# Scientific project

## *Driving a UR10e with gesture recognition*

---

## *How it works ?*

**Done in the laboratory ARS Control of the University of**

**UNIMORE in the field of a Erasmus exchange**


Written by : **BAHEU JUSTIN**


The : 05/01/2022

*Supervised by :*


**Cristian SECCHI** **Davide FERRARI** →Unimore

**Simona D'ATTANASIO** **Maria HERBAUT-AGUIRRE** → ICAM

To create and launch a machine learning model from scratch you will need to do it in 2 different phases. The first phase will create a database of all the coordinates that you export and the second one will take this database and create a model with machine learning.

- Phase 1 will have 2 programs, “Publisher.py” and “Subscriber.py”.
- Phase 2 will have 2 programs, “Simulation.py” and “Robot_Class.py”.

## How does it works with the terminal ?

### Phase 1 :

1. First of all, open two terminals and go into this folder.

```
baheu@OrdinateurdeJustin:~/ws_sk_tracking/src/sk_tracking$
```

2. On the first terminal, you need to “roslaunch” the Publisher node with this command

```
:~/ws_sk_tracking/src/sk_tracking$ roslaunch sk_tracking Publisher.launch
```

and on the second terminal you need to “rosrun” the Subscriber node with this command

```
:~/ws_sk_tracking/src/sk_tracking$ rosrun sk_tracking Subscriber.py
```

3. Go to the first terminal, this will be written :

```
setting /run_id to f7b44554-79fc-11ec-9487-6b0e8ebae88c
process[rosout-1]: started with pid [4409]
started core service [/rosout]
process[Publisher-2]: started with pid [4412]
The default parameters are the following one :
Webcam : 0
Right Hand : enable
Left Hand : enable
Pose : enable
If you want to use your webcam, 0 is the good value, if you want to use an external camera write webcam after
Do you want to change something in the setup ?
```

Here you will have to choose what functionalities allowed by mediapipe do you want to use. Write “Yes” if you want to disable a mediapie solution, and no if it’s good for you.

4. A new line will appear to ask you which functionalities you want to disable. To do that just write “Left Hand” for example.

```
The default parameters are the following one :
Webcam : 0
Right Hand : enable
Left Hand : enable
Pose : enable
If you want to use your webcam, 0 is the good value, if you want to use an external camera write webcam after
Do you want to change something in the setup ?YES
Please write the name of the parameters that you want to disable
```

5. After that, a new line will appear, to know if it’s good now or if you want to disable anything else.
   For us it is good so I write “GO” and the following line appear.

```
The default parameters are the following one :
Webcam : 0
Right Hand : enable
Left Hand : enable
Pose : enable
If you want to use your webcam, 0 is the good value, if you want to use an external camera write webcam after
Do you want to change something in the setup ?YES
Please write the name of the parameters that you want to disable Left Hand
Are you done with the setup, if it's okay please write : GO , if it's not write anything
```

6. Now you need to set up a project name.

```
Are you done with the setup, if it's okay please write : GO , if it's not write anything GO
What is your project name ?
```

7.1. When your project name is created, firstly go on the second terminal and write “GO”

```
Write GO when you have set the project name : GO
```

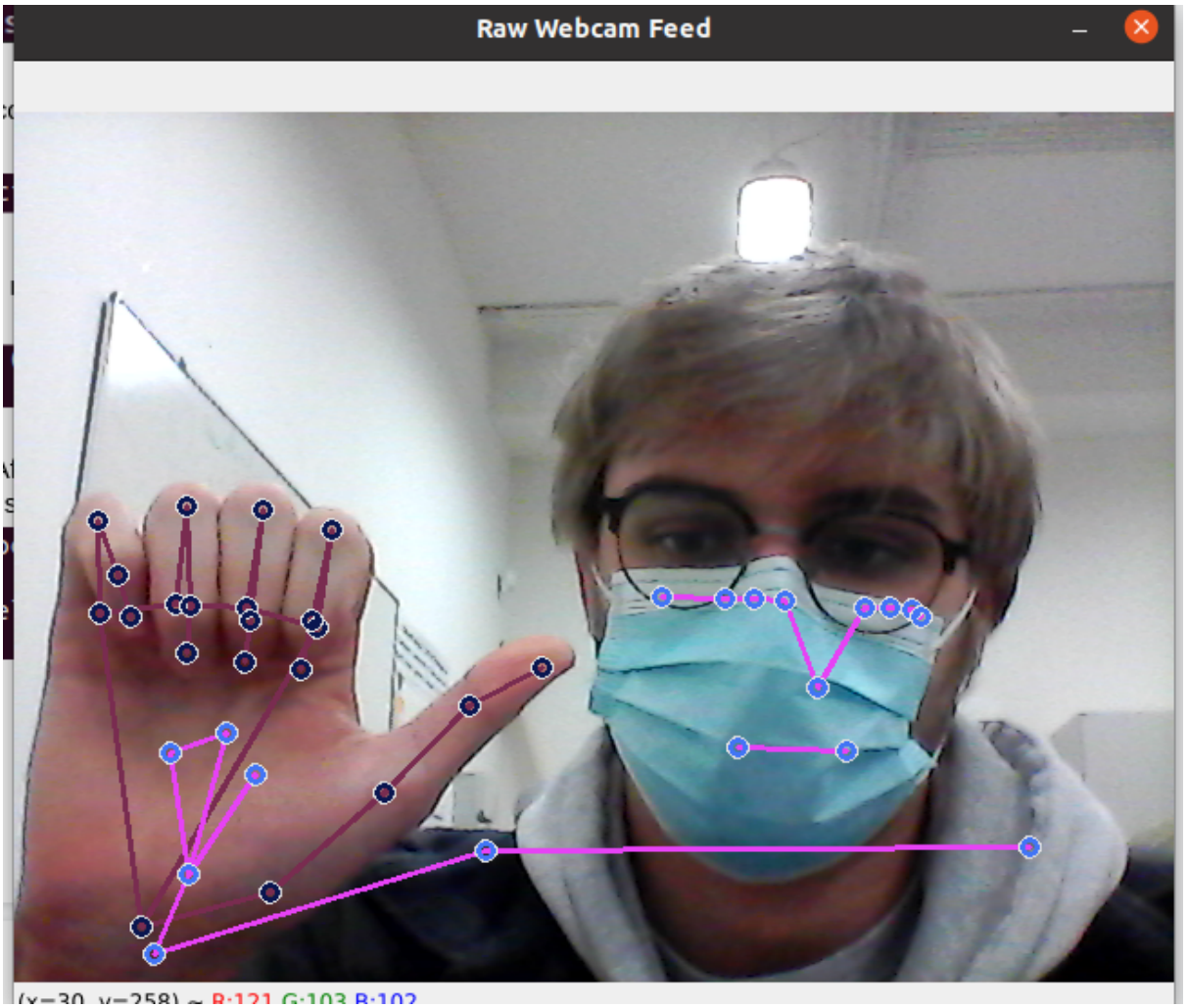
7.2. Secondly, come back to your first terminal and write the number of positions that you want to set up. Here I will create 3 positions.

```
What is your project name ?Tutoriel
How many position do you want to setup? (min of 2)
```

```
How many position do you want to setup? (min of 2) 3
What's the name of your position ?
```

8. When it's done, you need to name your first position. After that a screen will appear and you will need to make the position that you want on the screen with the webcam.

```
What's the name of your position ?rightposition
Press q want your position is setup
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```




So, when this screen is open, make your position for the time that you want, more time you did it, more it will be precise in your machine learning model.

9. To close the screen and pass to the next position, click on the webcam screen and press "Q".

10. The same schema will be done for all the position that you have to set up.
When all your positions is setted up, you can press "control C" to close the two terminals, the first phase is finished.

11. The first phase is finished, now if you all done good, 4 files has been created, you can check if you want to be sure.

- The database which is a CSV file has been created and is stored in that path : /home/baheu/ws_sk_tracking/src/sk_tracking/CSV files

| | Standard | Standard | Standard | Standard | Standard |
|---|---|---|---|---|---|
| 1 | class | x1 | y1 | z1 | v1 |
| 2 | rightpos | 0.4286181330680847 | 1.1068238019943237 | 1.6258208290764742e-07 | 0.0 |
| 3 | rightpos | 0.44762200117111206 | 1.0285969972610474 | 1.0404809245301294e-06 | 0.0 |
| 4 | rightpos | 0.4899757206439972 | 0.8714994192123413 | 9.661128075322267e-08 | 0.0 |
| 5 | rightpos | 0.5192852020263672 | 0.8485918641090393 | -3.068653882110084e-07 | 0.0 |
| 6 | rightpos | 0.5350005030632019 | 0.8553470373153687 | -6.716803113704373e-07 | 0.0 |
| 7 | rightpos | 0.5277411341667175 | 0.864111602306366 | -6.869734079373302e-07 | 0.0 |
| 8 | rightpos | 0.5242841243743896 | 0.8672296404838562 | -6.646804990850796e-07 | 0.0 |

and will store all your landmarks in a CSV file (like an excel)

- A TXT file with your project name

```
1  Tutoriel
```

- A TXT file with your position name

```
1  rightpos leftpos gopos
```

- A TXT file with the solution that you used

```
1  Right Pose
```

## Phase 2 :

1. Before doing anything, in this phase we will be able to pilot the robot so the first thing that is needed to do is to connect to the master of the robot. So after, turning on the robot. Tap this line on another terminal.

```
$ export ROS_MASTER_URI=http://192.168.1.116:11311 && export ROS_IP=192.168.1.142
```

   Just be careful, the ROS_IP need to be egal to your IP address after connecting to the wifi of the lab.

2. So, for this phase you will only need to “rosrun” one program with this line :

```
:~/ws_sk_tracking/src/sk_tracking$ rosrun sk_tracking Simulation.py
```

   After, typing this line you will have this :

```
baheu@OrdinateurdeJustin:~/ws_sk_tracking/src/sk_tracking$ rosrun sk_tracking Simulation.py
Do you want to build a new solution model or to use a previous one ?
Write NEW for a new solution and PREVIOUS for a previous solution :
```

Here, you have two choose, the first one, if you want to build a new model with a CSV that you just created, you need to write “NEW”, if you want to use a previous model that you already created, you can if you write “PREVIOUS”.

Here, we will create a new model with the CSV that you created in the phase 1.

3. Now, you need to choose to start the simulation with an external camera or your webcam.
   Here, we gonna use our webcam so I write “NO”

```
Write NEW for a new solution and PREVIOUS for a previous solution :  NEW
Write YES if you want to use an external camera, otherwise write NO
```

4. This following line will appear. Here you can see all our differents CSV file created. Ours is here, that's the "Tutoriel.csv" file. So to continue, you only need to write the exact name of the file not including the ".csv". So, we gonna write "Tutoriel".

```
Write YES if you want to use an external camera, otherwise write NOno

The following name are your previous position saving, type the name before the point to use it !
['TRobot.csv', 'Testing.csv', 'testrobot.csv', 'Trobot.csv', 'Tutoriel.csv', 'Test.csv']

Which want do you want to use ? (Type the exact name before the point) :
```

5. These few lines will appear. Here you can find the name of the position that you setted up before, the name of the solution that you use.
   You also have a recap of all the action that is setted up to pilot the robot and some lines to ask to the user what action will do our position.

```
Which want do you want to use ? (Type the exact name before the point) :  Tutoriel

You choose Tutoriel and you set up these following position : rightpos leftpos gopos
 This position will use the following solution : Right Pose

For the moment the robot can have 5 different information, GO, Turn left, Turn right, stop and back
In this next line of code you will define what action will do your position

To set up a position you just have to write :
1 for GO
2 for LEFT
3 for RIGHT
4 for STOP
5 for BACK

For which action your position rightpos will be used ?
```

6. So if you follow the instructions we're gonna have this.

```
For the moment the robot can have 5 different information, GO, Turn left, Turn right, stop and back
In this next line of code you will define what action will do your position

To set up a position you just have to write :
1 for GO
2 for LEFT
3 for RIGHT
4 for STOP
5 for BACK

For which action your position rightpos will be used ? 3
3

For which action your position leftpos will be used ? 2
2

For which action your position gopos will be used ? 1
1
```

7. After that, a question will appear to ask you if you made a mistake during the previous steps.
   If it doesn't, write "YES".

```
For which action your position gopos will be used ? 1
1

If you made a mistake during the setup please write NO, otherwise write YES :
```

8. After that, you can wait a few seconds or minutes, it depends on the quantity of data that you create in your CSV file. And a screen will pop up like in the phase 1, but now it will recognize your movements and move the robot.

## How can you add more actions to your robots ?

- First things to know is that, you can build a minimum of two positions and for the moment a maximum of 10 . If you want to build more, you need to modify the "***Setup_of_robot_action***()" function. You just have to add more value on the list called "***liste***" and add more "***elif***" loop.

```
def Setup_of_robot_action():
    with open(f"/home/baheu/ws_sk_tracking/src/sk_tracking/TXT file/Position_Name/Position_Name_{Solution_Choice}.txt", "r") as file:
        allText = file.read()
        words = list(map(str, allText.split()))
        print('\nFor the moment the robot can have 5 different information, GO, Turn left, Turn right, stop and back\nIn this next line of code you will define what action will do your position')
        print('\nTo set up a position you just have to write :\n1 for GO\n2 for LEFT\n3 for RIGHT\n4 for STOP\n5 for BACK')

    global liste
    nbr_pos=len(words)
    liste=["P1","P2","P3","P4","P5","P6","P7","P8","P9","P10"]
    for i in range(0,nbr_pos):
        pos=input(f'\nFor which action your position {words[i]} will be used ? ')
        print(pos)
        if (pos=="1"):
            liste[0]=words[i]
        elif (pos=="2"):
            liste[1]=words[i]
        elif (pos=="3"):
            liste[2]=words[i]
        elif (pos=="4"):
            liste[3]=words[i]
        elif (pos=="5"):
            liste[4]=words[i]
        elif (pos=="6"):
            liste[5]=words[i]
        elif (pos=="7"):
            liste[6]=words[i]
        elif (pos=="8"):
            liste[4]=words[i]
        elif (pos=="9"):
            liste[8]=words[i]
        elif (pos=="10"):
            liste[9]=words[i]
```

- Second thing that you need to do is to create your function to pilot the robot in the "***RobotControl***" class in the "***Robot_Class.py***" file. Here is some example of existing functions, like "***move_straight***" to said to the robot "go straight"

```
def move_straight(self):

    # Initilize velocities
    self.cmd.linear.x = 0.1
    self.cmd.linear.y = 0
    self.cmd.linear.z = 0
    self.cmd.angular.x = 0
    self.cmd.angular.y = 0
    self.cmd.angular.z = 0

    # Publish the velocity
    self.publish_once_in_cmd_vel()

def move_back(self):

    # Initilize velocities
    self.cmd.linear.x = -0.1
    self.cmd.linear.y = 0
    self.cmd.linear.z = 0
    self.cmd.angular.x = 0
    self.cmd.angular.y = 0
    self.cmd.angular.z = 0

    # Publish the velocity
    self.publish_once_in_cmd_vel()
```

- The last thing to do is to manage your loop as you want in the "***detection()***" function in the "***Simulation.py***" file.
  Don't forget to modify it in each if loop called "i***f 'Mediapipie_Solution == "..."*** ".

```
# Here you can set up your robot driving parameters with the name of your class (like STOP, LEFT, RIGHT...) and the probability of precision
Prob=max(pose_recognition_prob)
if (pose_recognition_class==liste[0]) and (Prob>0.7):
    rc.move_straight()
elif (pose_recognition_class==liste[1]) and (Prob>0.7):
    rc.turn_left()
elif (pose_recognition_class==liste[2]) and (Prob>0.6):
    rc.turn_right()
elif (pose_recognition_class==liste[3]) and (Prob>0.7):
    rc.stop()
elif (pose_recognition_class==liste[4]) and (Prob>0.7):
    rc.move_back()
```