obtained results were evaluated according to the following criteria: Maps, F1score, recall and precision of DO, and MSE PSNR, cosine similarity of the obtained trajectories.

The obtained results show that the method provides a very high degree of similarity between real trajectories and generated trajectories. This method is able to generate trajectories with high accuracy and minimal deviation from the desired path. We conclude that by accurately positioning objects using OD via CNN, we can obtain precise and reliable results when generating trajectories for the SCARA robotic arm. In this project, our first experiment is to try to trace and create a simple trajectory. Such work can bring many benefits in several fields, such as the medical field, where robots can help doctors perform surgeries by tracking required objects. The project did not take into account some constraints, such as Speed, acceleration and collisions.

REFERENCES

[01] M. Sushma Sri, B. Rajendra Naik , K. Jayasankar, B.Ravi, P. Praveen Kumar, ‘‘On the Use of Region Convolutional Neural Network for Object Detection’’, Conference paper ,First Online: 24 May 2021 Data Engineering and Communication Technology, Lecture Notes on Data Engineering and Communications Technologies 63, pp 315-324

[02] C. Vimala ,T. Thenmozhi ,M. Jagadeesh Kumar ,C. Subramani,’’ Convolutional Neural Network-Based Automatic Object Detection on Aerial Images’’, Sixth International Conference on Intelligent Computing and Applications, pp 363-370, Conference paper ,First Online: 28 July 2021,

[03] Aribowo, Wisnu & Terashima, Kazuhiko. (2014). Cubic Spline Trajectory Planning and Vibration Suppression of Semiconductor Wafer Transfer Robot Arm. International Journal of Automation Technology. 8. 265-274.

[04] Tarek Teama ,Hongbin Ma, Ali Maher,Mohamed A. Kassab ,’’ Real Time Object Detection Based on Deep Neural Network’’, International Conference on Intelligent Robotics and Applications ;ICIRA 2019: Intelligent Robotics and Applications pp 493-504| Conference paper First Online: 03 August 2019.

[05] KeJun Ning • Tomas Kulvicius ,Minija Tamosiunaite • Florentin Wörgötter,’’ A Novel Trajectory Generation Method for Robot Control’’,J Intell Robot Syst (2012) 68:165–184, DOI 10.1007/s10846-012-9683-8, Article Published online: 16 June 2012.

[06] Amine Abadi. « Contribution à la génération de trajectoires optimales pour les systèmes differentiellement

plats application au cas d’un Quadri-rotor ». Robotique [cs.RO]. Université d’Orléans ; Ecole Nationale d’Ingénieurs de Sousse (Tunisie), 2020. Français. NNT : 2020ORLE3070. tel-03214986

[07] Abdelkader BENMISRA ,"Programming of industrial robots and application on the robot manipulator Algeria machine tool 1" , Magisterium in Mechanical Engineering , University of Saad Dahleb of Blida (Algeria) - 2007.

[08] Lozano-P´erez, T., “A Simple Motion-Planning Algorithm for General Robot Manipulators”, IEEE Journal of Robotics and Automation, Vol. RA-3,No. 3, pp.224–238, June 1987.

[09] Torsten Kröger , EventsBruno Siciliano • Oussama Khatib • Frans Groen, ‘’On-Line Trajectory Generation in Robotic Systems ,Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor), Springer Tracts in Advanced Robotics Volume 58, 2009.

[10] Khalil W., Dombre E., Modelisation, identification and control of robots, Hermes Penton Science, London,

ISBN 1-90399-613-9, 2002, 480 p.

[11] Luis Arturo Soriano,, José de Jesús Rubio , Eduardo Orozco,, Daniel Andres Cordova,, Genaro Ochoa ,,

Ricardo Balcazar,, David Ricardo Cruz , Jesus Alberto, Meda-Campaña,, Alejandro Zacarias and Guadalupe Juliana Gutierrez.” Optimization of Sliding Mode Control to Save Energy in a SCARA Robot”. Mathematics 2021, 9, 3160. https://doi.org/10.3390/math9243160.

[12] Fateh MAKHLOUFI, ‘’ Modélisation et commande des robots manipulateurs par les outils de l’intelligence artificielle’’, Doctoral thesis. Option :Mechanical Engineering , Badji Mokhtar – ANNABA University , Algeria ,2015.

[13] Vincent Schülé , Dr Dominique Genoud Sierre , " Automatic detection of agricultural objects with Machine Learning " , MSc HES-SO in Business Administration Orientation : Management of Information Systems , university of Applied sciences and arts Western Switzerland 24.08.2018

[14] Simon Corde, ‘’Image Classification And Object Detection By Cnn’’;26 February 2020.

[15] Raghavan M., Roth B. "Inverse kinematics of the general 6R manipulator and related linkages". Trans, of the ASME, J. of Mechanical Design, Vol. 115, 1990, p. 502-508.

[16] Pieper D.L. "The kinematics of manipulators under computer control". Ph.D.Thesis, Stanford University, 1968.

[17] Alia Kamel, S. Aoughellanet, " Neural trajectory planner". Master's thesis. Department of Electronics – University of Batna.2011.

[18] Riad Mansouri and Patrick Siarry. "Metaheuristics for trajectory planning of redundant manipulator arms: application to assistance to surgical gestures in craniotomy", PhD thesis in Signal, Image, Automatic, 01-12-2015, in Paris Est , as part of Doctoral School of Mathematics, Sciences and Technologies of Information and Communication (Champs-sur-Marne, Seine-et-Marne ; 2015-....) , in partnership with Laboratoire Images, Signals and Intelligent Systems (Créteil) (laboratory) and LISSI (laboratory).

[19] José Gutierrez," Modeling and geometric identification of robots used for machining operations ", These Presented to obtain the grade of Docteur d' universite, Order No. : D. U: 2728 E DSPIC: 771 University Blaise Pascal - Clermont II Doctoral School Sciences for the engineer of Clermont-Ferrand Speciality : Mechanical Engineering .20 September 2016.

[20] Chenhuan Feng1, a, Guanbin Gao1, b, Yongli Cao2,’’Kinematic modeling and verification for a SCARA robot’’,, c3rd International Conference on Materials Engineering, Manufacturing Technology and Control. ISBN

10.2991/icmemtc-16.2016.180.ISSN:2352-5401.DOI

10.2991/icmemtc-16.2016.180. (ICMEMTC 2016).

[21] Montazeri, Allahyar & Can, Aydin & Imran, Imil. (2021). Unmanned aerial systems: autonomy, cognition, and control. 10.1016/B978-0-12-820276-0.00010-8.

[22] Paul R.C.P., "Robot manipulators: mathematics, programming and control". MIT, Press. Cambridge, 1981.

[23] Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.:Robotics: Modelling, Planning and Control. https://doi.org/10.1007/978-1-84628-642-1, Springer London , Series E-ISSN 2510-3814,Edition Number:1,

PXXIV, 632. (2009)

[24] Nystrom, M., Norrlof, M.: Path generation for in-dustrial robots. Division of Automatic , Control Department of Electrical Engineering Link¨opings universitet, SE-581 83 Link¨oping, Sweden (2003).

[25] RIAD MANSOURI & PATRICK SIARRY , "Metaheuristics for the trajectory planning of redundant manipulator arms: application to the assistance of the surgical gesture in craniotomy", Doctoral thesis in Signal, Image, Automation. 01-12-2015, in Paris Est, within the framework of the Doctoral School of Mathematics, Sciences and Technologies of Information and Communication. (Champs-sur-Marne, Seine-et- Marne; 2015-....) , in partnership with Laboratoire Images, Signals et Systèmes Intelligents (Créteil) (laboratory) and LISSI (laboratory).

[26] Hiroshi Makino,” Development of the SCARA”, Makino Automation Research Institute, 1-24-14-703 Koenji-Minami, Suginami, Tokyo 166-0003, Japan. JRM Vol.26 No.1 pp. 5-8doi: 10.20965/jrm.2014.p0005, February 20, 2014.

[27] : Bernard Bayle, ‘’ ’Kinematic modeling and control of wheeled mobile manipulators’, Paul Sabatier University, Toulouse, Doctoral thesis. December 2001.

[28] Quentin Agrapart ,Alain Batailly, ‘’ Cubic and bicubic spline interpolation in Python’’, the vibration and acoustic analysis laboratory. Polytechnic Montreal, 20-11-2020. hal-03017566v2.

[29] Lounis ADOUANE.’’. TP Modélisation et Commande des Robots’’. Polytech’Clermont-Ferrand. Novembre 2010.

[30] Karl Mathia ,”Robotics for Electronics Manufacturing - Principles and Applications in Cleanroom Automation”, P26-S:2.4.3. July 2010. DOI:10.1017/CBO9780511712173. Edition: First EditionPublisher: Cambridge University PressEditor: Dr. Philip Meyler.ISBN: 9780521876520.

[31] Jean-Louis Boimond , ‘’ ROBOTIQUE ‘’, cours ISTIA, Université Angers , 2011 . http://lars.mec.ua.pt/public/LAR%20Projects/Humanoid/20.11_RicardoGodinho/Pesquisa%20Disserta%C3%A7%C3%A3o/Cours_robotique.pdf. consulted the 25/08/202.

[32] Abhishek Gupta, Alagan Anpalagan, Ling Guan, Ahmed Shaharyar Khwaja,“Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues”,Array, Volume 10,2021,100057,ISSN 2590-0056,

[33] Khalil W., Kleinfinger J.-F. "A new geometric notation for open and closed-loop robots". Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, p.1174-1 180.

[34] Kousai Smeda ,”Understand the architecture of CNN | by”. (2023). récupéré Sep^tember 24, 2023, form towardsdatascience.com

Aicha Belalia

Bac in 2003.

2008 USTO state computer engineer (computer networks).

2014 USTO master's degree in Computer science (simulation and modeling of systems).

Computer science doctorate student (using AI to simulate robotic systems).

Currently an assistant professor of computer science at a university and a supervisor of L3 students.

Samira Chouraqui

Professor in computer science. USTO.

Robotics laboratory manager.

Supervision of PFE master's, license and doctoral students.

President of CPC of Computer Science Department.

More than 20 publications in several journals.

M’hammed Boussir

Bac, 1991.

1998 USTO state computer engineer (computer networks).

2012 USTO master's degree in physics (matter and radiation).

PhD candidate studying radiation and matter physics.

Is employed with the vocational training institute as a teacher.

Overseeing trainee students in IT technology.

Algorithm researcher in materials physics and radiation